EPFL

Thèse n° 8794

Controllability and Interpretability in Affective Speech Synthesis

Présentée le 24 février 2022

Faculté des sciences et techniques de l'ingénieur
Laboratoire de l'IDIAP
Programme doctoral en génie électrique

pour l'obtention du grade de Docteur ès Sciences

par

Bastian SCHNELL

Acceptée sur proposition du jury

Prof. D. N. A. Van De Ville, président du jury
Prof. H. Bourlard, Dr Ph. N. Garner, directeurs de thèse
Prof. J. Beskow, rapporteur
Dr O. Watts, rapporteur
Prof. J.-Ph. Thiran, rapporteur

École
polytechnique
fédérale
de Lausanne

2022

Live as if you were to die tomorrow.
Learn as if you were to live forever.
— Mahatma Gandhi

To my beloved wife and parents ...

Acknowledgements

I like to thank my co-supervisor Phil Garner for his support, guidance, clarity, and nearly full-time availability to help. Not many can explain technical details so coherently and patiently as him. I highly appreciate that he would always give a clear direction on difficult crossroads with his almost infallible intuition. I also like to thank my thesis director Prof. Hervé Bourlard for his relentless effort to make Idiap the comfortable working place it is today, to provide everyone with the required resources, and to make Idiap a well-known institute in the world, which has already opened many doors for me for my future career. I want to extend my gratitude to the staff of Idiap, especially to the administration and system team, for their support and patience. It is thanks to them that we take many things for granted at Idiap, which actually are not in many other places in the world.

I thank the Swiss National Science Fundation for funding my work via the MASS and NAST project. And I want to thank my thesis committee, Prof. Dimitri Van De Ville, Prof. Jean-Philippe Thiran, Prof. Jonas Beskow, and Dr. Oliver Watts for their time reading my thesis and providing constructive feedback to improve it to its current state.

Living abroad is not always easy and comes with challenges, just to name a few: the different language and mentality and the far distance from home. But at Idiap I have found a warm welcoming community from the very first week. With my colleagues I have enjoyed hikes and skiing events, pup crawls and boardgame parties, and many more special occasions. Whether I was following a crazy Chinese on his snowboard through the forests or trying to beat a Belgian in his favourite boardgame, the people around me have made this place a home. I hesitate to provide an exhaustive list (because I will forget some, so please bear with me), but I have to name a few: Angelos, Angel, Apoorv, Banri, François, Nicholas, Pranay, PE, Sargam, Weipeng, and many more. I really hope we will stay in touch even after our paths diverge.

I want to thank my parents for equipping me with all the required skills, internal drive, and stamina to succeed in this challenging task. I still do not understand how they achieved it, but I found everything needed inside me. Last but not least I want to thank my wife, who has supported me throughout this journey, tolerated me when I was stressed, and accepted to accompany me to Switzerland. This thesis would not have been possible without her. I dedicate it all to her.

Martigny, October 24, 2021 Bastian Schnell

Abstract

Thanks to *Deep Learning* Text-To-Speech (TTS) has achieved high audio quality with large databases. But at the same time the complex models lost any ability to control or interpret the generation process. For the big challenge of affective TTS it is infeasible to record databases for all varieties. We belive that affective TTS can only be enabled with models which generalise better to the variability in speech thanks to components which are interpretable by humans.

In this thesis we aim to do so by incorporating prior knowledge about speech and the physiological production of it in the TTS framework. We introduce well-established signal processing techniques to Neural Networks. Starting from emphasised speech we investigate the intonation production with a physiological plausible intonation model previously developed at Idiap. In order to generalise the model to longer prosodic sequences, we emulate a Spiking Neural Network (SNN) with a Recurrent Neural Network with trainable second-order recurrent elements trained with a learning function inspired from SNNs. The model synthesises neutral intonation with high naturalness and retains the physiological plausibility and controllability of the intonation model. After intonation, we look into spectral features in the aspect of formant frequencies, which have shown to be indicators of certain emotions. Based on the speaker adaptation technique *Vocal Tract Length Normalisation* we propose a back-propagatable time-varying All Pass Warp (APW). Experiments of the APW in few- and zero-shot speaker adaptation shows its effectiveness in low-data regimes. In emotional TTS it is not able to increase expressiveness or audio quality, but our analysis shows that the warping correlates with the level of valence in the emotion. We assume that localisation of emotion within an utterance is necessary for the warping (and an affective TTS model in general). We suggest to extract frame-level *emotion intensity* with an emotion recogniser in an unsupervised manner. The emotion intensity input is a scalable and interpretable control and is able to increase the amount of correctly perceived emotion by humans. We also propose to increase the quantity of emotional data with a language-agnostic emotional voice conversion model. The model achieves high-quality emotion conversion in German by exploiting large amounts of emotional data in English. To train it we develop a novel contribution to *gradient reversal*.

We are able to demonstrate that *Deep Learning* TTS models can benefit from well-established signal processing techniques and interpretable low-dimensional controls to improve their generalisability in low-data regimes and/or allow simple controllability without loosing on quality. The developed models also allow interpretability for future physiological and linguistic analysis. While this thesis provides a toolbox of independent controls their combination presents itself as a next step towards a comprehensive framework for affective TTS.

Keywords text-to-speech, affective speech synthesis, intonation modelling, generalised command response model, vocal tract length normalisation, all pass warp, few-shot speaker adaptation, zero-shot speaker adaptation, emotional voice conversion, unsupervised learning, saliency map

Zusammenfassung

Dank *Deep Learning* in Kombination mit grossen Datenbanken hat Text-To-Speech (TTS) mittlerweile eine hohe Audioqualität erreicht. Gleichzeitig haben die komplexen Modelle jegliche Möglichkeit der Kontrolle und Interpretation des Generierungsprozesses verloren. Für die grosse Herausforderung des emotionalen TTS ist es unmöglich grosse Datenbanken in allen Variationen aufzunehmen. Wir glauben, dass emotionales TTS nur mit Modellen ermöglicht werden kann, die besser über die Variationen in Sprache generalisieren und das dank Komponenten, die von Menschen interpretierbar sind.

In dieser Dissertation zielen wir darauf ab dies zu erreichen, indem wir bewährtes Wissen über die Sprache und ihre physiologische Erzeugung in das TTS Framework aufnehmen. Wir bringen wohl-etablierte Techniken der Signalverarbeitung in Neuronale Netze ein. Beginnend mit betonter Sprache untersuchen wir die Erzeugung von Intonation mit einem physiologisch plausiblen Intonationsmodells, das zuvor bei Idiap entwickelt wurde. Um das Modell für längere prosodische Sequenzen zu nutzen, emulieren wir ein Spiking Neural Network (SNN) durch ein Recurrent Neural Network mit trainierbaren Filtern zweiter Ordnung. Das Modell wird mit einer Lernfunktion trainiert, die von SNNs inspiriert wurde. Unser Modell generiert natürliche Intonation für neutrale Sprache und erhält dabei die physiologische Plausibilität und Kontrolle des Intonationsmodells. Nach Intonation betrachten wir spektrale Merkmale in Form von Frequenzen von Formanten, da gezeigt wurde, dass sie Indikatoren für gewisse Emotionen sind. Basierend auf der Sprecheranpassungstechnik *Vocal Tract Length Normalisation* entwickeln wir einen back-propagierbaren, zeitlich variirenden All Pass Warp (APW). Experimente mit dem APW in few- und zero-shot Sprecheranpassung demonstrieren seine Effektivität in Szenarien mit kleinen Datenmengen. In emotionalem TTS ist er nicht in der Lage die Ausdrucksstärke oder Audioqualität zu erhöhen, aber unsere Analyse zeigt, dass das Warping mit dem Level der Valenz in der Emotion korreliert. Wir nehmen an, dass die Lokalisierung der Emotion innerhalb eines Satzes für den APW (und generell für ein emotionales TTS Modell) benötigt wird. Wir schlagen vor mit einem Emotionserkennungsmodell eine Emotionsintensität auf Frame-level durch unüberwachtes Lernen zu extrahieren. Die Emotionsintensität ist eine skalierbare und interpretierbare Stellschraube und erhöht die Menge der korrekt identifizierten Emotionen durch menschliche Zuhörer. Wir schlagen zudem vor die Quantität der emotionalen Daten mit einem sprachen-unabhängigen Emotionskonvertierungsmodells zu erweitern. Das Modell erlaubt die Konvertierung von neutraler zu emotionaler Sprache in Deutsch mit hoher Qualität indem es grosse Mengen emotionaler Daten in Englisch ausnutzt. Um es zu trainieren, entwickeln wir eine *gradient reversal* Erweiterung.

Zusammenfassung

Wir demonstrieren, dass *Deep Learning* TTS Modelle von wohl-etablierten Techniken der Signalverarbeitung und interpretierbaren niedrig-dimensionalen Kontrollparametern profitieren können. So können ihre Fähigkeit zur Generalisierung in Szenarien mit kleinen Datenmengen erhöht und/oder eine einfache Kontrolle ermöglicht werden, ohne dabei an Audioqualität zu verlieren. Die entwickelten Modelle ermöglichen zudem die physiologische und linguistische Analyse in der Zukunft. Während diese Dissertation nur einen Werkzeugkasten von unabhängigen Kontrollmechanismen vorstellt, ist die Kombination derselben der naheliegende nächste Schritt in Richtung eines umfassenden Frameworks für emotionales TTS.

Keywords Text-zu-Sprache, text-to-speech, emotionale Sprachsynthese, Intonationsmodell, Generalised Command Response Model, Vocal Tract Length Normalisation, All Pass Warp, few-shot Sprecheranpassung, zero-shot Sprecheranpassung, Emotionskonvertierung, unüberwachtes Lernen, saliency map

Contents

List of Figures

List of Tables

Glossary

AI	Artificial Intelligence
APW	All Pass Warp
ASR	Automatic Speech Recognition
BAP	Band-Aperiodicity
BiLSTM	Bidirectional Long Short-Term Memory
CNN	Convolutional Neural Network
CR	Command-Response
DNN	Deep Neural Network
DTW	Dynamic Time Warping (Berndt and Clifford, 1994)
E2E	End-to-End
EM	Expectation Maximization
EVC	Emotional Voice Conversion
FFT	Fast Fourier Transform
G2P	Grapheme-to-Phoneme
GAN	Generative Adversarial Network
GCR	*Generalised* Command Response
GMM	Gaussian Mixture Model
GRU	Gated Recurrent Unit
HMM	Hidden-Markov-Model
HTK	Hidden-Markov-Model Tookit (official website)
IIR	Infinite Impulse Response
KL	Kullback-Leibler
KLD	Kullback-Leibler Divergence

LF_0	Logarithmic fundamental frequency
LPC	Linear Predictive Coding
LSTM	Long Short-Term Memory
MCD	mel-cepstral Distortion
MCEP	mel Cepstrum
MFCC	mel Frequency Cepstral Coefficients
MGC	mel-generalised Ceptrum
MLP	Multi-layered Perceptron
MLPG	Maximum Likelihood Parameter Estimation (Tokuda et al., 2000)
MOS	Mean Opinion Score
MSE	Mean-Squared-Error
NLU	Natural Language Understanding
NN	Neural Network
ReLU	Rectified Linear Unit
RMSE	Root-Mean-Square Error
RNN	Recurrent Neural Network
SDM	Spring-Damper-Mass
SELU	Scaled Exponential Linear Unit
SNN	Spiking Neural Network
TTS	Text-To-Speech
UA	Unweighted Accuracy
V/UV	Voiced/Unvoiced
VAE	Variational Autoencoder
VC	Voice Conversion
VQ-VAE	Vector-Quantised Variational Autoencoder
VTLN	Vocal Tract Length Normalisation
WA	Weighted Accuracy
WMSE	Weighted Mean-Squared-Error

1 Introduction

The last 10 years have seen a big boom in Artificial Intelligence (AI) assistants often as part of operating systems for mobile devices and computers, like Google Assistant, Siri, or Cortana; or as the main interface for smart home devices like Alexa, Google Home, Swisscom Box. AI assistants rely on a sophisticated pipeline of AI components combined through heavy engineering (Figure 1.1). Most assistants are activated through a dedicated wakeup word or phrase like 'Alexa' or 'Hey Google' which is detected on device. Once active the device starts recording and sends it to the online Automatic Speech Recognition (ASR) system. The recognised text runs through a Natural Language Understanding (NLU) component that identifies keywords to detect the user's command(s). The commands can activate a variety of (third-party) services which generate a response. Depending on the device the response triggers an action on the device and/or the assistant replies with its own voice. To reply an online Text-To-Speech (TTS) system is used. The generated audio is sent back to the device and played there to communicate the result(s) to the user.

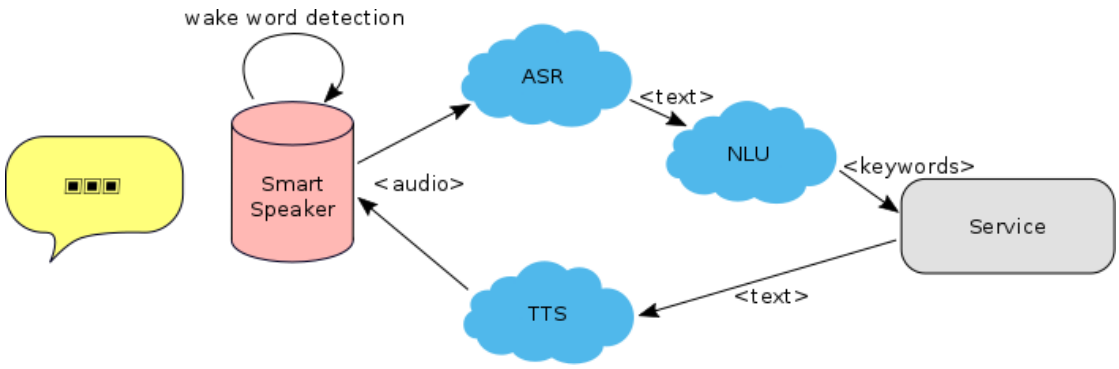


Figure 1.1 – An oversimplified representation of the different AI components of a smart speaker with online processing.

1.1 Motivation

In most scenarios the AI assistant understands the user and performs the task reliably. However, today's systems are completely incapable of handling affect neither on the recognition nor on the generation side. The next step in human-computer interaction requires detection and generation of affect, e.g. emotions, on the AI side. We can roughly order the types of non-neutral speech by increasing intensity into emphatic/emotional, over expressive, to affective speech. Moving Picture, Audio and Data Coding by Artificial Intelligence (MPAI) has just released a call for technologies to enable a standard for the conversation with emotion in full AI-based implementations.[1]

Detection of emotion is a controversial topic. More and more companies claim to be able to predict emotions from audio-visual cues. The emotion recognition industry is likely to substantially grow in the next five years. A moderen AI would certainly benefit from the ability to recognise frustration or joy to react to it. Other applications may be dangerous or unethical like flagging pupils because a Neural Network (NN) predicts that they look angry or surveillance of remote workers during the pandemic. This is especially concerning because no formal validation of the systems from a government body is required. Experts of the field start to ask for regulation for tools that make claims about the human mind similar to regulations in medicines (Crawford, 2021). Such regulations might also enforce a level of explainability; on what bases did the NN make the decision. A question that is still totally open in *Deep Learning*.

Modern TTS has achieved audio quality indistinguishable from human speech. The development is driven by NNs of increasing complexity trained on an ever increasing amount of data. Every component from grapheme-to-phoneme modelling, over duration and acoustic features prediction, to vocoding can now be done by NNs (we give an introduction to each of them in Chapter 2); often better than any other method before. While the performance of these models is impressive, controllability and interpretability have been lost on the way.

One of the next big challenges in TTS is affective speech synthesis, varying from simple emphatic, over expressive, to emotional speech. It would certainly be possible to record large affective databases and apply the same complex models, however, the mere variability of languages, speakers, emotions, and affect intensities makes this the proverbial fight against windmills. What can we do when the general trend of big data, big networks, big computation is not the solution? Modern affective TTS models have to make better use of the limited data they can train on, they need integrated mechanisms which help to produce affective speech, and they need interpratable controls to be useful in many scenarios. Affective TTS is not only a dream for smart assistants, it is also desirable for the automatic generation of audiobooks or automatic dubbing, eventually leading to dubbing for low resource languages for which usually no/low-quality dubbing is available.

[1] https://mpai.community/2021/02/06/having-a-conversation-with-a-machine/

1.2 Scope of the Thesis

We believe that modern affective TTS can only be enabled with models which make better use of the available training data with increased generalisability and which offer control over the different generation aspects in a way that is interpretable/understandable by human users. In this thesis we aim to do so by incorporating prior knowledge about speech in general and the physiological production of it in the TTS framework. We introduce well-established signal processing techniques to NNs, smoothing the ways for their renaissance in modern deep learning. In this work we investigate a set of aspects of affective speech synthesis independently, but this forms only the first step towards a comprehensive controllable and interpretable framework of affective speech synthesis.

Pitch is an important aspect of affective speech. An intonation model offers interpretability and control of it. Incorporating an intonation model into a NN would allow to benefit from recent advances in deep learning. As a starting point we can rely on a physiologically plausible model of intonation previously developed at Idiap; the *Generalised* Command Response (GCR) model (Honnet et al., 2015).

Adaptations in spectral features play another important role in affective speech. Emotion recognition research found that the first two formant frequencies are effective indicators of certain emotions (Vlasenko et al., 2011). We aim to develop a technique to shift the whole spectrum depending on the target emotion in a continuous way that maintains high naturalness over the entire transformation. Such a technique would provide us with high controllability while maintaining high naturalness. Vocal Tract Length Normalisation (VTLN) is well established as a speaker adaptation technique that can work with very little adaptation data (Sundermann and Ney, 2003). Adapting it to a time-varying NN component could provide us with the necessary controllable formant shifting.

Many emotions are not displayed continuously in an otherwise emotional utterance; rather, the intensity varies with time. We hypothesise that emotion is localised within an utterance and that this intensity information would benefit TTS models trained in low data regimes. With more data one can expect that the model learns the inherent patterns on its own. We expect that translation agents could infer the intensity information from the source language. TTS systems are likely to require more training data to infer it from text. An emotion intensity input to the TTS model is also desirable because it is a scalable and interpretable control to vary the emotion intensity in the whole or parts of the sentence. We identified two research directions: 1) Increase the quality of the emotional training data e.g. with explicit emotion localisation labels. 2) Increase the quantity of the emotional training data so that the model learns to infer the localisation from text without explicit labels.

1.3 Main Contributions

This thesis contains four broad contributions, which is also reflected in the chapter structure:

i In Chapter 3, "A Neural Generalised Command Response Model", we present a Recurrent Neural Network (RNN), which emulates a Spiking Neural Network (SNN), followed by a post-processing step with a fixed dictionary of muscle respones to model intonation with the GCR model. We show that a loss function for error backpropagation can be formulated analogously to that of the Spike Pattern Association Neuron (SPAN, Mohemmed et al. (2013)) method for spiking networks. We then propose an end-to-end neural architecture that replaces the post-processing step with fixed dictionary with trainable second-order recurrent elements (Marelli, 2018) analogous to recursive filters. Subjective listening tests demonstrate that both system can synthesize neutral intonation with high naturalness, comparable to state-of-the-art acoustic models, and retain the physiological plausibility and controllability of the GCR model.

ii It is well known that VTLN can be cast as a linear transform in the cepstral domain (Pitz and Ney, 2005). Building on this latter property, we show in Chapter 4, "Neural All Pass Warp", that it can be cast as a (linear) layer in a NN. We generalise it to a back-propagatable time-varying warp, which is best described as an All Pass Warp (APW). The APW offers a low-dimensional interpretable control to alter the spectrum, which by design generalises over different speakers. We investigate few- and zero-shot speaker adaptation in multi-speaker models, followed by emotional multi-speaker TTS in low-data regimes. The models with APW layer are able to outperform those without in the multi-speaker tasks in subjective listening tests, supporting our hypothesis that the APW improves generalisability. While the APW is not able to increase expressiveness or audio quality in the emotional TTS task, our analysis shows that the warping correlates with the level of valence in the emotion.

iii Following our results with the APW we assume that the localisation of emotion within an utterance is required for better use of the warping. In Chapter 5, "Emotion Intensity", we show that an emotion recogniser is capable of producing a measure of emotion intensity via attention or saliency; this measure is appropriate to label utterances (thus increasing their quality) subsequently used to train a speech synthesiser. We evaluate novel and published means to do this showing that, whilst it is no longer state of the art for emotion recognition, attention is a good way to indicate emotion intensity for speech synthesis.

iv To increase the quantity of the emotional training data we propose in Chapter 6, "Emotional Voice Conversion", a language-agnostic Emotional Voice Conversion (EVC) model. The model achieves high-quality emotion conversion in German with limited data by exploiting large amounts of emotional data in US English. It is an encoder-decoder model with adversarial training to remove emotion leakage from the encoder to the decoder. The adversarial training is improved by a novel contribution to gradient reversal to truly reverse gradients. This allows to remove only the leaking information and to converge to better

optima with higher conversion performance. Evaluations show that the model can convert to different emotions at high quality but misses on emotion intensity compared to the recordings, especially for very expressive emotions.

1.4 Outline

This thesis consists of seven chapters of which Chapter 3 - 6 contain the four broad contributions. The current chapter started with an introduction to the field, the motivation for the thesis, and its scope.

The next Chapter, 2, covers background information required at different points in the thesis. It covers the major developments the field of TTS has gone through in the last five years, as well as new methods in general machine learing, that already found their way into TTS systems. The chapter also contains objective and subjective measurements, databases, and features commonly used in TTS and also this thesis.

Chapter 3 describes how to drive the GCR model with an RNN, emulating an SNN. It also include the extension to an end-to-end model with integrated neural filters.

In Chapter 4 we present the generalisation of VTLN to a time-varying APW and its back-propagatable implementation as an integrated component of an RNN.

In Chapter 5 we present unsupervised methods to extract emotion intensity from emotional recordings. We prove that an emotional TTS model with this emotion intensity input greatly improves the expressiveness without degradation in audio quality.

Chapter 6 proposes an EVC model to generate synthetic emotional data that can be used to improve a TTS model in the future. To prevent emotion leakage a novel improvement to gradient reversal is proposed.

Finally, Chapter 7 concludes the thesis and summarises the key points. It also gives recommendations for future work.

2 Background

In this chapter, we cover background topics required to understand general aspects of this thesis. It will give an overview over the main developments in text-to-speech research and put them in temporal order. It also presents objective and subjective performance measures used as standards in the field and explains the different databases as well as common features used across all experiments.

2.1 Deep Learning

Deep learning has emerged as a dominant concept in the machine learning community. Over the last two decades it has achieved outstanding results in many fields such as image (Krizhevsky et al., 2012), face (Taigman et al., 2014), and pose (Cao et al., 2017) recognition, any kind of pattern recognition (LeCun et al., 2015) in large databases (e.g. bioinformatics), ASR (Amodei et al., 2016), NLU (Vaswani et al., 2017), and TTS (Wang et al., 2017b). The increase of computational power and the advent of graphical computing was an important driver for the success of NNs, but only in the sense that it made them applicable to many more problems. The main reason why NNs are so successful/powerful is that they are universal function approximators (Cybenko, 1989; Hornik et al., 1989). The Universal Approximation Theorem proves that a NN is able to approximate any continuous function as long as it has sufficient capacity; and most realworld problems can be expressed as functions (some with excessive complexity though). This holds even true for NNs with a single hidden layer. However, practically the model might fail to generalise or to learn. It is important to state that the layers have to have a non-linearity like Rectified Linear Unit (ReLU) or sigmoid for the Universal Approximation Theorem to hold. While linear transformations, ReLUs, and sigmoids make NNs universal function approximators, they move them further away from their biological origin, the human brain. Processes in the brain rather follow signal processing operations with asynchronous timings. If we want to resemble processes of the human body into NNs we need to integrate signal processing techniques in the models.

The algorithm which made NNs trainable is the backpropagation algorithm (Rumelhart et al., 1986). It computes the partial derivatives of the loss w.r.t. to every trainable parameter in the network. It traverses the network in reverse order computing the partial derivatives of every layer and reuses the intermediate results for the next error (dynamic programming). In this thesis we rely on the PyTorch deep learning framework (Paszke et al., 2019) writtin in the Python programming language. PyTorch provides a rich *Autograd* class (Paszke et al., 2017) that handles most backpropagation computations entirely concealed from the programmer. In every forward pass it automatically builds a dynamic computational graph that can be traversed backwards to assign gradients to every network parameter involved in the computation. As long as we rely on PyTorch operators in new NN components, Autograd will enable backpropagation without explicit implementation. We will exploit this property at various points in this thesis.

2.1.1 An introduction to VAE

A *Generative Model* is a statistical model which models the joint probability distribution of an observable and a target random variable. In contrast to discriminative models they allow to generate new data similar to the training data. One of the two major families in generative models is the Variational Autoencoder (VAE) (Kingma and Welling, 2014). As the name suggests the model is practically an autoencoder (Rumelhart et al., 1985; Bourlard and Kamp, 1988) with some constraints on the latent encodings. The goal is to create a latent space from which one can sample unseen representations of the data, thus generate new data of the same domain. This is a fundamental shortcoming of the standard autoencoder. As there is no constraint on the latent space (the objective is only to encode and decode with minimal reconstruction error), selecting a random point from it will not necessarily generate good results, as it could lie far away from the training points.

A VAE encodes the input as a distribution instead of a single point. During training points sampled from the distribution are decoded with the goal of lowest reconstruction error. The encoder predicts the mean and (log) variance conditioned on the input. This alone does not produce a better latent space. The encoder could learn to predict very small variances effectively transforming to a point distribution similar to autoencoders, or it could predict the means very far from each other. VAEs prevent that by introducing a regularisation term. It requests the latent space to be close to a chosen prior distribution in terms of Kullback-Leibler Divergence (KLD). In most applications the prior distribution is chosen to be Standard Normal and the predicted latent distribution is Gaussian. The main reason for it is that a closed form solution for the KLD between the two exists, which is not the case for most other distributions. The regularisation forces the latent space to be close together and possibly overlap, it also prevents the variance from becoming too small. The network has to learn a meaningful way of overlapping samples to minimise the reconstruction error. When properly trained sampling from the prior distribution will generate meaningful new samples.

VAEs are an interesting method to sample different representations for the same input text/-conditioning. They allow variations for the same speaker and text which makes the audio less monotonic. They have shown to be useful to model the space of speaker embeddings as they nicely model variations in the speakers' voice (Hsu et al., 2017b).

2.1.2 An introduction to GAN

A Generative Adversarial Network (GAN) in machine learning is a fairly recent type of generative model (Goodfellow et al., 2014). The idea is to train two networks: a generator and a discriminator. The input to the generator are sampled from a prior distribution and the generator network generates samples of the target domain. To apply backpropagation it would be necessary to compare the true and generated distributions, e.g. with maximum mean discrepancy (Dziugaite et al., 2015). However, as it is practically difficult to implement, GANs use a second auxillary network called discriminator. The discriminator is trained to identify whether a sample was generated by the generator or is a real sample from the database. When the generator and discriminator are concatenated the whole structure can be trained with the classification loss of the discriminator alone. To handle multiple parts of the target domain generator and discriminator can also be conditioned, e.g. on a class label.

Training GANs is known to be tricky (Salimans et al., 2016). The goal is to reach an equilibrium where the generator produces only samples closely following the target domain and the discriminator's classification accuracy dropped to 50%, i.e. purely random. The networks are trained alternating with the other network fixed. The generator is trained to increase the classification error of the discriminator while the discriminator is fixed. The discriminator is trained to decrease its classification loss while the generator is fixed. From a back-propagation perspective the difference is just in the sign of the gradients. An overly powerful discriminator can make the generator be stuck, thus the generator usually receives more training epochs than the discriminator.

Some GANs have been used in TTS (Zhao et al., 2018), neural vocoders (Juvela et al., 2019b), full end-to-end models (Binkowski et al., 2020), speech enhancement methods (Donahue et al., 2018), and voice conversion (Gao et al., 2018).

2.2 Features

In this section we describe commonly used features in TTS and throughout this thesis. Instead of describing each feature over and over again we will refer back to this section in the rest of the thesis where possible. While their description might seem a bit abstract and unrelated here, they are necessary for understanding the recent trends in TTS presented in the following section.

Phonemes are units of sound which represent words in a phonological and linguistic way. Phonemes have a lingustic origin but here we interpret them as acoustic realisations in the context of TTS. A phonetic representation is then available from so called Grapheme-to-Phoneme (G2P) models. The first models were simple dictionary look-ups. They are obviously limited by their size, costly to create, too big for mobile devices, and do not work on out-of-domain words. Rule-based methods formulated as finite-state machines overcame some of the limitations (Kaplan and Kay, 1994). However, they usually require a dictionary for pronunciation exceptions and the creation of the rules requires lingustic experts. Data-driven approaches came for the rescue, trading the requirement of providing rules against providing sufficient examples. All kinds of machine learning techniques have been applied to the problem. A thorough review is beyond the scope of this thesis. A good overview of pre-NN techniques can be found in Bisani and Ney (2008). More recently some approaches of neural G2P models were proposed (Bilcu, 2008; Rao et al., 2015; Yao and Zweig, 2015). For our experiments we rely on the rule-based system of the Festival toolkit (Black et al., 1998) to extract phoneme sequences from text.

Context embeddings (sometimes also referred to as question features) represent contextual input features originally developed for ASR and later expanded for TTS models developed with the Hidden-Markov-Model Tookit (official website) (HTK) (Woodland et al., 1994). They were later used for Merlin-style TTS models as well (Section 2.3.3). They first need to be forced aligned (with the audio) so that the length of the input context embeddings matches the length of the acoustic frames. Forced alignment describes the process of creating a time-aligned version of a transcription (in our case phonemes) with the audio. In principal the model builds increasingly complex ASR systems to recognise the phonemes in the audio. After each training the data is aligned and reused in the next iteration. Because the correct sequence is known the decoding step is trivial and the alignment very accurate. The alignment will vary negligibly between algorithms, but this is also the case for human annotators. In most of our experiments we use HTK to compute forced-alignments with context-independent HMMs. HTK computes state-level alignments, which means that every phoneme is split into five states. For the experiments in Chapter 6 we use the Montreal Forced Aligner (McAuliffe et al., 2017), which produces phoneme-level alignments (single state per phoneme). From the aligned phoneme sequences we generate context embeddings with 425 text-derived binary and numerical features. The questions insert context and prior knowledge into the context embedding. For example, context is added by explicitly describing the previous and following two phonemes, as well as the position of the current word in the utterance, and the position of the current phoneme in the word. Prior knowledge is added through flags for vowels, fricatives, and other properties of the language. The 425 questions we use are suitable for UK and US English. The context embeddings are normalized to [0.01, 0.99] before passed to any network.

Durations describe the length of each phoneme in the audio. They require a forced-alignment step before. For state-level alignments (e.g. HTK) the durations of the five states per phoneme are summed. Duration prediction is a challenging task on its own, especially for affective speech. In this thesis we exclusively rely on oracle durations.

WORLD acoustic features, consisting of Logarithmic fundamental frequency (LF_0), 30-dim mel-generalised Ceptrum (MGC), and one Band-Aperiodicity (BAP) at 5 ms frame step, are extracted with the WORLD vocoder (Morise et al., 2016) (D4C edition (Morise, 2016)). We interpolate LF_0 before training and add a binary Voiced/Unvoiced (V/UV) flag to represent voicing information. We perform mean/variance normalization for all but V/UV. From these features waveforms are generated with the WORLD vocoder. Some models also require the dynamic components of those features, also referred to as deltas and double deltas/delta deltas (Δ and $\Delta\Delta$) for all but the V/UV flag. They correspond to the first and second derivatives. As speech is continuous over short time windows synthetic speech can be improved by enforcing smooth contours of the signal and its derivates. Maximum Likelihood Parameter Estimation (Tokuda et al., 2000) (MLPG) is an algorithm which produces a smooth trajectory of a predicted sequence based on the predicted first and second derivatives and their statistical properties. This technique has shown to be superior compared to models predicting only the output features without their dynamic component.

openSMILE (Open-source Speech and Music Interpretation by Large-space Extraction) is an open-source framework for the extraction of hand-crafted audio features from speech and music signals. It provides the most commonly used feature sets in emotion recognition (El Ayadi et al., 2011). The toolkit is able to extract features from chunks of frames up to the full audio. In speech processing it is often used with a fixed length sliding window (Ramet et al., 2018), for frames of the same phoneme (Lei et al., 2020), or the entire utterance (Zhu et al., 2019). A well known feature set is the *IS09* set, which was used in the INTERSPEECH 2009 Emotion Challenge (Schuller et al., 2009). The 384-dim features consist of hand-crafted Low-Level Descriptors (pitch, energy, zero-crossing rate, voicing probability), 12 mel Cepstrum (MCEP) coefficients, and others as well as their first derivative.

2.3 TTS through the ages

A TTS system is a generative model. It models the joint probability of the input character / phoneme sequence with the sequence of the acoustic features. Following Bayes' theorem it can alternatively be formulated as modelling the conditional probability of the acoustic features given the character/phoneme sequence. In all methods the text to synthesise runs through a normalisation step, where numbers and dates are spelled out based on rules, e.g. '1st Dec' becomes 'the first of December'. Then a G2P model is used to convert to a sequence of phonemes.

2.3.1 Unit Selection

Unit selection tries to select and concatenate units of speech extracted from a speech database (Hunt and Black, 1996; Campbell and Black, 1997). Because the units are actual speech, which already contain all the fine details and nuances of human speech, the quality is very high. For a given sequence of phonemes the task is to select units best matching the required context

with minimal distortion when concatenated. Smoothing with signal processing techniques is still necessary at the transitions between units to address concatenation artefacts. NNs were also proposed to help with the smoothing (Merritt et al., 2016).

While it is tempting to speak about unit selection as *pre-deep learning*, it is not at all true. Unit selection speech synthesis has been the method of choice for most synthetic voices for a long time, because of its high quality. Only with the invention of *Parallel WaveNet* (Oord et al., 2018) the quality and performance requirements of industrial TTS systems were met. Over the last three years NN-based TTS has replaced unit selection in most applications.

2.3.2 HMM Synthesis

Early data-driven generative models for TTS were build from Hidden-Markov-Models (HMMs) (Yoshimura et al., 1999). Similar to many other fields HMMs were the most successful data-driven machine learning technique before the advent of NNs. Context dependent phoneme HMMs are used, where the context can be thought of as a subset of the context embeddings described in the previous section. The context is often obtained with a decision-tree-based clustering technique (Shinoda and Watanabe, 2000). The output of the multistream HMMs are continuous probability distributions. For the prediction of LF_0 slightly different HMMs are used than for spectral features. The HMM parameters are learned from the data with the Expectation Maximization (EM) algorithm. During synthesis a text is first converted with a G2P model and context-dependent phoneme labels are obtained. Then a sentence HMM is created by concatenation of context-dependent phoneme HMMs. Spectral and F_0 parameters are sampled from the sentence HMM based on the Maximum Likelihood criterium. To prevent discontinuities in the generated speech the dynamic features are predicted as well and an algorithm like MLPG is used to smooth them. HMM speech synthesis is mostly intelligible, but its quality is far from the recordings.

2.3.3 Merlin-style TTS (pre 2017)

The successor of HMM-based TTS is NN-based TTS, which can be split into two successive developments. In this thesis we will refer to the first as the old paradigm or Merlin-style TTS (named after the Merlin toolkit, Wu et al. (2016)). For the second we will use new paradigm, encoder-decoder models (named after its underlying architecture), or Tacotron-style models (named after the first published model of its kind, Wang et al. (2017b)) interchangably. We will present the former in this section and the latter in the next section.

In the old paradigm duration and acoustic feature prediction is split (Figure 2.1); we distinguish duration and acoustic model. The duration model receives phoneme inputs as one-hot vectors (the vector has as many dimensions as phonemes used; the vector is one at the dimension of the current phoneme and zero otherwise). It then predicts the duration for each of the phonemes. This is usually done at state-level and requires state-level forced alignment.

The inputs to the acoustic model are context embeddings. They are upsampled to match the durations. At training time the oracle durations are used, for inference the durations are predicted by the duration model. The acoustic model then predicts the acoustic features for each frame. Usually the features extracted by well known vocoders are used as acoustic features, so that the vocoder can convert them to the waveform. Similar to other fields feed-forward NNs were later replaced by RNNs in both models. The duration and acoustic model are trained independently, which is one of the major weaknesses of the old paradigm. While the new paradigm does not split into duration and acoustic model, recently separated models came into fashion again, because they allow better control over the duration (Ren et al., 2019; Yu et al., 2020).

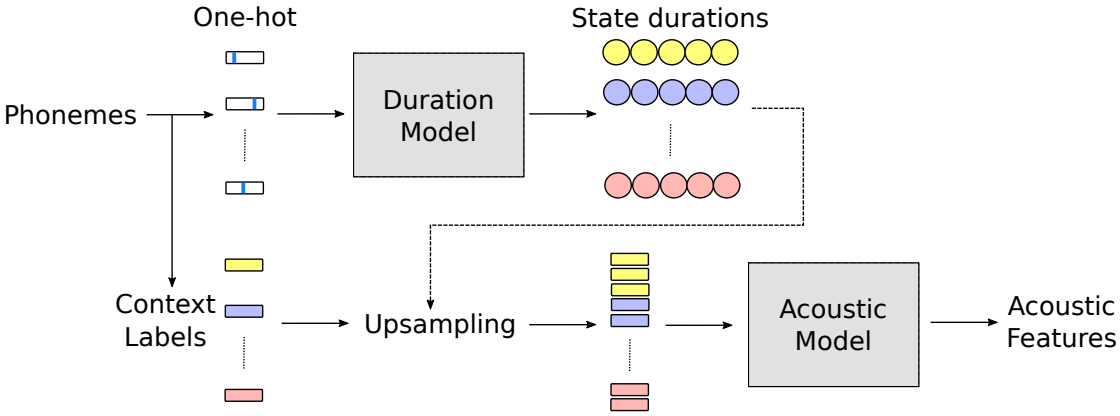


Figure 2.1 – Visualisation of old paradigm TTS pipeline.

2.3.4 Tacotron - The era of encoder-decoder models

With the publication of Tacotron (Wang et al., 2017b) in 2017 the era of encoder-decoder models was heralded. It was the first attempt to end-to-end speech synthesis combining the duration and acoustic model as well as the MLPG step (Figure 2.2). In contrast to the old paradigm it predicts mel-spectrogram and uses the Griffin-Lim algorithm (Griffin and Lim, 1984) to generate the waveform. Griffin-Lim has its problems and the success of the new paradigm was significantly boosted by neural vocoders, which effectively convert mel-spectrum to waveform. Neural vocoders are discussed in the next section.

Instead of context embeddings phoneme or even character embeddings are used. The encoder network consists of multiple convolutional layers and a bidirectional recurrent layer. With the receptive field of the Convolutional Neural Network (CNN)-stack and the recurrence it is expected that the encoder learns the context embeddings on its own, which leads to better embeddings than hand-crafted features.

The duration prediction is taken care of by an attention mechnanism. It also uses recurrence and state information inside and is computed for each decoder step. The model learns the alignment on its own, thus no forced alignment is necessary. With the attention mechanism

the controllability is partly lost. It can also lead to repetitions and skipping of words or entire utterances.

The decoder is an autoregressive model with unidirectional recurrence. It predicts a chunk of frames at the same time (usually five) and feeds the last predicted frame back as autoregressive input. Before being concatenated with the attention context the autoregressive input is transformed by a pre-net. At training time the model is trained with teacher forcing (the oracle features are used as autoregressive input). This allows parallel training and the simple computation of L1 loss on the prediction, because the alignment matches. The decoder also predicts a stop token. It indicates that the synthesis as finished and that the autoregressive model should stop.

The decoder output is passed through a post-net consisting of 2D convolutional layers, which predicts an additive residual. This step smooths the prediction similarly to the MLPG step in the old paradigm. It can also add some finer details to the spectrum. When the model is used with the Griffin-Lim algorithm the post-net also converts from mel to linear-frequency scale (no residual in this case).

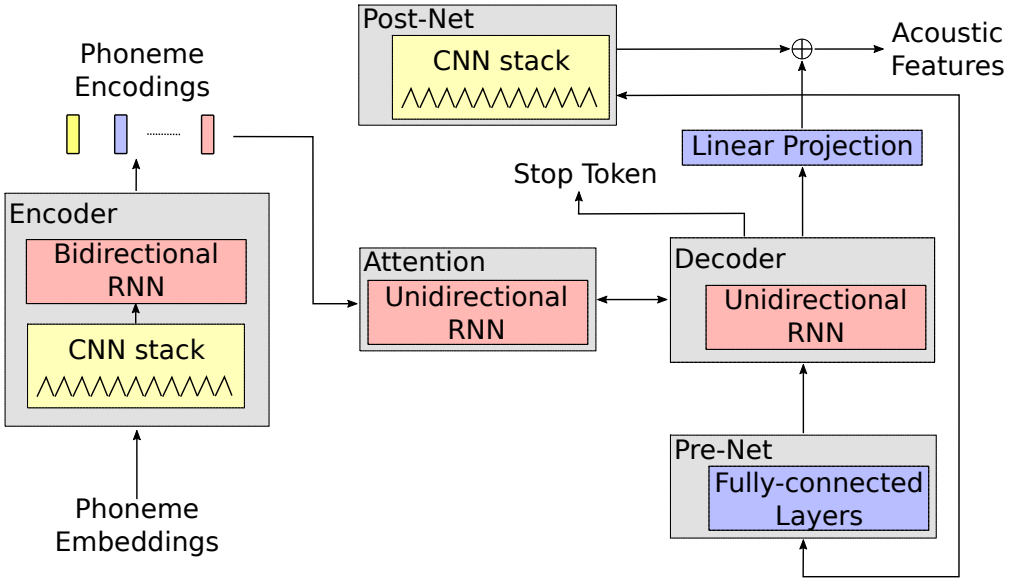


Figure 2.2 – Visualisation of new paradigm TTS pipeline.

2.3.5 Neural vocoder

WaveNet (Figure 2.3, van den Oord et al. (2016)) was published in September 2016 and started the advent of neural vocoders substituting previous vocoders based on signal-processing techniques like WORLD and STRAIGHT. WaveNet is an autoregressive network based on gated dilated convolutional layers. In each iteration it predicts the next sample of the waveform based on a receptive field over the previous predictions. Though less useful, the model can be run without conditioning, which allows infinite sampling of speech-like babbling from it. In

the original publication the model is conditioned on a global speaker embedding and context embeddings with LF_0, which are upsampled by deconvolution layers to match the sampling frequency of the target waveform. In this case it operates as an end-to-end TTS model. It was later used as a neural vocoder only by using a conditioning on mel spectrogram (Shen et al., 2018). The model is trained in teacher forcing mode, which brings efficient training speed. At inference time the model runs heavily sequential making it slow. This makes it unusable in production TTS systems and is its main shortcoming.

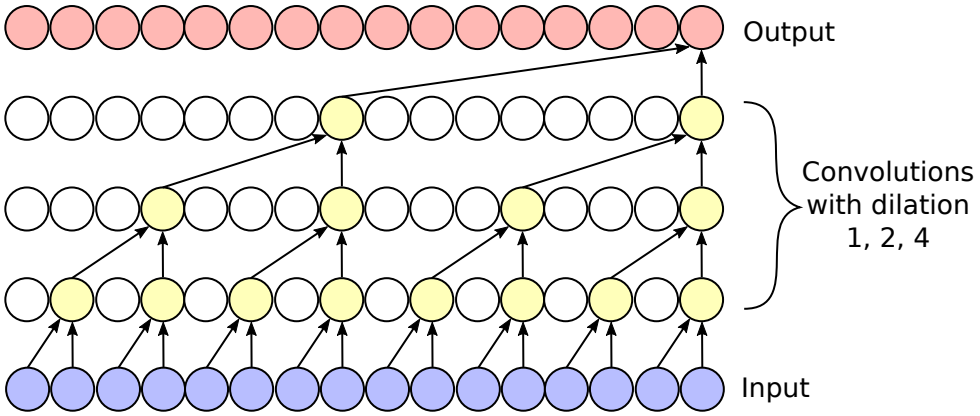


Figure 2.3 – Visualisation of WaveNet's receptive field when predicting the next waveform sample. The inputs are the previously predicted samples at inference time or the oracle waveform at training time. The image does not show the conditioning, which is concatenated to the input of every convolutional layer, and the aggregating network with residual connections.

Some architectures were proposed to overcome this limitation. WaveRNN (Kalchbrenner et al., 2018) is a Gated Recurrent Unit (GRU)-based network with reduced computational cost thanks to sparse matrices. FFTNet (Jin et al., 2018) is a significantly smaller version of WaveNet, that is able to achieve the same quality because it models the Fast Fourier Transform (FFT) with suitable network components. It is designed to transform F_0 and mel Frequency Cepstral Coefficients (MFCC) to the waveform and requires a post-processing step. WaveGlow (Prenger et al., 2019) is a non-autoregressive network based on normalising flows, which is parallelisable because it is entirely feed-forward.

Neural vocoders based on signal-processing techniques

With increasing interest in neural vocoders, some models were developed, that overcome the speed limitation by incorporating signal-processing knowledge. The idea is that many concepts of speech are well understood, like linear prediction or harmonic sine waves, but are difficult to model by conventional NN components.

Wang et al. (2019a) propose a set of neural source filter models. The models use a source module to produce a sine-based excitation signal. The excitation signal is then transformed into the waveform with a filter module. A conditioning module transforms the inputs to the

source and filter model. Their best model also incorporates a harmonic-plus-noise structure where transformed Gaussian noise is mixed with the filter module output in a ratio based on whether the frame is voiced or not.

LPCNet (Valin and Skoglund, 2019) enhances WaveRNN with an explicit Linear Predictive Coding (LPC) block. Based on the previous samples it computes the expected next sample and feeds it to the network. The RNN then only has to add the residual excitation to it.

GlottNet (Juvela et al., 2019a) predicts the glottal excitation signal from acoustic features in an autoregressive manner. It has a similar structure as WaveNet but is much smaller, because it only has to predict the glottal excitation. The excitation is filtered by a vocal tract filter, whose parameters are computed from the acoustic features, to generate the waveform.

Parallel WaveNet

Parallel WaveNet (Oord et al., 2018) is an inverse autoregressive flow (Kingma et al., 2016) that overcomes the speed limitations of WaveNet with nearly equal audio quality. An inverse autoregressive flow converts a base distribution to the target distribution by a chain of bijective transformations. The probability density of one sample is conditioned on the base distribution up to that sample. As the prediction is deterministic it implicitly allows the model to internally know what it had predicted before without explicit autoregressive input. At inference time the transformation can be computed in parallel when the entire input sequence is available. However, during training the gradient of the N-th input is required to compute the gradient of the $N+1$-th input. This makes training sequential and increasingly time consuming, especially because Parallel WaveNet is a much bigger network than WaveNet.

Parallel WaveNet uses logistic noise as base distribution which is close to the mixture of logistics generated by WaveNet. The Parallel WaveNet (student) predicts shift and scale for every sample based on the logistic input noise. The sample value is computed by scaling the input noise and adding the shift. The predicted waveform is given to a pre-trained WaveNet (teacher), which predicts the most likely mixture of logistics for every sample given all previous samples. The main optimisation loss is the KLD between the teacher and student distributions. Three other auxillary losses further improve the results: the power loss (Mean-Squared-Error (MSE) of power spectrum), a contrastive loss (maximised KLD when student and teacher are conditioned differently), and a perceptual loss (accuracy of ASR system).

2.4 Audio Quality Measures

To evaluate a TTS system clear measures/metrics are needed. The measures used in today's TTS research can be split into objective and subjective measures. Objective measures are based on mathematical functions and clearly defined. Subjective measures are based on the ratings of human listeners. Based on the task the listeners have to be trained or require a certain level of language fluency or even nativeness.

2.4.1 Objective measures

F_0 is often computed in terms of Root-Mean-Square Error (RMSE) on voiced frames. Errors in voicing are captured by the percentage of wrongly (un)voiced frames.

The error in the cepstrum is computed by mel-cepstral Distortion (MCD) which is simply an MSE on each dimension independently excluding energy (c_0). When ground truth durations are not used the prediction first needs to be aligned with the target. Dynamic time warping is usually used to align the sequences. It is chosen so that the resulting MCD is minimal.

While those metrics give rough estimates of the synthesis quality, they were found to not reflect the ratings of humans for modern TTS systems. The quality of today's systems is too high to be differentiated by simple metrics including PESQ (Rix et al., 2001).

Some works compare TTS systems in terms of recognition performance of other machine learning algorithms, e.g. ASR, emotion recognition, and speaker identification systems. Some NN even attempt to predict the ratings of human listeners from audio (Lo et al., 2019). However, a convincing measurement has not been found so that every system has to be validated by subjective measures.

2.4.2 Subjective measures[1]

Mean Opinion Score

Mean Opinion Score (MOS) is a commonly used test for audio quality. The listener is presented with a single sample at the time and has to rate it on a 1 to 5 scale. The step size is either 1 or 0.5. While MOS tests are simple they require listeners with a certain level of expertise. Because there is no anchor in the test, listeners might rate the best system as 5 and the worst as 1. MOS scores can only be compared when the same set of listeners was used. While it is still used much in the literature we usually recommend a MUSHRA test.

MUSHRA

The MUltiple Stimuli with Hidden Reference and Anchor (MUSHRA) test was designed to compare the audio quality of audio codecs (Series, 2014). In contrast to MOS tests it presents samples of all systems at the same time. This allows to get statistically significant results based on a paired t-test with much fewer participants. In the test a reference audio is presented as groundtruth. Then the listener has to rate all the systems under test on a scale from 0 to 100 on a given criterium (mostly audio quality). The reference also reappears between the systems and the listener is asked to rate it at 100 points. Excluding tests where the listener has rated the reference below 90 points is a valid strategy to exclude outliers. The order of the presented systems is randomised. Between the systems also a hidden lower anchor exists. It

[1] Listening tests are designed with the BeaqleJS toolkit (Kraft and Zölzer, 2014).

should be a system which is clearly worse than all the systems under test. The hidden lower anchor causes the ratings to be closer together. It mitigates the problem of MOS tests where some participants rate the worst system with the lowest possible score.

2.4.3 Preference test

In a preference test the listener chooses which of the two options better matches a certain criterion. This can be audio quality (even though a MUSHRA test is better suited), speaker similarity, expressiveness, and others. When a reference sample is given as groundtruth to compare against, we speak about an ABX test; if not, it is an AB test. The order of the systems is random so that the participants do not know which of the provided audio came from which system. Sometimes the listeners can also select that they do not prefer either of the two systems.

2.5 Databases

2.5.1 2008 Blizzard Challenge

The speech database released for the 2008 Blizzard Challenge (Karaiskos et al., 2008) consists of the native UK English voice A (Roger) with about 15 hours, UK English voice B with about one hour, and a Mandarin voice C with about 6.5 hours of recordings on a 16 kHz sampling rate. In this work only voice A is used which contains parts of Lewis Carroll's children's stories (Strom et al., 2006) (subset carroll), the Arctic corpus (Kominek et al., 2003) (subset arctic), and newspaper text (subset theherald 1, 2, 3). Additionally it has emphatic recordings (subset emphasis), world list recordings (subset worldlist), and carrier sentences (subset address and spelling).

2.5.2 VCTK

The CSTR VCTK database (Veaux et al., 2017) consists of 110 English speakers with different accents. The recordings are at 48 kHz and have ~400 utterances per speaker. The 400 sentences are from a newspaper, the rainbox passage, and an elicitation paragraph. Only the newscaster sentences vary between the speakers and were selected based on a greedy algorithm for contextual and phonetic coverage.

2.5.3 WSJCAM0

The WSJCAM0 (Fransen et al., 1994) database forms a UK English equivalent of a subset of the US American English WSJ0 database (Paul and Baker, 1992). It consists of 92 speakers reading 90 non-parallel speaker-independent utterances (training subset) and additional 48 speakers reading 40 sentences containing only words from a fixed 5000 word vocabulary and

40 sentences from a fixed 64000 word vocabulary. Another 18 sentences are shared between all speakers. A head-mounted close-talking and a far-field desk microphone were used for the recordings at 16 kHz. In this thesis we use only the head-mounted close-talking microphone recordings.

To compensate for loudness differences we use a loudness normalization technique (Equation 2.1) to normalize all samples to an average root-mean squared value of $RMS = 0.1$.

$$\tilde{x} = x * \sqrt{\frac{T * RMS^2}{\sum^T (x - x_{mean})^2}} \tag{2.1}$$

We also found a significant amount of background noise in some of the recordings. To reduce the noise we use a single channel spectral enhancement scheme (Cauchi et al., 2015) to pre-process the entire database.

2.5.4 SAVEE

The Surrey Audio-Visual Expressed Emotion (SAVEE) database (Haq et al., 2008) is an audio-visual British English database with sentences from TIMIT phonetically-balanced for each emotion. For each emotion three common, two emotion-specific, and ten generic sentences (different for each emotion) were taken from TIMIT. For neutral the three common and $2 * 6$ emotion-specific sentences were additionally recorded, giving 30 neutral sentences in total. Four male (postgraduate students and researchers) acted in seven different emotions (neutral, anger, disgust, fear, happiness, sadness, and surprise) resulting in a total of 480 utterances. The audio was recorded at 44.1 kHz. The database also contains visual and 3d dynamic face capture data, which we do not use in our experiments. The recordings of speaker 'KL' are significantly quieter than those of the other three speakers which can have a negative effect on the training of a TTS system. Thus we use the same loudness normalisation and background noise reduction technique as on WSJCAM0 (compare Section 2.5.3).

2.5.5 IEMOCAP

The Interactive Emotional Dyadic Motion Capture (IEMOCAP) (Busso et al., 2008) database is a commonly used database for speech emotion recognition (Mirsamadi et al., 2017; Ramet et al., 2018; Tarantino et al., 2019). It splits into five dialogue sessions of acted and spontaneous emotions with two different professional actors each, totalling ten speakers and approximately 12 hours of 48 KHz recordings. At least three fluent English speakers annotated the perceived emotion and the final emotion label was chosen based on majority vote. We apply the same loudness normalization and noise reduction techniques as on WSJCAM0 (see Section 2.5.3).

3 A Neural Generalised Command Response Model

Many aspects of speech control its affect such as duration, intonation, and frequency patterns. In our endeavour to improve affective speech synthesis we investigate aspects independently and try to improve perceived affect and/or their control. In this chapter we focus on intonation, i.e. pitch or fundamental frequency (F_0). Proper intonation like emphasis is important. For example, speech to speech translation requires transfer of paralinguistics from one language to another. If a speaker expresses emotion or emphasis in an input language, we would like those features to be present in the synthetic speech resulting from machine translation of speech recognition output.

In previous work at Idiap (Honnet et al., 2015), a model of prosody (actually intonation, F_0) based on the Command-Response (CR) model of Fujisaki et al. (1998) was studied. We give a detailed introduction to the CR model in Section 3.1.1. By contrast to the CR model, this *Generalised* Command Response (GCR) model can be extracted easily from an intonation contour using a matching pursuit algorithm (Mallat and Zhang, 1993). The time-local nature of its constituent *atoms* was shown (by design) to lend itself to transfer of emphasis. In particular, sections of intonation contours can be replaced with others that carry different meaning, all whilst retaining naturalness. A detailed explanation of the GCR model is given in Section 3.1.2.

In this chapter, we report on an investigation into how to use the GCR to generate longer intonation contours for more general models. Of course, such contours can be generated by any modern TTS system. However, we hope to retain the transfer capability of the GCR. The GCR also enables analysis of the underlying physiological process.

While the design of the GCR suggests the use of SNNs we instead emulate an SNN using a bidirectional RNN which is capable of generating spikes, hence atoms, for a given text. This allows us to rely on back-propagation training. It is a less risky task as we rely on a familiar technology. This work also forms a necessary precursor to the work of a new PhD student, who has recently started at Idiap, on the use of SNNs for ASR and TTS. The work additionally relies on an evolving number of training algorithms for SNNs with back-propagation proposed over the course of this thesis. It has already resulted in a publication (Bittar and Garner, 2021).

We show that a bidirectional GRU based RNN can simulate the spikes that might be expected to come from a biological spiking network. We introduce a loss function for the training of spiking outputs which is inspired by losses in SNNs. It generalises the Spike Pattern Association Neuron (SPAN) algorithm (Mohemmed et al., 2013) from the literature to construct a loss function from GCR atoms, which can be backpropagated. We test the hypothesis that prosody generated by our neural model is natural (objectively and subjectively), even though it might vary from the ground truth and that generated by a baseline model. While our first model relies on a post-processing step, we extend it by embedding neural muscle models (Section 3.1.3) and adding a phrase component modelling ability. The model is able to learn the muscle parameters and thus improves on the previous version in objective and subjective scores.

The majority of the text in this chapter was originally published as:

- Schnell, B. and Garner, P. N. (2018). A neural model to predict parameters for a generalized command response model of intonation. *Proc. Interspeech 2018*, pages 3147–3151 doi: /10.21437/ssw.2019-6
- Marelli, F., Schnell, B., Bourlard, H., Dutoit, T., and Garner, P. N. (2019). An end-to-end network to synthesize intonation using a generalized command response model. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7040–7044. IEEE doi: /10.1109/icassp.2019.8683815

3.1 Background

This section covers necessary background information for the research presented in this chapter. It starts with an introduction to the intonation model of Fujisaki et al. (1998), followed by the *Generalised* Command Response (GCR), which we model with an RNN in our research. We then introduce the neural filters previously developed at Idiap, which allow us to model the entire GCR with a single RNN. We give a rough overview over other research trends in intonation modelling and describe the main characteristcs of SNNs.

3.1.1 Fujisaki's Command Response Model

Fujisaki's Command-Response (CR) Model (Fujisaki et al., 1998) is one of the most well known intonation models. Fujisaki assumes the LF_0 contour is a superposition of three components:

1. A constant base component depending on the speaker, speaking style, and emotion.
2. A global phrase component associated with a slow movement related to the depleting lungs.
3. Multiple local accent components for fast adaptation of LF_0.

The model has been successfully applied to a wide variety of languages including but not limited to Japanese, English, German, Greek, Korean, Spanish, Mandarin, Thai, Swedish and Bengal. Phrase and accent components can be negative, which was necessary for tonal languages and languages with pitch accents. The preeminent aspect of the model is its physiological interpretability. Fujisaki was able to relate the time varying components to the movement of the muscles controlling the vocal folds. In the following we will give the mathematical description of the CR model.

The constant base can be treated as a simple bias of the form $\ln F_b$. The time varying components differ in their activation. The phrase component is caused by impulses (phrase commands) while the accent components are represented as stepwise functions (accent commands). The phrase component is the impulse response of a second-order, critically-damped linear filter:

$$G_{pi}(t) = \begin{cases} \alpha_i^2 t e^{-\alpha_i t} & \text{if } t \geq 0 \\ 0 & \text{if } t < 0 \end{cases} \tag{3.1}$$

α_i^2 is the natural angular frequency of the ith phrase control mechanism G_{pi}. The accent components are the response of a stepwise accent command of a second-order, critically-damped linear filter:

$$G_{aj}(t) = \begin{cases} \min\left[1 - \left(1 + \beta_j t\right) e^{-\beta_j t}, \gamma\right] & \text{if } t \geq 0 \\ 0 & \text{if } t < 0 \end{cases} \tag{3.2}$$

β_j^2 is the natural angular frequency of the jth accent control mechanism component G_{aj}. γ is the maximum threshold, which Fujisaki set to 0.9. The LF_0 contour can be represented as:

$$\ln F_0(t) = \ln F_b + \sum_{i=1}^{I} A_{pi} G_{pi}\left(t - T_{0i}\right) + \sum_{j=1}^{J} A_{aj} \left\{G_{aj}\left(t - T_{1j}\right) - G_{aj}\left(t - T_{2j}\right)\right\} \tag{3.3}$$

I is the number of phrase components and J the number of accent components. A_{pi} and A_{pj} are the magnitudes of the ith phrase and jth accent comand respectively. T_{0i} is the time of the impulse of the phrase command i. T_{1j} and T_{2j} are the onset and offset times of the accent command j.

It is possible to extract the parameters of the CR model from audio. Several methods have been proposed including those of Agüero and Bonafonte (2005); Agüero et al. (2004); Kameoka et al. (2010); Mixdorff (2000); Narusawa et al. (2002) and more recently Torres and Gurlekian (2015). Generation of the model parameters for TTS is more difficult but was successfully implemented in the framework of HMMs in Kameoka et al. (2015).

3.1.2 Generalised Command-Response Model

Based on Fujisaki's CR model a *Generalised* Command Response (GCR) model was developed in previous work at Idiap (Honnet et al., 2015; Gerazov et al., 2015) which has the same representative power. The main goal of the GCR was to develop a physiologically plausible intonation model based on muscle responses. At its core lies a muscle response to a nerve impulse, mathematically similar to the phrase component and phrase commands in the CR model. From this command-response perspective muscles are triggered by the brain through nerve impulses. Each nerve impulse creates a muscle twitch (Figure 3.1) and higher muscle activations are caused by sequences of impulses with a period shorther than that of the twitch. Honnet et al. (2015) argued that the step function accent commands of the CR model can be modelled by a train of spikes where the spikes are separated by only one frame. This allows to use the same type of critically-damped linear filters for the accent commands as for the phrase command.

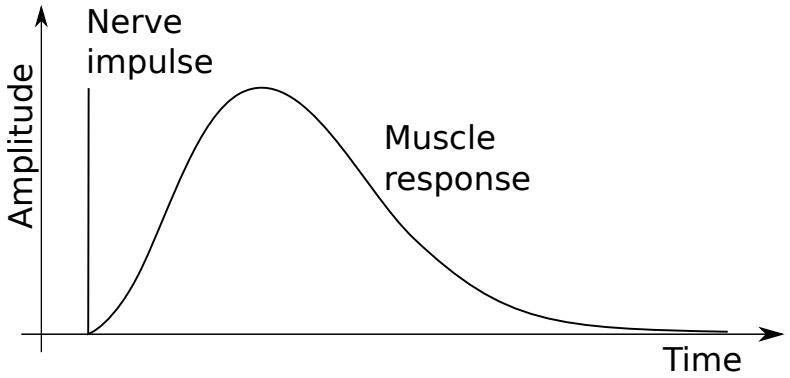


Figure 3.1 – Muscle response to a nerve impulse. Image taken from Honnet (2017) Section 5.3.2

To model the impulse response of a muscle the gamma kernel was proposed:

$$G_{k,\theta}(t) = \frac{1}{\theta^k \Gamma(k)} t^{k-1} e^{-t/\theta} \quad \text{for} \quad t \geq 0 \tag{3.4}$$

with k being the system order, Γ being the gamma function, and θ determining the length of the kernel. $k = 2$ for a critically damped second-order system as well as the CR model, however, previous research (Gerazov et al., 2015) has found that $k = 6$ gives better approximations of the original LF_0 contour.

To approximate the LF_0 curve with a superposition of kernel functions the matching pursuit algorithm (Mallat and Zhang, 1993) was proposed. It uses a fixed length dictionary of kernel functions, which we will refer to as **atoms** from here on. The iterative algorithm finds an atom and its weight that correlates best with the signal. It then subtracts the chosen atom from the signal and continues with searching for the next best atom. The algorithm ends when a certain accuracy is achieved or a maximum number of atoms is reached. The matching pursuit algorithm finds the atoms, which allows direct computation of the impulse positions, however,

each impulse has an amplitude/weight and the conversion to a sequence of impulses is non-trivial. Therefore the GCR assigns an amplitude to each spike instead of using a sequence of unit size spikes. The result of the matching pursuit algorithm can be seen in Figure 3.2. More atoms could be extracted to decrease the error between reconstruction and original LF_0.

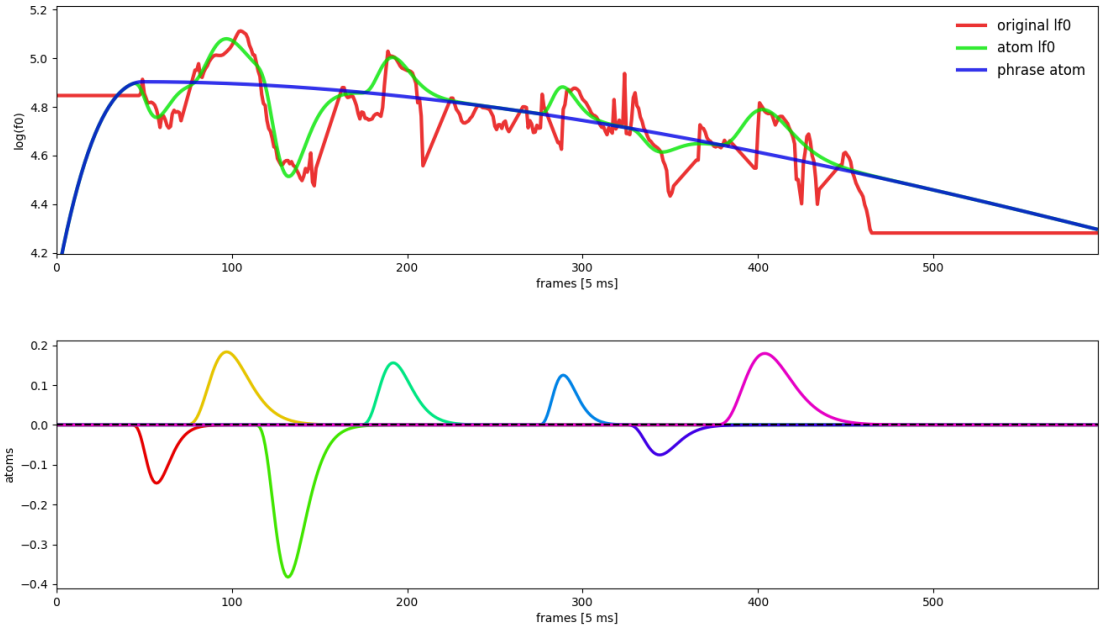


Figure 3.2 – Atom decomposition of LF_0 contour. Upper plot: Original LF_0 contour (red), reconstruction from atoms (green), phrase component (blue). Lower plot: Atom impulse responses with one colour per atom.

Classical matching pursuit does not take into account the interpolated LF_0 of unvoiced regions and will try to follow the signal closely in them as well. Thus Gerazov et al. (2015) proposed to use a weighted RMSE as cost function within the matching pursuit algorithm. It assigns a weight to the whole signal based on the multiplication of voicing probability $p(t)$ (Ghahremani et al., 2014) and energy $e(t)$. This encourages the algorithm to ignore errors in unvoiced regions. Their algorithm also skips over silence in the beginning and end of the signal.

The GCR also modifies the phrase component to use the same Gamma kernel. It is mainly correlated to the physics of the speaker's lung volume. It was necessary to use a constant θ_r during the rise and a variable θ_f for the fall. The phrase component is estimated and subtracted from the signal before running the matching pursuit algorithm.

$$G_{k,\theta_r,\theta_f}(t) = \begin{cases} \frac{1}{\theta_r^k \Gamma(k)} t^{k-1} e^{-t/\theta_r} & \text{for } 0 \leq t \leq t_{rm} \\ \frac{1}{\theta_f^k \Gamma(k)} t'^{k-1} e^{-t'/\theta_f} & \text{for } t > t_{rm} \end{cases} \tag{3.5}$$

$$\begin{aligned} t_{rm} &= (k-1)\theta_r \\ t_{fm} &= (k-1)\theta_f \\ t' &= t - (t_{rm} - t_{fm}) \end{aligned} \tag{3.6}$$

Experiments have shown that the proposed model is capable of producing good representations in English, German, and French. Until now it was not possible to use the GCR for TTS (Honnet, 2017). Previous attempts used only non-recurrent NNs, which were state of the art at that time.

However, the GCR was successfully used to transplant emphasis from one to another language (Honnet and Garner, 2016). Atom parameters are clustered with a random forest algorithm based on contextual factors on the word-level. Observations showed that a maximum of five atoms are required for an emphasised word. In a first step a TTS system synthesises the neutral version of the sentence. Then matching pursuit is used to extract its atoms. From the random forests the atoms of the emphasised word are obtained and substitute the neutral atoms of that word. The method greatly improved the perceived emphasis.

3.1.3 Neural Filters

As part of a Master Thesis at Idiap Marelli (2018) has developed a neural implementation of a Spring-Damper-Mass (SDM) system. The filter was designed to substitute the muscle response behaviour in the GCR model and also matches Fujisaki's assumption that command signals are filtered by second order systems. The filter itself is similar to a second-order Infinite Impulse Response (IIR) synapse, which has been analysed before (Back and Tsoi, 1991; Campolucci and Piazza, 2000). He extended that analysis by investigating and solving the gradient explosion issues that can prevent the convergence of recurrent units (Pascanu et al., 2012), when applying back-propagation through time (Werbos, 1990). Parts of this section can be found in the collaborative publication (Marelli et al., 2019), however, it must be noted that the content of this Section 3.1.3 was developed by Marelli and is repeated for context here. The collaborative work is limited to the integration of the neural filters in the TTS system discussed in Section 3.4.

The generic discrete transfer function of an SDM system is

$$y_{(k)} = Gx_{(k)} + \alpha y_{(k-1)} + \beta y_{(k-2)} \tag{3.7}$$

with x the command signal, y the response, G the gain, α and β the recurrence coefficients of the model, and k the discrete time step. It is equivalent to the equation of a second-order linear all-pole digital filter (Figure 3.3).

Given that the SDM system is second-order time recurrent, it would make sense to model it by an RNN. Implementations such as Long Short-Term Memorys (LSTMs) or GRUs can indeed

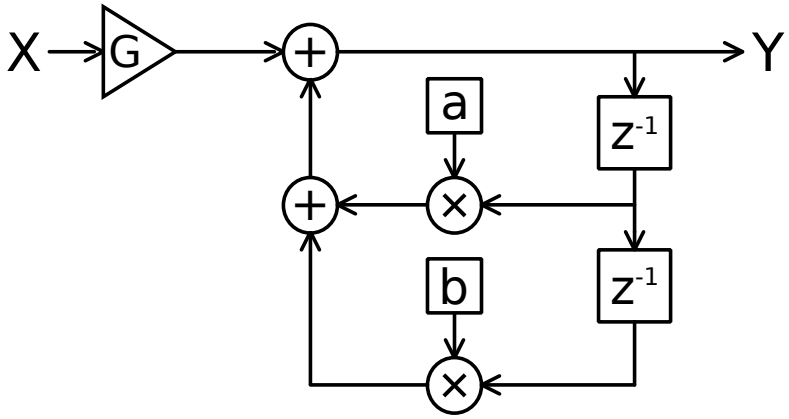


Figure 3.3 – Second-order linear all-pole digital filter.

learn second order recurrence. Nevertheless their behaviour is strongly non-linear and they are over-parametrised for an SDM model. Therefore he proposed a simpler implementation of linear second-order recurrent units based on (3.7).

The expectation is that gradient descent optimisation can be used for the extraction of the filter parameters, as iterative methods have proven to be efficient for digital filter design problems (Howell and Gordon, 2001). However, gradient explosion issues are inherent to recurrent units training (Pascanu et al., 2012). To study the convergence properties of the model, he derived the gradient equations of (3.7) using back-propagation through time (Werbos, 1990). The obtained expressions are given in (3.8 – 3.10), with K_n defined in (3.11).

$$\frac{\partial y_{(k)}}{\partial \alpha} = \sum_{n=0}^{k-1} \left[y_{(k-1-n)} \cdot K_n \right] \tag{3.8}$$

$$\frac{\partial y_{(k)}}{\partial \beta} = \sum_{n=0}^{k-2} \left[y_{(k-2-n)} \cdot K_n \right] \tag{3.9}$$

$$\frac{\partial y_{(k)}}{\partial x_{(k-n)}} = G K_n \tag{3.10}$$

$$K_n = \begin{cases} \alpha K_{n-1} + \beta K_{n-2} & \text{if } n > 0 \\ 1 & \text{if } n = 0 \\ 0 & \text{if } n < 0 \end{cases} \tag{3.11}$$

The gradient explosion is caused by the recurrence in K_n. The analysis of (3.11) reveals that a sufficient condition to prevent gradient explosion is that all the poles of the model have a modulus lower than one (i.e. the modelled system stability implies the stability of the gradient). When targeting only complex conjugate pairs the model can be expressed in polar notation

$$y_{(k)} = G x_{(k)} + 2\rho \cos(\phi)\ y_{(k-1)} - \rho^2\ y_{(k-2)} \tag{3.12}$$

with ρ the modulus and ϕ the phase of the poles. This assumption is not a limitation for muscle modelling, as muscle responses tend to behave as under-damped or critically damped systems (Gerazov and Garner, 2016). This, in turn, allows to express the stability constraint as (3.13), which can be imposed by using a compressing transformation (Campolucci and Piazza, 2000) as sigmoid (3.14). The cosine of the pole angle can also be transformed to use the whole parameter space by defining c in (3.15).

$$|\rho| \leq 1 \tag{3.13}$$

$$\rho = \sigma(p) \tag{3.14}$$

$$\cos(\phi) = \tanh(c) \tag{3.15}$$

The reformulation of the system

$$y_{(k)} = Gx_{(k)} + 2\,\sigma(p)\,\tanh(c)\,y_{(k-1)} - \sigma^2(p)\,y_{(k-2)} \tag{3.16}$$

is used to implement trainable muscle models integrated in neural networks. The network simply has to learn p and c and the gradient explosion issues are prevented. Marelli (2018) also investigated the vanishing gradient problem and could show that using the Adam (Kingma and Ba, 2015) optimiser efficiently prevents it.

Toy experiments were conducted to demonstrate the modelling capabilities of the proposed neural filter. A database of 500 white noise samples of 200 frames was filtered by the target filter and white noise was added. 80% of the database was used to train a neural filter. The learned filter parameters indeed approached the target filter parameters with sufficient training steps for a wide range of filter parameters. It was also tested whether the system can learn a superposition of two filters. For that the white noise samples were processed by two filters, then summed, and random noise was added afterwards. The experiments showed that is was indeed possible to learn the filter parameters of both filters.

3.1.4 Related Work

In this section we will list related works of intonation modelling. One can roughly separate the research into two categories: First, the models of superposition, which describe the intonation contour as a superposition of functions. Those functions arise from lingustic or physiological events. Second, pure Deep Neural Network (DNN) methods which assume that relevant patterns are extracted from sufficient training data. We further split DNN models into models which predict only F_0 and those which provide full TTS where the intonation features only appear in latent representations within the network.

Superposition Intonation Models

Numerous approaches to modelling prosody by the superposition of multiple F_0 contours exist. E.g. the General Superposition Model of Intonation (Van Santen and Möbius, 2000) models the pitch contour through a decomposition of microprosodic segmental perturbations, an accent and a phrase curve.

The Tilt model (Taylor, 2000) describes the pitch contour as a sequence of events with specific shapes that can be automatically extracted. The model produces a sequence of Tilt components with a position, amplitude, and a Tilt parameter describing the shape of the component. The Tilt parameter specifies the rise and fall shape of the event in the range of $[-1, 1]$, where 0 indicates a symmetric shape. F_0 is reconstructed by adding the Tilt shapes and applying linear interpolation between them.

Hirst and Espesser (1993) create a smoothed macromelodic intonation curve by approximating F_0 with a (piece-wise) quadratic spline function. The approximation is chosen so that no value of the original curve is more than a specific threshold above it. The authors assume that all microprosodic effects consist of lowering the macroprosodic curve. "Interesting" tonal segments are extracted based on an alphabet of 8 symbols, consisting of absolute tones (top, mid, and bottom), relative tones (higher, same, and lower), and iterative relative tones (upstepped or downstepped). This labelling system is called the INTSINT (INternational Transcription System for INTonation) model (Hirst et al., 2000). It also allows automatic parameter extraction of the Tone and Break Indices (ToBI) model (Silverman et al., 1992) that divides prosody into multiple tiers of linguistic focus.

The Superposition of Functional Contours (SFC) model (Bailly and Holm, 2005) is a data driven approach that models the pitch contour by a superposition of intonation prototypes. A NN learns a set of function contours (prototypes) based on metalinguistic information. Once trained, an intonation contour can be decomposed into these contours. The shape of the prototypes is not restricted and entirely learned from the data.

A common drawback of all models above is that none of them is based on observations of the physiological production aspect. The work closest to our approach is that of Hojo et al. (2017) where the CR model is represented by a constrained HMM, and a DNN predicts the posteriori probability of its states. A Viterbi-like algorithm extracts the most probable sequence based on the posteriors. The LF_0 generation based on the sequence is then straight forward and has been done before (Kameoka et al., 2015).

Neural Intonation Models

With the advent of DNNs in all speech related research areas many neural intonation models were developed, starting already in the 90s (Traber, 1991; Chen et al., 1998). While they are less relevant for the work in this chapter, we give a quick overview over recent techniques. The first DNN models proposed, predict only LF_0 from textual or other context features. Newer

methods are integrated in the TTS pipeline and predict some latent representation of prosody as an intermediate representation within the neural network. We will list work in both domains in the following.

One of the first approaches was Fernandez et al. (2014). The authors use a bidirectional LSTM to model the long and short dependencies in F_0 prediction. Their model outperforms non-recurrent networks in subjective listening tests.

A major problem of prosody generation on expressive speech is the averaging effect. Prosody generation is an ill-posed problem, in the sense that the same sentence can have multiple plausible intonation curves, which depend on the desired meaning. When training a model on a database with the different intonation curves, it will learn to achieve the lowest MSE on all of them by predicting an average kind of prosody. This average prosody is usually perceived as flat, boring, and also unnatural by human listeners.

Hodari et al. (2019) proposed to use a VAE-based generative model to address the averaging effect. The VAE places the most "average" representation close to the mean of the Gaussian prior. By sampling towards the tails of the prior distribution, it is possible to generate varying and plausible intonation curves. The network is trained with F_0 and linguistic features as input in an encoder-decoder structure with the VAE at the bottleneck between them. The decoder is also conditioned on the linguistic features. At inference time either the prior mean or the tails from the distribution are used. Subjective tests proved that the sampling from the tails produced more varied intonation while maintaining the same level of naturalness.

Wang et al. (2017a) proposed an autoregressive network with multiple levels of feedback. This is achieved by summarising previously predicted values over linguistic segments (frames, phonemes, and syllables) with a moving average. Instead of LF_0, which is artificially interpolated in unvoiced regions, the model predicts quantised F_0 steps including an *unvoiced* symbol. The model is trained with a hierarchical softmax layer which takes into account the special characteristics of the unvoiced symbol. The model produces shapes close to human production. The model was further improved in Wang et al. (2019b) by using a Vector-Quantised Variational Autoencoder (VQ-VAE) where the latent vectors operate on the phoneme level. As the model learns a codebook of F_0 events it can be seen as a modern approach to tilt events.

Kenter et al. (2019a) generate duration, as well as energy c_0 besides F_0. Similar to Hodari et al. (2019) they use a VAE to obtain sentence-level prosody embeddings. Additionally, they propose a hierarchical model which has layers operating at different frequencies. On the encoder side one RNN is used to aggregate frame-level features to syllable-level features, another to do the same from phoneme-level features. The results are concatenated with upsampled word- and sentence-level features before fed into another RNN. The decoder operates similarly in reverse order. The model is able to transfer prosody from a source sentence but also produce a variety of valid contours thanks to the VAE.

As part of the complete TTS pipeline Stanton et al. (2018) extend Tactron GST (detailed description in Section 4.1.2) with a GST predictor from textual features. They propose two models, where the first predicts the weight of the learned GSTs, while the second predicts one style embedding directly. Listeners did not prefer one over the other. The model is able to learn speaking styles when trained on a 147 hours single speaker database of English audiobooks.

Hodari et al. (2021) suggest to use suprasegmental context information to predict latent word-level prosody representations. Those representations are first learned with an auto-encoder with temporal bottleneck based on words. Then a prosody predictor is trained to predict them from text-derived features. The word-level prosody representations were also found helpful as input to an external duration model.

3.1.5 Spiking Neural Networks

Spiking Neural Networks (SNNs) are built out of neurons which attempt to model biological neurons as closely as possible. Information travels within the network in the form of spike trains with equal amplitude. Neurons of SNNs can perform multiple operations depending on the incoming spike trains (Kouh and Poggio, 2008), which makes them more flexible compared to their counter parts in artificial NNs. Based on the spiking nature of the model stimuli can be modelled as complex spatio-temporal spike patterns or synchrony on a set of neurons in the network, which is impossible in classical artificial NNs.

Spiking neurons rest at a constant membrane potential which is lowered or raised by incoming spikes, but always decays back to its resting potential over time. When enough incoming spikes raise the potential above a threshold, the neuron fires a spike and enters a phase of refractoriness before it accepts incoming spikes again. A well known implementation is the leaky integrate-and-fire model, which is similar to a simple electrical circuit with a capacitor in parallel with a resistor, which is in line with a battery, driven by a current. Spikes are received through step functions in the current.

While SNNs offer a set of desirable properties, training them remains challenging. The classical stochastic gradient descent relies on gradients, which cannot be obtained easily due to the non-differentiable nature of spiking neurons. A range of alternative learning approaches were developed in recent years (Bodyanskiy and Dolotov, 2013; Bengio et al., 2015; Bellec et al., 2018; Wu et al., 2020; Comsa et al., 2020). However, the computational costs of these models forces small network complexity far off from conventional artificial NNs, and even further from the human brain they try to resemble.

3.2 Atom Prediction

Given that GCR atoms approximate (groups of) muscle responses to neural spikes, it would make sense to use an SNN to generate these atoms. The generated spikes would be filtered

by muscle responses to generate the pitch contour. However, the choice of a spiking network paradigm is not obvious. Rather than use an explicit spiking paradigm such as leaky integrate and fire, we instead emulate such a network using a conventional backpropagation network, given the authors' familiarity with conventional back-propagation based deep learning algorithms and toolkits. In this work we use a bidirectional RNN which is capable of generating spikes, hence atoms, for a given text. This in turn allows us to introduce a loss function for the training of spiking outputs in this section which is inspired by losses in SNNs. Furthermore, we explain how to weight the loss of different frames with respect to spike positions and V/UV decision to achieve good generalisation.

3.2.1 Atom Loss

For a regression task, which targets spiking output of varying amplitudes, the commonly used MSE is not an appropriate loss function as it does not consider any temporal information of spikes. As an example, a spike which is shifted by a few frames away from the target spike results in the same error as a spike far away from any target spike. This misleading evaluation of error significantly worsens the ability of the NN to generalise. The problem breaks down to measuring the distance between two spike trains. Various methods exist to compute such a distance such as the Victor-Purpura metric (Victor and Purpura, 1997), the Van Rossum Similarity Measure (Rossum, 2001), the Schreiber *et al.* Similarity Measure (Schreiber et al., 2003), the Hunter-Milton Similarity Measure (Hunter and Milton, 2003), Event Synchronisation (Quiroga et al., 2002), Stochastic Event Synchrony (SES) (Dauwels et al., 2008), and the modulus- and max-metrics (Rusu and Florian, 2014).

In general we are interested in a learning rule that uses such temporally-aware measurements to compute losses during training. We have not found a suitable learning rule in the literature for feed-forward NN or RNNs but instead in the field of SNNs. The closest precedent to the learning rule we propose is the SPAN method (Mohemmed et al., 2013). In SPAN, each spike is convolved with an "alpha" kernel which adds temporal information of the spike to all surrounding/succeeding frames. On the resulting continuous output MSE can be used as the learning rule. The authors of the SPAN method state that other kernel functions are possible as Gaussian, linear, and exponential kernels (Ponulak and Kasiński, 2010) . The choice of kernel in the literature is driven by the supposed shape of the post-synaptic potential of neurons in the human brain. However, the spikes we are interested in represent muscle impulses with responses modelled by a gamma kernel as described in 3.1.2. We therefore use the gamma kernel as the kernel function. The length θ of the kernel is by no means obvious. While the desired length of correctly placed spikes is known, no ground truth is available for incorrectly placed spikes. We found that a single short kernel with $\theta = 0.01$ for all convolutions adds the required temporal information to each spike.

Let us define the matrix G which has the coefficients of the gamma kernel on its leading and above leading diagonals with size $(T \times T)$ where T is the number of frames in a training

sample. Further define y_o as the output of the NN and y_d as the desired output each of size $(T \times 1)$. All spikes can be convolved independently from each other with the kernel function by $\text{diag}(y) \cdot G = Y$ (compare Figure 3.4).

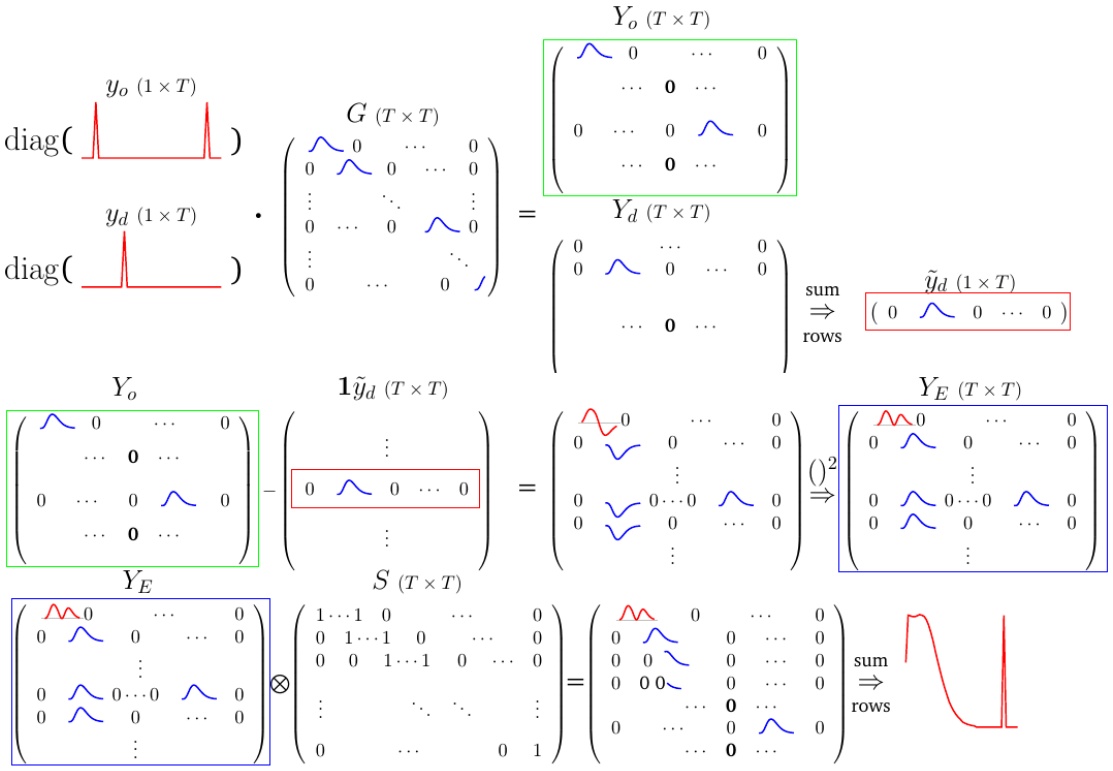


Figure 3.4 – Frame-wise convolution of NN output y_o and desired output y_d and the following steps to compute the frame-level error. The coloured boxes indicate reappearing components during the computation.

We denote $\tilde{y}_d$ as the desired enveloped output given by the sum of all rows of Y_d which corresponds to a superposition of envelopes. The error at each time step t is computed by

$$err_t = \sum_{i=t}^{t+\Delta t} (Y_{o,t,i} - \tilde{y}_{d,i})^2 \tag{3.17}$$

with $Y_{o,t,i}$ being the t-th row and i-th column of Y_o, and $\tilde{y}_{d,i}$ being the i-th entry in $\tilde{y}_d$. Δt is given by the length of the gamma kernel used to convolve each spike and represents the number of frames where a spike takes effect. To limit the interval of the sum to $[t, t+\Delta t]$ is critical so that the error is not affected by succeeding parts of the sequence where the spike cannot take effect. To compute the sum efficiently we define the matrix S of size $(T \times T)$ which is the same matrix as G but with ones at non-zero entries of G and zero otherwise. By utilising the Hadamard product $E = S \otimes Y_E$, with $Y_E = \text{square}[Y_o - \mathbf{1}\tilde{y}_d]$ and square[ ] being the element-wise square operation, entries outside the $[t, t+\Delta t]$ interval are zeroed. The error at

time step t is given by the squared norm of the t-th row of E.

$$err_t = \|E_t\|_2^2 \tag{3.18}$$

Note that the error is computed frame-wise without the superposition of the enveloped NN output, which means that neighbouring spikes cannot interfere. When allowing the interference of spikes two problems arise:

- The NN learns to represent a single target spike by multiple smaller spikes (Figure 3.5 left).
- The NN predicts many spikes with opposite amplitude which cancel out (Figure 3.5 right).

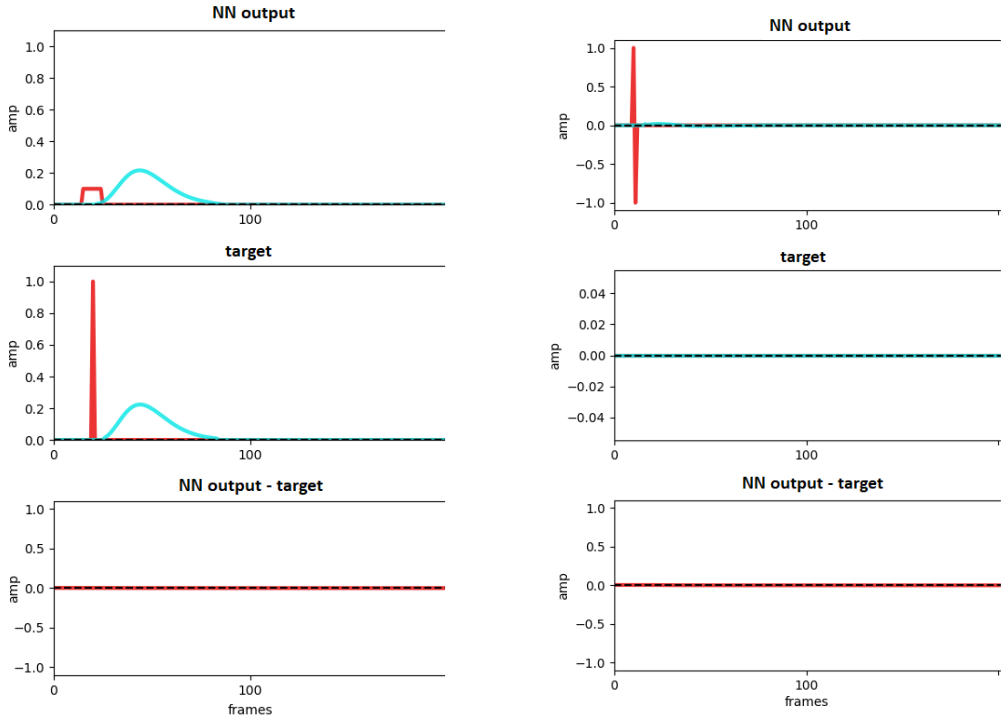


Figure 3.5 – Problems arising when allowing neighbouring spikes to interfere in the loss function.

The former problem is an acceptable variation to our model when assuming that a muscle response is not triggered by a single nerve impulse but multiple ones. However, the latter problem gives clearly unintended and physiological implausible behaviour. Therefore we use the frame-wise atom loss hereafter. A NN trained with the above learning rule gives an activation around each spike and thus requires a post-processing step to identify the peaks.[1]

[1] We use the `scipy.signal.find_peaks_cwt` function.

3.2.2 Amplitude Prediction

For any atom the prediction of position, amplitude and length θ is required. A single position flag trained with atom loss introduced above in Section 3.2.1 gives good estimates of the position of spikes ($y_{d,t} \in \{-1, 0, 1\}$) but cannot predict amplitude and length at the same time. Unfortunately, we were not able to train a NN to predict a θ value directly. The set of θs was found empirical in previous research and might not be optimal. This might hamper the NN to learn it from the data. Instead, besides the position flag, the NN is trained with MSE to predict one amplitude per θ for a fixed set of θs. The set of θs needs enough values to allow an approximation of the target LF_0 contour with low error, but is limited which corresponds to the limited number of articulators in the human larynx. When training on amplitude spikes the problem of a highly unbalanced training set arises (>99.8% of all frames are zero). A network trained with MSE will therefore uniformly predict zeros and achieve a >99.8% accuracy. The problem can be solved by small adaptations to data and loss. First each amplitude spike is convolved by a normal distribution in time with a window of 51 frames. Secondly the loss of frames which are non-zero in the desired output are increased while all others are decreased resulting in a Weighted Mean-Squared-Error (WMSE).

3.2.3 Voiced/Unvoiced Prediction

The network also predicts a flag for V/UV LF_0 where values >0.5 are mapped to voiced frames. The target V/UV flag is used to decrease the weight of both losses (atoms and amplitudes) by 0.5 on unvoiced frames. The value of 0.5 was confirmed by a heuristic search. By this the network spends less effort on improving parts which are silent after synthesis.

3.3 Experiments

In running experiments, we mean to test the hypothesis that the basic procedure described above is a plausible approach to generate natural sounding intonation. The system is preliminary. A-priori we do not expect it to generate state-of-the-art intonation contours; rather, we simply aim to validate that the approach produces plausible intonation contours while retaining the physiological interpretability of the GCR model.

3.3.1 Experimental Setup

We test our proposed model on the speech database released for the 2008 Blizzard Challenge (detailed description in 2.5.1). We only use those samples which can be represented by a single phrase atom. 5% of all samples are set aside for testing which corresponds to approximately 20 minutes. We extract context embeddings and WORLD vocoder features as described in Section 2.2. From the extracted LF_0, atoms are computed by matching pursuit as proposed by Honnet et al. (2015) including a single phrase atom. Atom amplitudes are mean-variance normalised.

The length of an atom is limited to nine discrete values $\theta \in \{0.01, 0.015, 0.02, \ldots, 0.05\}$ which were found to be able to model the LF_0 contour with low error in previous research (Gerazov et al., 2015).

3.3.2 Network Topologies

The baseline system is similar to the one used by Ronanki et al. (2017) following the usual approach by predicting acoustic features plus their dynamic components. It consists of two feed-forward ReLU layers of 1024 nodes, three bidirectional GRUs with 512 nodes each, and a final linear output layer with 187 nodes (features + Δ + $\Delta\Delta$). The model is trained with Adam (Kingma and Ba, 2015) on 35 epochs (learning rate 0.002).

To emulate an SNN we use a bidirectional RNN as described in Gers et al. (2002). Rather than use LSTMs with peepholes as in that paper, we use the GRU of Cho et al. (2014) where peepholes are moot. The model we propose consists of three feed-forward ReLU layers with 128 nodes, two bi-directional GRUs with 64 nodes each, two feed-forward ReLU layers with 128 nodes, and a final linear output layer with 11 nodes. It predicts one V/UV flag, nine amplitudes (one per θ), and a spike position flag. The model is trained with Adam on 55 epochs (learning rate 0.0002). In both cases we use $\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-8}$ for Adam.

3.3.3 Synthesis

For all tests the original durations, MFCCs, and BAPs are used as we are only interested in the impact of different LF_0s on naturalness. For the baseline system LF_0 is improved by MLPG using variances computed from the training data. The waveform is synthesised by the WORLD vocoder.

In our proposed model, the spike position flag is post-processed to identify its peaks which results in a value of $\{-1, 0, 1\}$ per frame. Atoms are constructed by taking the maximum of the nine predicted amplitudes for positive spikes and the minimum for negative spikes respectively. The θ value is implicitly given by the index of the selected amplitude within the nine outputs. LF_0 is reconstructed by superposition of all predicted atoms and the original phrase atom (Figure 3.6, more figures can be found in Appendix A).

3.3.4 Objective Results

To objectively compare the models we compute the RMSE of F_0 on all frames which are voiced either in the target data or in the network prediction, and the V/UV error rate. Our model performs slightly worse than the baseline system (compare Table 3.1) but certainly close enough to validate our hypothesis that the approach is plausible.

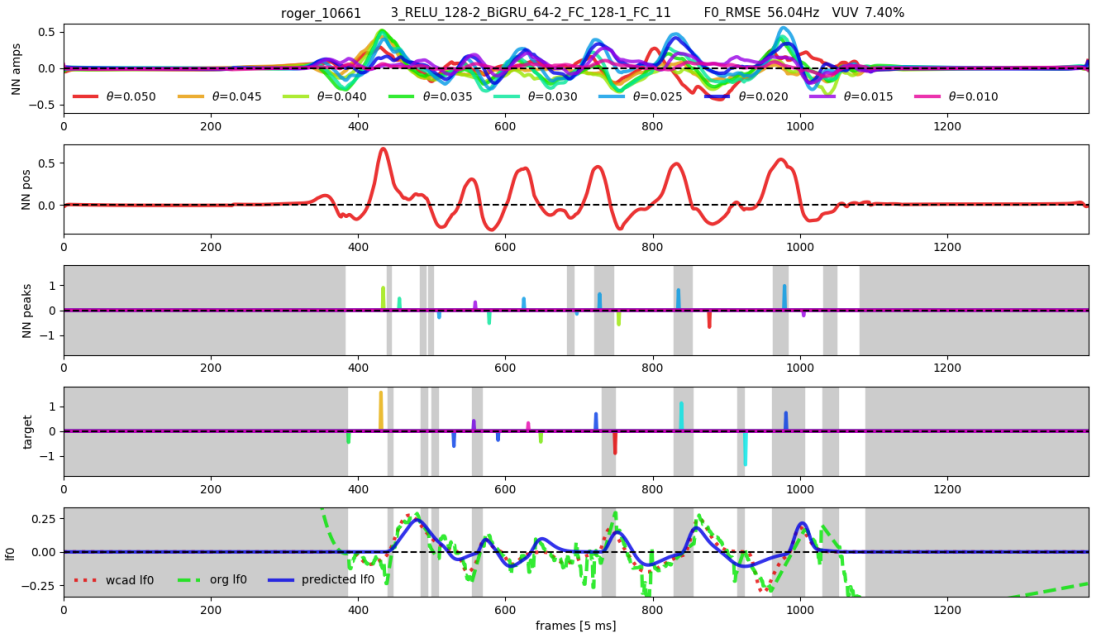


Figure 3.6 – Synthetic features on a temporal scale of 5 ms per frame. Plot descriptions from top to bottom: **1:** Nine amplitude outputs (one per θ). **2:** Spike position flag before post-processing. **3:** Atom spikes generated from spike position and amplitude max/min, V/UV flag (unvoiced frames grey). **4:** Target atom spikes and target V/UV flag (unvoiced frames grey). **5:** LF_0 (without phrase component) NN reconstruction (blue, solid), target reconstruction (red, dotted), original (green, dashed) and target V/UV flag (unvoiced frames grey).

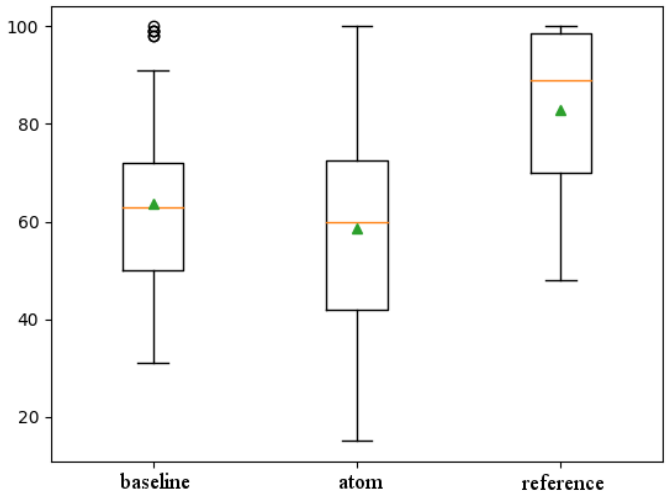


Figure 3.7 – Subjective score of MUSHRA intonation test. Medians as orange lines. Sample averages as green triangles. Outliers as circles.

Table 3.1 – Objective results.

Model	F_0 RMSE	V/UV
baseline	44.46 Hz	5.43 %
atom	49.89 Hz	5.94 %

3.3.5 Subjective Results

We measure the naturalness of the synthesised speech by a MUSHRA test (Figure 3.7) where we compare our model (*atom*) and the baseline (*baseline*) with the speech produced by the vocoder (copy synthesis) with the original acoustic features (*reference*). We randomly select a subset of 20 samples from the test set excluding those where the speaker takes a breath half way through as those samples require further phrase atoms. 17 non-native but fluent English speakers participated in the test. Each of them was asked to listen to 5 randomly selected samples from that subset and rate them on a scale from 0 to 100. They were told to focus on prosody only and ignore minor fuzzy/buzzy artefacts. As the most natural prosody is found in the reference sample, we excluded 18 results where the listener rated the baseline or the atom system more than 10 points higher than that reference. A two-tailed paired t-test on the individual ratings for the baseline and atom system gives a p-value of $p = 0.12 > 0.05$, i.e. we do not have evidence that the two systems are significantly different on a difference level of 0.05. The two-tailed paired t-tests show that both systems are significantly different to the reference. The p-value for baseline – reference is $p = 1.39\text{e}{-9}$, and for atom – reference: $p = 2.54\text{e}{-12}$.

3.4 End-to-End Atom Prediction

The model proposed in the last section has three limitations:

1. The Gamma kernels that filter the spikes are not trainable. Their parameters are imposed before training the system and may not be optimal.
2. The trained RNN is not able to generate true spikes. The amplitude and position of the command signals are split into two separate channels, requiring post-processing of the outputs to recover spikes. This post-processing operation prevents gradient back-propagation from the pitch curve.
3. The system cannot be used to predict the phrase component used by the GCR to reconstruct the LF_0. It therefore requires an external source providing the phrase curve to generate intonation contours. The model can therefore not be used to generate intonation contours without additional information regarding the phrase curve.

In this section we propose to overcome the aforementioned limitations by training an End-to-End (E2E) neural network to generate LF_0. This system includes trainable muscle models

and the generation of phrase components, both without post-processing, that allow the optimisation of the muscle parameters for intonation synthesis. To build this system, we take advantage of the existing model and replace the post-processing steps by trainable muscle models based on the second-order IIR filter explained in Section 3.1.3 (Figure 3.8). The model is a source-filter model which differs from the commonly used speech production models by generating only LF_0 and using temporal static but trained filters. Additionally, the source and filter prediction models are trained together in an E2E fashion.

Of course, any modern TTS systems can predict LF_0 (e.g. Wu et al. (2016)). The proposed system differs in the sense that it retains the physiologically inspired behaviour of the GCR model, by enforcing spiky command signals and muscle model filtering. This will allow us to conserve its transfer capability and physiological interpretation.

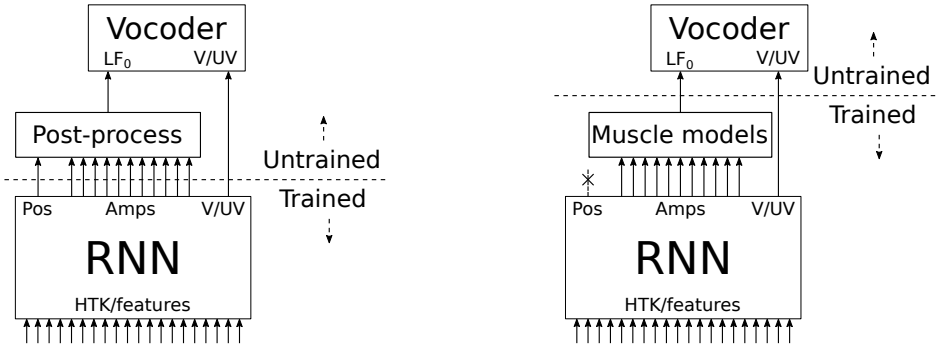


Figure 3.8 – GCR intonation synthesis systems. Left: previous model. Right: proposed E2E model. The post-processing step is replaced by trainable muscle models to enable training directly on the LF_0 curve in an E2E fashion.

3.4.1 Neural Network Implementation

In the GCR model, the output filters approximate muscle activation. Different models for muscle response are investigated by Gerazov and Garner (2015). Even though previous research (Prom-On et al., 2009; Gerazov et al., 2015) has shown that higher order systems can improve intonation modelling performance, we use a second-order SDM muscle model in this work. This choice is consistent with Fujisaki's assumption that intonation is generated by second order systems (Fujisaki and Hirose, 1984). Moreover, it is possible to obtain more complex responses by combining multiple second order models, so that this choice is not restrictive.

We integrate the trainable muscle models into the existing system by replacing the post-processing step with a new muscle models layer (Figure 3.8). In contrast to the previous system, the command signals are not split into separate position and amplitude channels. The former position output signal is therefore removed from the system. The amplitude outputs become the command inputs for the new muscle models layer. The V/UV prediction output remains unchanged as it is independent from the post-processing.

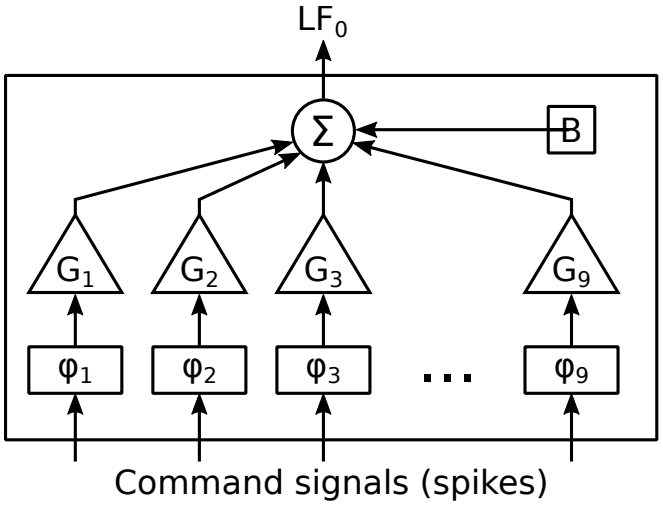


Figure 3.9 – Muscle models layer. Each muscle model φ_i is associated to a normalisation gain G_i. All the outputs are summed and the phrase bias B is added to reconstruct LF_0.

The new muscle models layer has one recurrent unit φ_i per GCR muscle (Figure 3.9). Each output is multiplied by a gain that normalises the L2 norm of the impulse response of the filters (the linearity of the models implies that the gains can be applied on the input or the output signals equally). The normalisation allows an easier interpretation of the command signals, and is consistent with the previous system that also uses normalised muscle responses. Moreover, normalisation gains help balance the impact of the gradients on the different muscle models. Trainable gains could create dominant filters that would receive higher gradients, resulting in uneven convergence speed across the muscle models.

The gain that normalises the impulse response of a filter depends only on its poles. This relationship can be computed analytically. However, as the exact expressions are difficult and computationally heavy, we use the numerical approximation in Equation 3.19. The parameters A_0 to A_4 are computed with the linear least squares method. The approximation achieves less than 5% error for atoms with $\theta \in [0.01, 0.35]$.

$$G = A_0 + A_1\rho + A_2\rho^2 + A_3 \exp\rho + A_4 \exp\rho^2 \tag{3.19}$$

The final LF_0 contour is given by summing up the normalised filter responses and adding a trainable bias, which compensates for the non-zero mean of LF_0. We believe that a trainable bias, which can depend on a global speaker ID input, is more flexible than training on a normalised LF_0 target. Compared to the existing model, the bias compensates the main shift of the phrase component.

Instead of starting from a random initialisation point we can start with a pre-trained non E2E atom model, following the same training procedure as in the last section. Predicting spikes is

a complex task, and we can use a priori knowledge provided by the existing system to provide a sensible initialisation point for our model. Thanks to the use of a similar architecture, the parameters of the input RNN can be loaded from a trained atom system. The associated parameters of the non-trainable muscle models can serve as initialisation point for our new muscle model. Nevertheless, the muscle models used in the former architecture are gamma-shaped atoms whose impulse response is

$$f_{K,\theta}(t) = \frac{1}{\theta^K \, \Gamma(K)} \; t^{K-1} \; e^{-t/\theta} \qquad \text{for} \;\; t \geq 0 \tag{3.20}$$

with K and θ the shape and scale of the atom and t the continuous time variable. To use the previous muscle parameters a relationship between gamma atoms and discrete linear filters is required. Setting $K = 2$ and using an impulse-invariant transformation (Gardner, 1986) to discretise (3.20) relates the pole modulus of discrete linear filters to the scale of gamma atoms

$$\rho = \exp\left(\frac{-T_s}{\theta}\right) \tag{3.21}$$

with T_s the sampling period. Since the filters are normalised, the gain relationship can be ignored.

3.4.2 Experimental Validation

In running experiments, we want to validate the assumption that the proposed E2E system can reproduce the behaviour of the GCR model, and generate the phrase contribution. We are also testing the hypothesis that the fixed muscle parameters used in the atom model are not optimal for intonation generation by a neural network, and that trainable models will converge to values giving better performance matching the quality of a strong baseline.

We use a subset of the 2008 Blizzard Challenge speech database (see 2.5.1; about 5.7 hours) to test our model. 5% (17 minutes) is set aside for test and evaluation set respectively. Context embeddings and WORLD vocoder features are used as before. We use a set of nine muscles for the GCR, initialised with gamma scales $\theta \in \{0.03, 0.045, ..., 0.15\}$, approximating the ones used in previous research (Gerazov et al., 2015).

Network Topologies and Training

We use the same baseline system as in Section 3.3.2. The E2E system is initialised with a pre-trained atom model, which uses the same topology and training as described before. At first the E2E model is trained for 50 epochs (learning rate of 0.001), without the phrase bias, on LF_0 from which the phrase contribution is removed. It is then trained with phrase bias on non-normalised LF_0 for another 50 epochs (learning rate of 0.0006). Note that training the system without initialisation converges but deteriorates the reconstruction performance.

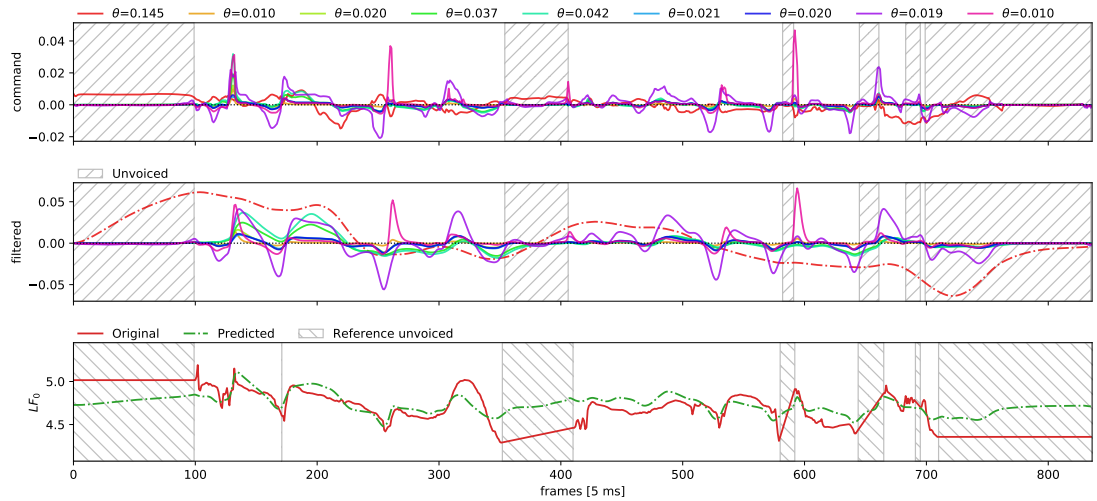


Figure 3.10 – Signals generated by the E2E model. From top to bottom: **1:** Command signals generated by the RNN, which can be assimilated to spikes of a GCR. **2:** Muscle model responses, where the slow phrase component is clearly visible (dash-dotted line). **3:** LF_0 reconstruction (dash-dotted) and original (solid). The striped regions represent unvoiced frames.

The loss is computed by summing the MSE of LF_0 on the voiced frames and the MSE of the V/UV output weighted by 0.3. In order to generate spiky command signals that fit the behaviour of an GCR model, we apply a temporal L1 constraint (Tibshirani, 1996) on the outputs of the atom model weighted by 0.3. Without this penalisation the generated command signals are not sparse and cannot be assimilated to GCR spikes (Figure 3.11). The learning rate is reduced using a plateau scheduler with a patience of five, a relative threshold of 0.001 and a factor of 0.3. All the networks are trained using Adam (Kingma and Ba, 2015) with $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$.

Objective Scores

The performance of the models is objectively measured by the RMSE of F_0 on all the target voiced frames and the V/UV error rate. Table 3.2 shows that the proposed E2E system significantly improves the performance on the existing atom model, and the obtained objective performance closely matches the score of the strong baseline system. The signals generated by the trained system are plotted in Figure 3.10, which shows the ability of the model to generate spiky signals and to synthesise the phrase component. For the tested E2E system the muscle parameters converge to the values $\theta \in \{0.01, 0.019, 0.02, 0.021, 0.037, 0.042, 0.145\}$ which are different from the initial ones. This validates our hypothesis that the fixed values used in the atom model are not optimal for this task.

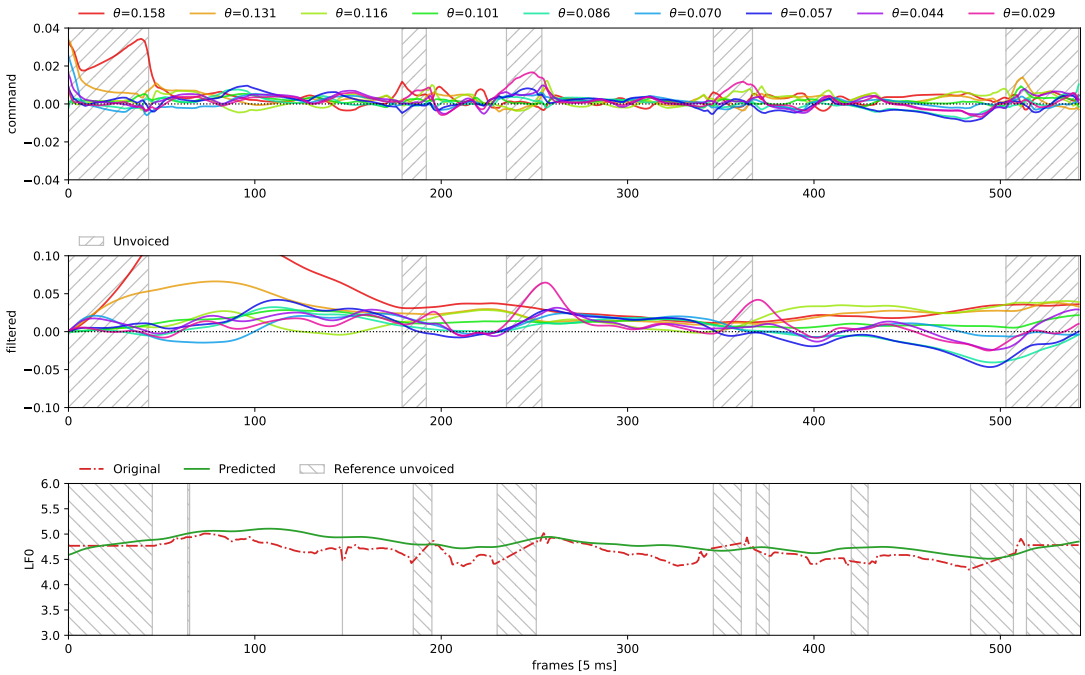


Figure 3.11 – Signals generated by the E2E model when trained without a temporal L1 constraint. Description as in Figure 3.10

Table 3.2 – Objective scores

Model	F_0 RMSE	V/UV error
Baseline	21.3 Hz	10.4 %
Atom	28.8 Hz	14.9 %
E2E	22.3 Hz	10.7 %

Subjective Measurements

We synthesise the samples with the WORLD vocoder using the original durations, MGCs, and BAPs; only the impact of LF_0 is measured. For the baseline LF_0 is improved by MLPG. The naturalness of the synthesised speech is evaluated through a MUSHRA test. It compares our model (E2E) to the previous system (atom), an anchor (only the phrase component), the baseline, and the speech synthesised using the original LF_0 (reference). 20 random samples from the evaluation set have been selected for evaluation by 42 fluent English speakers. Each of them was asked to rate 5 random samples of this subset on a scale from 0 to 100, focusing on prosody only and ignoring minor artefacts. About half of the participants had background in speech processing.

A two-tailed paired t-test on the individual ratings of the baseline and E2E system gives

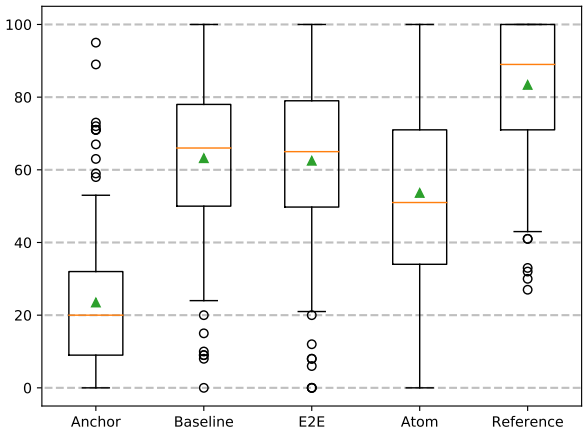


Figure 3.12 – Subjective score of MUSHRA intonation test. Medians as lines, sample averages as triangles, outliers as circles.

$p = 0.746 > 0.1$, proving that the proposed model achieves the same perceived quality as a strong baseline system. The quality of the atom model is worse, with the p-value of E2E–Atom being $p = 0.0002 < 0.1$. This is expected because the evaluation set contains multi-phrase samples, while the atom model can correctly model single-phrase samples only. Thus, the proposed model achieves a better performance on a more complex task.

3.5 Conclusion

We have shown that the combination of an emulated spiking network, a dictionary of atoms representing muscle responses, and a SPAN-inspired training algorithm can generate reasonable intonation contours. Although “reasonable” is open to interpretation, the algorithm produces subjective results that are not significantly different from an accepted baseline. The proposed training algorithm for spiking targets enables the use of DNNs in other research fields currently dominated by SNNs.

Furthermore we extended our model to an E2E model with embedded trainable second-order linear all-pole digital filters, that can generate natural sounding intonation, provided that suitable stability conditions are imposed. A temporal L1 constraint allows to produce spiky command signals to drive muscle responses, thus reproducing the behaviour of a GCR model. Taking advantage of the flexibility of E2E networks, the system can also generate the phrase component in LF_0. The objective and subjective results of this model closely match those of a strong baseline.

We noticed a clustering effect in the muscle models of the E2E model, indicating that fewer filters might be necessary. Future work could focus on analysing the effect of a reduced filter number. Additionally, the obtained muscle parameters and command signal shapes allow the psycho-linguistic analysis of the behaviour, to improve understanding the GCR model.

4 Neural All Pass Warp

In the previous chapter we have looked exclusively at pitch (more precicely the fundamental frequency F_0) and how we can develop an interpretable control within the neural network framework. But pitch is only one of the aspects which changes between neutral and affective speech. In this chapter we focus on the spectral features of emotional speech and how to develop a low-dimensional simple control for it.

VTLN is a technique first developed in ASR to normalise between different speakers (Cohen et al., 1995; Zolnay et al., 2005; Giuliani and Gerosa, 2003; Jaitly and Hinton, 2013). The technique was inspired by the observation that one significant difference between speakers is the length of their vocal tract. This length can vary from around 18 cm in males to around 13 cm in females. The length of the vocal tract is inversely proportional to the formant frequency positions. This leads to a variation of around 25% in formant center frequencies among speakers. ASR systems use VTLN to normalise different speakers to an average speaker before processing their speech with the same backbone system. It is well known that TTS systems can use the same technique to adapt an average voice to a specific voice, also called reverse VTLN. The number of parameters required for VTLN is much smaller compared to transform-based adaptations, offering an effective control for spectral feature adaptation.

Based on VTLN (section 4.2) we develop a neural APW layer (section 4.3), which is capable of temporal-aware warping of the spectrum and thus can also manipulate speech characteristics as speaking style besides speaker identity. We investigate its effectiveness in three challenges of today's TTS research. As VTLN has shown to be effective for multi-speaker speech synthesis in HMM models due to its small parameter space, we start by investigating the proposed APW in two speaker adaptation scenarios before we move on to emotional speech. This chapter covers the following three experiment blocks:

1. (Section 4.4) Speaker adaptation in multi-speaker TTS systems with small amounts of adaptation data. We use a Merlin-style model (details in 2.3.3) and compare it with the same model extended by the APW. After training both models on a multi-speaker database, we adapt both to unseen speakers with a limited amount of target data.

2. (Section 4.5) Zero-shot speaker adaptation in multi-speaker TTS systems. A parallel version of the encoder-decoder model (compare Section 2.3.4) with and without APW is investigated on the task of adapting to an unseen speaker with a single sample without transcription.

3. (Section 4.6) Emotional multi-speaker TTS. Work in emotion recognition has shown that some emotions cause a formant shift. As the APW is an effective control for formant shifting, we expect it to be a beneficial neural network component for emotional TTS.

Much of the text in this chapter appeared as:

- Schnell, B. and Garner, P. N. (2019). Neural VTLN for speaker adaptation in TTS. In *Proc. 10th ISCA Speech Synthesis Workshop*, pages 29–34 doi: /10.21437/ssw.2019-6

with an extended version under review as:

- Schnell, B. and Garner, P. N. (2021b). Investigating a neural all pass warp in modern TTS applications. *Speech Communication*

4.1 Background

In this section we try to give a thorough overview over current research trends covering related work in zero-shot speaker adaptation, affective speech synthesis, and the APW transformation in a three-fold way. In the first part we describe zero-shot and few-shot speaker adaptation methods for adaptation data with and without transcriptions. The second part gives a thorough list of unsupervised methods to achieve affective speech synthesis. The last part describes work related to the proposed all pass warp layer, which has been used only for multi-speaker systems before.

4.1.1 Few- & Zero-Shot Speaker Adaptation

Few- & zero-shot speaker adaptation is taken to mean creation of a TTS system which sounds like one or multiple target speakers unseen during training, while using only a very small amount of adaptation data for each of the target speakers. The border between zero- and few-shot adaptation is blurred, but few-shot adaptation methods usually require multiple transcribed observations of the target speaker and involve a fine-tuning step, i.e. change the network weights. Zero-shot adaptation usually relies on a single observation without transcription. Thus current research distinguishes whether the adaptation data is transcribed or not. Transcribed data allows to fine-tune the whole, or parts, of the model (sometimes referred to as meta-learning). For example, it allows to learn a speaker embedding for the target speakers. When no transcription is available models rely on extracting the speaker identity from the adaptation data through a reference encoder network. Acoustic features

are extracted from the adaptation data and form the input of the reference encoder, which generates speaker embeddings. Those embeddings differ in their granularity from global, over clustered, to frame-level embeddings.

Untranscribed adaptation data

Jia et al. (2018) use a Tacotron 2 (Shen et al., 2018) plus WaveNet combination with speaker encoder network. The speaker encoder network is trained on external untranscribed data in a speaker verification task. Then the speaker embedding is obtained from an intermediate representation at the end of the network (d-vector approach). In the speaker verification task a database with 18k speakers is used. While still good, the results show slight degradation of signal quality on embeddings extracted from unseen speakers compared to speakers seen during training. The authors report high speaker similarity and signal quality for unseen speakers, but in an ablation study they found that both quickly drop when using fewer speakers when training the speaker encoder network. The results show a significant drop in speaker similarity from 18k to 8.4k to 1.2k speakers.

Arik et al. (2018) extend Deep voice 3 (Ping et al., 2018) with a speaker encoder network. The study focuses especially on utilising multiple adaptation samples for a target speaker. The speaker encoder network consists of a CNN and mean pooling, this allows using multiple adaptation samples, as well as a self-attention mechanism to generate a global speaker embedding. To capture only the speaker identity in the embedding, they use an intercross training approach, where the speaker encoder input belongs to a different sample of the same speaker than the actual training sample. To stabilise the training they first train the speaker encoder network to predict the speaker embeddings of a previously trained multi-speaker system. They find that fine-tuning the speaker encoder network on the adaptation data improves the results. When using more than five adaptation samples fine-tuning the whole model brings additional improvements.

Cooper et al. (2020) use a speaker encoder network pre-trained on a speaker verification task in combination with a Tacotron (Wang et al., 2017b) variant (Yasuda et al., 2019). They investigate using the speaker embedding as input to: 1) the attention, 2) the attention and the pre-net, 3) the attention, pre-net, and post-net. They report variant 2) to achieve the best results. Additionally, they compare an x-vector approach with statistical pooling to learnable dictionary encodings (clustered embeddings), where they find the latter (with three clusters) to be superior.

Choi et al. (2020) combine Tacotron 2 with two speaker encoders. One coarse-grained bidirectional LSTM based encoder which produces a global speaker embedding through average pooling and one fine-grained encoder with the same structure but without the average pooling. In each decoder step a dot-product attention mechanism is used to combine embeddings from the variable length speaker embeddings of the fine-grained encoder. They find that this approach outperforms learnable dictionary encodings (Cooper et al., 2020) and Gaussian

Mixture Variational Auto-Encoder (GMVAE) (Hsu et al., 2019). An ablation study suggests that the coarse-grained encoder prevents attention collapse while the fine-grained encoder improves the speaker similarity. The model performs best when the speaker encoders receive multiple samples as input during inference and training.

Transcribed adaptation data

Nachmani et al. (2018) use the VoiceLoop model with WORLD vocoder features, but replace the speaker look-up-table with a reference encoder. They train the model with a reconstruction loss and a contrastive as well as cycle loss on the speaker embeddings. Results show that even training on a relatively small database with 85 speakers captures sufficient variation over the general population of all speaker as long as all three loss terms are used. Additionally, they argue that the training procedure of VoiceLoop encourages the model to continue with the same speaker identity over an utterance. Thus they investigate priming, where 1500 ms of the target speaker are fed in teacher forcing mode before the actual inference step starts. In their adaptation scheme only priming relies on a transcription.

Chen et al. (2019) reuse the same speaker encoder network as in (Jia et al., 2018) but combine it with a CNN-based model to capture missing residual information helpful for TTS but not speaker verification. Instead of Tacotron 2 they use the original WaveNet with linguistic features, fundamental frequency (F0), and speaker embedding input. F0 and durations are predicted by classical LSTM-based networks (Zen et al., 2016) of the old paradigm. The study compares three variants where the first two involve meta-learning and thus transcriptions: 1) speaker embeddings from a look-up-table trained on the adaptation data, 2) learn the speaker embedding as in 1) then fine-tuning of the whole model (10% of the data is used as validation set for early stopping), 3) use the speaker embedding from the speaker encoder network. Best results are obtained with variant 2), suggesting that fine-tuning is preferable in the presence of transcription. Variant 3) shows some degradation in speaker similarity.

4.1.2 Affective Speech Synthesis

Since the publication of Tacotron with Global Style Tokens (GST) (Wang et al., 2018b) the research community has worked extensively on unsupervised methods, which are especially useful for affective speech synthesis as affective databases are rare and too small for training big end-to-end TTS models. Additionally the creation of bigger affective corpora is expensive and annotating emotions error prone. We provide a list of unsupervised methods in the following. The primary objective of all these works is to extract rich (partly also interpretable) latent representations of speaking styles/emotions/affect to use them during the speech generation process. For the experiments on emotional speech synthesis in this chapter we do not rely on any of those methods but on simple learned emotion embeddings. We argue that all of the unsupervised methods can be used with our proposed APW layer and richer latent representation will only be beneficial for it. Whether richer representations decrease

the relative benefit of our proposed layer, because the TTS models become better in general, is an open research question.

Global Style Tokens

Tacotron with Global Style Tokens (GST) (Wang et al., 2018b) uses a reference encoder to compress the prosody of a variable length audio signal into a fixed-length vector which is called a reference embedding. Then an attention module is used to compute a similarity measure between a set of randomly initialized embeddings (the elements in the set are called global style tokens) and returns the weights to combine the global style tokens to a style embedding. The style embedding is used by the decoder for conditioning at every timestep. The style tokens are jointly trained with the model driven only by the reconstruction loss from the Tacotron decoder. At inference time the style encoding can either be extracted from any other audio signal or manually selected by a combination of global style tokens. The experiments show that a GST model yields interpretable embeddings that can be used to control and transfer style. It also decomposes various noise and speaker factors when trained on unlabelled noisy data.

Prosody Tacotron (Skerry-Ryan et al., 2018) is very similar to the GST model but its reference encoder generates the fixed length embedding by using a CNN followed by a GRU based RNN. The reference encoding is broadcast concatenated with the text encoder outputs and given as conditioning to the decoder. The authors state that they have also tried variable-length embeddings by using a second attention head which selects one of the GRU outputs based on the current text encoder output. However, they report that the variable length system is less robust to text and speaker perturbations. Experiments also showed that prosody is transferred in a pitch-absolute manner thus results in unwanted pitch shifts for other gender references.

Similar reference encoders have been used in voice conversion research (Lian et al., 2019) to add unsupervised prosody embeddings to the conversion model input which additionally consists of phonetic posterior grams (frame-level linguistic features obtained from a speaker independent automatic speech recognition system), converted LF_0, and voicing information. The proposed model uses LPCNet (Valin and Skoglund, 2019) as neural vocoder and demonstrates prosody and voice conversion abilities for singing references.

Lee and Kim (2019) extended the GST model further by using frame-level style embeddings. They tested the performance of these style embeddings on the text encoder and speech decoder side. To map the frames from the reference speech to the text encoder frames another dot-product attention layer is used. For the speech decoder the length of the reference audio has to match the length of the generated speech. The size of the style embedding was two or four. Any bigger sizes resulted in overfitting presumably because the network was copying the reference audio to the output. They found that the low dimensional style embeddings contain entangled pitch, amplitude, and speed information and thus allow fine-grained frame-level control while inference. The model showed voice conversion abilities for a song.

Klimkov et al. (2019) proposed phoneme-based aggregation of prosodic features to enable fine-grained prosody transfer while overcoming the instability of the second attention layer of Lee and Kim (2019) on long sequences and single-speaker databases. Only pitch and power features of the reference audio are used. In the proposed model those features are aggregated per phoneme (forced-aligned) as mean F_0 and mean energy (c_0) for each of three phoneme states plus the phoneme duration. The aggregated features are given to the reference encoder to generate a style embedding which is concatenated with the text encoding. They also used a VAE as reference encoder and demonstrated phoneme-alignment in the absence of text information of the reference by using a Connectionist Temporal Classification (CTC) based ASR model.

Gururani et al. (2019) also extract only pitch and energy from a prosody reference but compute global statistics of them (mean, variance, maximum, minimum only for pitch) which are projected to the size of the text embeddings and summed to be used as a global style embeddings. Their experiments prove the quality of the simple yet effective method on a single-speaker expressive database.

Hierarchical Latent Spaces

In Capacitron (Battenberg et al., 2019) a Tacotron is used with the reference encoder from (Skerry-Ryan et al., 2018) with a Multi-layered Perceptron (MLP) that predicts the parameters of a variational posterior. The reference encoder also has access to the text and speaker information of the reference audio. The authors argue that this allows the model to use the whole posterior space for variability in the speech, while a model without text and speaker information is likely to divide the latent space into regions that correspond to different utterances. They also demonstrate a hierarchical decomposition of the latent space. At first high-level latents are generated from the reference audio with text and speaker information. Then the low-level latents are sampled based on the high-level latents, which are used as style embeddings. This allows to sample multiple realizations for the same reference audio. The split of the capacity of the latent into high and low balances how much the output will resemble the reference and how much style variability appears between the sampled realizations. The authors analyse the variability of generated speech samples based on the style embedding capacity, i.e. the representational mutual information, which is upper bounded by the average KLD over the data distribution.

Sub-Encoder

Bian et al. (2019) split the GST style encoder into multiple sub-encoders. Each sub-encoder should capture different kinds of style classes. In their experiments the first encoder encodes mainly the speaker while the second encodes mainly the prosody. The disentanglement is achieved by intercross training. The objective is to maximize the log-likelihood of the Tacotron decoder output for different style reference encodings that share the desired sub-encoder style

class. As an example, to learn the speaker embedding multiple samples from the same speaker with different prosody styles are encoded as reference encodings while the final output of the decoder should remain the same. Thus throughout those style encodings the prosody is ignored and mainly the speaker identity is captured. The auxiliary tasks of style classification from the style sub-embeddings and sub-encoder embedding orthogonality are necessary to train the model.

Intercross Training

Whitehill et al. (2020) extended the idea of Bian et al. (2019) to an adversarial cycle consistent loss. They use the same reconstruction loss for paired data and the orthogonality loss on the style embeddings of the different style encoder outputs. Additionally they train classifiers to predict the style classes from the style embedding of each style encoder where the gradients of the classifier from the other style classes are reversed. For unpaired data they encode the generated audio again with the style encoders and train the classifiers to predict the same style class as the reference audio, thus allowing training of unpaired data without any reconstruction loss.

The beforementioned works, which use reference encoders to extract speaker or emotion embeddings, achieve good results in multi-speaker scenarios and also allow zero-shot adaptation, but they only offer limited control. The populating of the embedding space is not restricted and linear interpolation between known speaker/emotion embeddings is not guaranteed to provide high quality results. Carfully designed interpolation techniques are required. Um et al. (2020) propose a method to control the intensity of an emotion in a GST model on a single-speaker database. Firstly they extract a representative embedding vector of each emotion category which maximizes the inter-category distance to the closest and farthest other category while minimizing the intra-category distance. Secondly, they propose a non-linear interpolation function to vary the emotion intensity from neutral to emotional speech. The APW we propose is an alternative that provides a low-dimensional control and guarantees high audio quality by design.

Variational Auto-Encoders

Akuzawa et al. (2018) combined VoiceLoop (Taigman et al., 2018) with a VAE reference encoder and showed that the quality of the generated speech exceeds that of the vanilla model. The model allows to sample new styles from the prior in the latent space as well as style transfer by encoding a given reference. The analysis in a similar work of Zhang et al. (2019b) revealed that several dimensions of the latent space could independently control style attributes such as pitch-height, local pitch variation, and speed. Thus they argued that the VAE has disentangled interpretable features in the latent dimensions. A simple control of these variables remains non-trivial.

Aggarwal et al. (2020) extended a VAE reference encoder with a Householder Normalizing Flow (Tomczak and Welling, 2016) which transforms samples from a diagonal Gaussian to samples from a full co-variance Gaussian. The authors argue that a full co-variance Gaussian distribution could improve disentanglement and reconstruction of the variability in natural speech. Experiments on one-shot adaptation to new expressive styles show the superiority of the proposed model compared to standard VAE reference encoders.

VAE-based models have shown good results in generating F_0 (Hodari et al., 2019), C_0, and durations (Kenter et al., 2019b) as well.

Generative adversarial networks

Ma et al. (2018) implemented the GST model in a GAN inspired setup (a short introduction to GANs can be found in Section 2.1.2). For a given text the style encoder (same as in GST) encodes a reference for paired and unpaired data. As in the GST model the combined content and style embeddings are given to the decoder to generate mel-spectrogram. The discriminator is trained as a ternary classifier to predict whether each of the generated samples and a real sample, given the context embeddings, is "fake from paired data", "fake from unpaired data", or "real audio sample". They also use the standard Tacotron reconstruction loss for the paired data and a style loss for the unpaired data. The style loss is the L2 distance between the gram matrices of the reference and generated mel-spectrograms. The model is able to capture speaker identities on the VCTK database and four emotion classes on an internal database. It is furthermore able to perform style transformation which results in speaker adaptation in the former and emotion adaption in the latter case.

4.1.3 All Pass Warp

All pass warp transformations have successfully been used before in multi-speaker speech synthesis systems. Sundermann and Ney (2003) clustered the source and target speaker's speech by frequency spectra of period-synchronous frames into artificial phonetic classes. To perform voice conversion for each source class the most similar target class is determined. For each class the warping parameters are selected which minimize the Euclidean distance of all warped source frames to all target frames.

Speaker adaptation from an average model with a single speaker-dependent warping selected by line search was proposed in (Eichner et al., 2004).

VTLN is an utterance level implementation of an all pass warp transformation. Shah et al. (2018) trained two DNNs to imitate the VTLN and reverse VTLN step for each speaker. To estimate the unknown normalized features the authors propose an iterative unsupervised algorithm: 1. Train a speaker-independent Gaussian Mixture Model (GMM), 2. estimate the warping parameters with Maximum Likelihood Estimation (MLE) between input features and predicted normalized features, 3. retrain the GMM with warped input features, 4. repeat step

2 and 3 five times. In contrast to us they only train a DNN to behave like a VTLN (implicit), but they do not provide a neural network component that explicitly implements it. I.e. at no point in the model the warping parameter is computed and thus cannot be controlled.

Previous work at our laboratory has already demonstrated speaker adaptation in the mathematical framework of HMMs. Speaker specific warping parameters were estimated with the expectation maximization (EM) algorithm with grid (Saheer et al., 2010) and Brent's search (Saheer et al., 2012) for different classes which are based on a regression task tree developed from decision tree questions. The proposed VTLN adaptation led to faster adaption that is more natural than unconstrained linear transformations.

The work closest to ours is that of Kotani et al. (2017). They predict a time-dependent linear conversion matrix and bias with two DNNs. In more recent work (Kotani and Saito, 2019) they perform voice conversion with a weighted sum of linear transformations on acoustic features. The conversion matrix and bias of each linear transformation are jointly computed from a mean and full-covariance matrix which are predicted by a mixture density network. For predicting the latter the Cholesky decomposition is used. This forms an explicit relation between conversion matrix and bias, however, they do not constrain the matrix to be a VTLN warping matrix, thus the benefit of a small parameter space, i.e., a single time-dependent warping parameter, is lost.

4.2 Vocal Tract Length Normalisation

The all pass warp technique we present is inspired by a well known technique for speaker adaptation in ASR and TTS: Vocal Tract Length Normalisation (VTLN). It stems from the fact that a key difference between speakers is the length of their vocal tract. The difference in length results in a shift of the formant frequencies. Intuitively VTLN is a warping of the spectrum; however, Pitz and Ney (2005) showed that it can be expressed as a linear transformation in the cepstral space, and can therefore be expressed as a matrix multiplication in that space. Usually an $N \times N$ warping matrix $\boldsymbol{A}$ pre-multiplies N mel-cepstral coefficients to produce their warped representation. A significant amount of research has gone into selecting an appropriate normalization function, which includes piecewise-linear, power, quadratic, and bilinear functions. Here we use a bilinear transform, i.e., a non-complex all-pass warp, to generate $\boldsymbol{A}_\alpha$ (it only depends on a single warping parameter α). This common implementation of VTLN is the same bilinear transform used in the generalized cepstral analysis (Tokuda et al., 1994).

The element in the k-th row and l-th column can be computed in two ways, 1) recursively (Oppenheim and Johnson, 1972; Saheer et al., 2010) by

$$\boldsymbol{A}_{k,l} = \begin{cases} \alpha^k & \text{if } l = 0 \\ 0 & \text{if } l > 0, k = 0 \\ \boldsymbol{A}_{k-1,l-1} + \alpha[\boldsymbol{A}_{k,l-1} - \boldsymbol{A}_{k-1,l}] & \text{otherwise,} \end{cases} \tag{4.1}$$

or 2) explicitly (equation (15) in Pitz and Ney (2005)) by

$$\boldsymbol{A}_{k,l} = \frac{1}{(l-1)!} \times \sum_{\substack{n= \\ \max(0,l-k)}}^{l} \binom{l}{n} \frac{(k+n-1)!}{(k+n-l)!} (-1)^{\overbrace{n+l+k}^{\text{added}}} \alpha^{2n+k-l}. \tag{4.2}$$

Equation (4.2) was originally developed in Pitz and Ney (2005) for positive alphas only. We extended it by the part marked *added* to make it also valid for negative alphas. The form of the resulting warping matrix $\boldsymbol{A}_\alpha$ for different alphas is qualitatively expressed in Figure 4.1. Most of the matrix is zero and the cepstral value after warping is a linear combination of the coefficients around the diagonal with alternating signs.

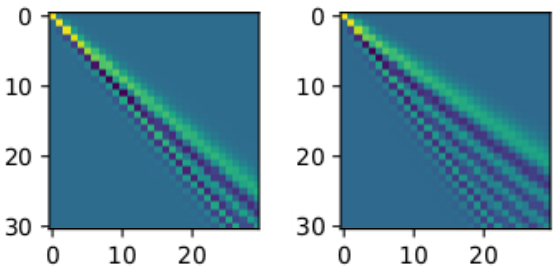


Figure 4.1 – Qualitative representation of a VTLN warping matrix for a bilinear transform (left: $\alpha = 0.1$, right: $\alpha = 0.2$).

Warping a mel-cepstral coefficient vector $x = (c_1, \ldots, c_N)^T$, or its extended version with deltas (Δ) and double deltas (Δ^2) of a single frame is as simple as Equation 4.3 and 4.4.

$$\boldsymbol{x}_\alpha = \boldsymbol{A}_\alpha \boldsymbol{x} \tag{4.3}$$

$$\begin{bmatrix} \boldsymbol{x}_\alpha \\ \Delta\boldsymbol{x}_\alpha \\ \Delta^2\boldsymbol{x}_\alpha \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_\alpha & 0 & 0 \\ 0 & \boldsymbol{A}_\alpha & 0 \\ 0 & 0 & \boldsymbol{A}_\alpha \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \Delta\boldsymbol{x} \\ \Delta^2\boldsymbol{x} \end{bmatrix} \tag{4.4}$$

When we turn α into a time-dependent parameter we cannot speak about VTLN any more as the vocal tract length of a speaker does not change while speaking. The terminology that describes the presented transformation best is an *all pass warp*.

4.3 Neural Network Implementation

To use the APW within modern state-of-the-art deep learning frameworks a neural implementation is required. We have developed an implementation in the PyTorch framework [1]. On the one hand, we found that any recursive structure based on Equation 4.1 does not allow efficient training. Even caching the `Autograd` computational graph of the forward pass leaves us with the high overhead of a recursive backward propagation. On the other hand, computing $\boldsymbol{A_\alpha}$ directly with Equation 4.2 recomputes many factorials each time. A possible solution is to pre-compute $\boldsymbol{A_\alpha}$ for a range of α with a certain precision. In the forward pass we then use the weighted sum of the two pre-computed matrices with the α value closest to the current input α. Even though this implementation gives good results we rejected it because it only approximates $\boldsymbol{A_\alpha}$. Instead we propose an efficient implementation that splits the constant and variable parts of Equation 4.2. Equation 4.2 can be represented by the sum of multiplications of constants with the $2N$-polynomial map of α which is $\boldsymbol{\alpha} = (1\ \alpha\ \alpha^2\ \alpha^3\ \dots\ \alpha^{2N})$. We designed this sum as the dot-product of the polynomial map vector $\boldsymbol{\alpha}$ along the third dimension of a constant matrix $\boldsymbol{A}^{3D}$, which has the size (N x N x 2N).

$$\boldsymbol{A}_{k,l} = \frac{1}{(l-1)!} \sum_{\substack{n= \\ \max(0,l-k)}}^{l} \binom{l}{n} \frac{(k+n-1)!}{(k+n-l)!} (-1)^{n+l+k} \alpha^{2n+k-l} = \boldsymbol{A}^{3D}_{k,l} \boldsymbol{\alpha}$$

$$\boldsymbol{A}^{3D}_{k,l,2n+k-l} = \begin{cases} \frac{1}{(l-1)!}\binom{l}{n}\frac{(k+n-1)!}{(k+n-l)!}(-1)^{n+l+k} & \text{if } l\text{-}k \le n \le l, \\ & n \ge 0 \\ 0 & \text{otherwise} \end{cases} \tag{4.5}$$

The matrix $\boldsymbol{A}^{3D}$ is computed once when the layer is created. The forward pass consists of three steps:

1. Compute the polynomial map $\boldsymbol{\alpha}$ (most efficiently by using `cumprod`)
2. Compute $\boldsymbol{A}_\alpha = \boldsymbol{A}^{3D}\boldsymbol{\alpha}$ with the dot-product along the third dimension
3. Compute one frame of warped mel-cepstrum coefficients $\tilde{\boldsymbol{x}} = \boldsymbol{A}_\alpha \boldsymbol{x}$

The above three step computation can be efficiently parallelized across all time frames and the whole batch by using the batched version of the matrix-matrix and matrix-vector multiplication. This is commonly implemented in modern matrix computation frameworks. Only step 1 contains a sequential operation, however, it is only sequential for a single frame and can be parallelized across frames. Additionally as the size of $\boldsymbol{\alpha}$ is only $2N$ with usually $N <= 60$ the computational cost is small. As we rely only on PyTorch tensor implementations the gradient is computed automatically by `Autograd`. With increasing number N of mel-cepstral coefficients our implementation becomes unstable due to high factorials in $\boldsymbol{A}^{3D}$ and small

[1] Code available at https://github.com/idiap/IdiapTTS.

polynomials in $\boldsymbol{\alpha}$. A comparison with a matrix computed recursively with Equation 4.1 reveals that up to $N = 35$ the error of our implementation is $< 10^{-8}$ for values in $\boldsymbol{A}_\alpha$ and $< 10^{-5}$ for the gradients based on floating point precision. The error quickly explodes for higher values of N. Moving the computation into log-space does not solve the problem as it compensates the error caused by the high factorials but increases the error caused by the small polynomials. We find using double precision computation for $N > 35$ solves the issue.

4.3.1 Memory Consumption

Even though PyTorch's `Autograd` computes the gradient automatically, we can look at the differential operations needed to estimate the required memory consumption. Assume that we have received the gradient $\frac{\partial L}{\partial \tilde{\boldsymbol{x}}} = \Delta_{\tilde{\boldsymbol{x}}}$ of the loss w.r.t. the warped features $\tilde{\boldsymbol{x}}$. We can now back-propagate through the three steps above:

3.
$$\frac{\partial L}{\partial \boldsymbol{x}} = \frac{\partial L}{\partial \tilde{\boldsymbol{x}}} \frac{\partial \tilde{\boldsymbol{x}}}{\partial \boldsymbol{x}} = \Delta_{\tilde{\boldsymbol{x}}} * \boldsymbol{A}_\alpha^T \tag{4.6}$$

$$\frac{\partial L}{\partial \boldsymbol{A}_\alpha} = \Delta_{\tilde{\boldsymbol{x}}} * \boldsymbol{x} = \Delta_{\boldsymbol{A}_\alpha} \tag{4.7}$$

2.
$$\frac{\partial L}{\partial \boldsymbol{\alpha}} = \Delta_{\boldsymbol{A}_\alpha} * \boldsymbol{A}^{3D} = \Delta_{\boldsymbol{\alpha}} \tag{4.8}$$

1.
$$\frac{\partial L}{\partial \alpha} = \Delta_{\boldsymbol{\alpha}} \cdot \begin{pmatrix} 0 & 1 & 2\alpha & \dots & (2N-1)\alpha^{2N-2} \end{pmatrix} \tag{4.9}$$

Looking at the memory we have to consider tensors cashed by `Autograd` and gradients flowing backwards. We denote T for the length of the sequence, B for the batch size and N for the number of mel-cepstrum coefficients. Cached values are $\boldsymbol{A}_\alpha$ ($T \times B \times N \times N$), $\boldsymbol{x}$ ($T \times B \times N$), $\boldsymbol{A}^{3D}$ ($N \times N \times 2N$), and $\boldsymbol{\alpha}$ ($T \times B \times 2N$). Gradients are $\Delta_{\tilde{\boldsymbol{x}}}$ ($T \times B \times N$), $\Delta_{\boldsymbol{A}_\alpha}$ ($T \times B \times N \times N$), and $\Delta_{\boldsymbol{\alpha}}$ ($T \times B \times 2N$). As $N \ll TB$ the overall memory complexity is $\mathcal{O}(TBN^2) \times 4$ bytes for floating point precision.

4.3.2 Model Integration

We integrate our proposed all pass warp layer into TTS neural network architectures by simply stacking it on top (Figure 4.2). We denote the neural network that generates acoustic features without warping as the *pre-net*. To generate a warping matrix we first have to predict a warping value α on a frame-wise basis ($\boldsymbol{\alpha}$ above). We use the output of the penultimate layer of the pre-net as one of the inputs to a fully-connected layer with a single output neuron. The other inputs to the layer can be embeddings that influence the warping. In Section 4.4.1 we use speaker embeddings concatenated with the penultimate pre-net layer output to use the all

pass warping layer for speaker adaptation. The same configuration is used in the zero-shot speaker adaptation experiments in Section 4.4.2. Additionally, we will use it with emotion embeddings to perform the emotion adaptation experiments Section 4.6. The activations are passed through a *tanh* non-linearity and scaled to be in a range that makes sense for the task. Our implementation allows multiple alpha layers that each predict an alpha value. Performing two all pass warp transformations with warping factors α_i and α_j is equivalent to a single all pass warp with a warping factor of

$$\alpha = \frac{\alpha_i + \alpha_j}{1 + \alpha_i \alpha_j}. \tag{4.10}$$

Our implementation combines multiple warping factors with Equation 4.10 first and then builds a single warping matrix to minimize computation. However, the experiments in this work do not include multiple warpings.

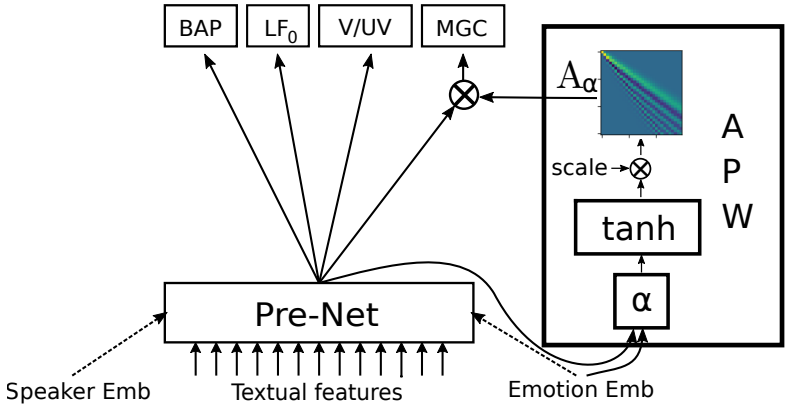


Figure 4.2 – Network structure with an APW layer. The α parameter is estimated per frame from the pre-network. The layer also has access to some embeddings that influence the warping. The figure shows the embedding for a multi-speaker emotional TTS system.

4.3.3 Experimental Proof of Concept

For this experiment we use the VCTK database (section 2.5.2), but only the 33 speakers (~11.5h) with English/no accent. We use context-embeddings as input and WORLD acoustic features with their dynamic features as target (Section 2.2). Whenever we use speaker embeddings we use 128 dimensions.

In this experiment we demonstrate that the proposed model is capable of learning a specific phoneme-dependent warping parameter α_t. Estimating α_t between phonemes of different speakers is difficult and would also require to consider preceding and succeeding phonemes as well as the mood of the speakers. To make sure we know what is the desired warping parameter, we create an artificial speaker from our base speaker (speaker p276, female). For that we randomly select a warping parameter between -0.2 and $+0.2$ for half of the phonemes and random values within the same range for the remaining phonemes and warp the MGC

features of the base speaker, belonging to that phoneme, with it. Using the same warping for half of the phonemes was introduced by an implementation bug and simplifies the task. We also report results where phonemes are randomly assigned to seven clusters where each cluster has a random warping value in the same range.

We first train the pre-network with the base speaker samples (~20 minutes) for 25 epochs, 0.05 dropout on all layers (PyTorch's implementation is used for recurrent layers), a batch size of 32, and a learning rate of 0.001. In all experiments we use a plateau scheduler with a patience of five. We then stack the APW layer on top of the pre-network, keep its weights fixed, and train only the APW layer on the artificial speaker samples for 15 epochs and the same hyper-parameters as before but with a batch size of two due to the high memory requirement of our APW implementation (we improved our implementation for experiment 2 and 3). The artificial samples are the same as used before but warped (so again ~20 minutes). We can now compare the α_t predicted internally by our model to the ground truth.

Our experiments show that the APW layer learned to compensate about 41% of the error introduced by the artificial warping. Table 4.1 shows that the compensating works better in the lower bins, which is the expected behaviour of APW. Figure 4.3 shows how the internally predicted α_t (green) follows the artificial random warping parameters (red, cornered) on a phoneme-basis. This result proves that the proposed model is capable of learning the expected warping parameter in a phoneme-dependent manner, which suggests that it will also perform well for more complex dependencies between warping parameter and phoneme+context+other (global) influences. In the seven clusters scenario the overall MCD of the pre-net is reduced from 6.04 to 5.45 dB. The MCD of the final output is slightly increased from 3.55 to 3.81 dB. The resulting reduction is thus reduced from 41% to 30.1% (Figure 4.4). However, we can draw the same conclusion from this scenario.

Table 4.1 – MCD compensation (last column) of α predicted by the APW layer of original MGCs to artificially warped MGCs for different sets of MGC bins (first column) in the scenario where half of the phonemes share the same artificial warping. Second column shows the MCD of original MGCs compared to artificially warped MGCs. Third column contains the MCD of original MGCs warped with the predicted α compared to the artificially warped MGCs.

Coef	**Org [dB]**	**Org NN α [db]**	**Compensation [%]**
1-10	4.03	2.30	43.0
1-11	4.24	2.43	42.9
1-12	4.41	2.53	42.7
1-13	4.56	2.61	42.7
1-18	4.97	2.92	41.3
all	6.04	3.55	41.1

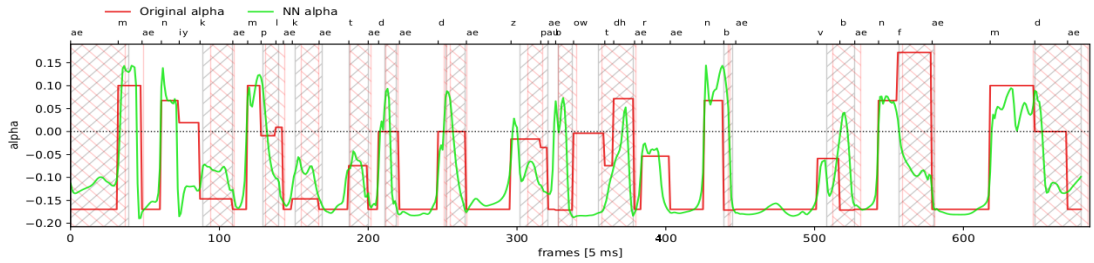


Figure 4.3 – Half of the phonemes share the same alpha, the rest have a random value. Internally predicted alpha (green) against artificial alpha (red, cornered) used to create the artificial speaker on a temporal scale of 5 ms per frame. Ground truth V/UV (grey, hatched upwards), predicted V/UV (red, hatched downwards).

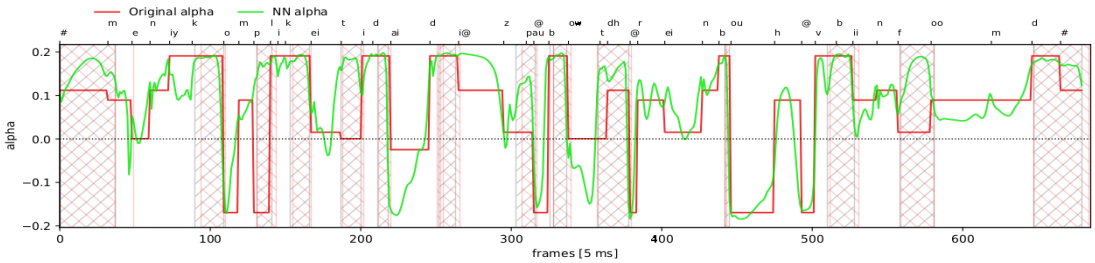


Figure 4.4 – Seven clusters scenario. Same plot description as the Figure 4.3 above.

4.4 Experiment 1: Few-Shot Adaptation

In this section we compare multi-speaker Merlin-style models (details in Section 2.3.3) with and without APW layer. We first present a comparison in objective scores for seen training speakers without adaptation (section 4.4.1). This simply tests the generalisability of the multi-speaker systems. Then we show objective and subjective scores on a speaker adaptation task with limited amount of adaptation data (section 4.4.2). We train multi-speaker systems with 29 out of the total 33 UK English speakers of the VCTK (section 2.5.2). We randomly exclude samples from training for validation and test set. We do not explicitly set specific utterances aside for testing. This means that the model is tested to produce a known utterance from a known speaker (note that the combination of the two is unseen for the network). This is necessary because the VCTK database consists of only ~400 different utterances and excluding specific utterances from training results in very low quality because the lexicon coverage is small.

4.4.1 Multi-Speaker System

As a baseline system we use a simple yet effective algorithm (Luong et al., 2017). It consists of a Merlin-style model, which takes an additional speaker embedding as input to all its layers. New speakers can be learned by only learning the embedding vector of the new speaker. In our experiments we use an RNN with two fully-connected layers with ReLU activation and 1024

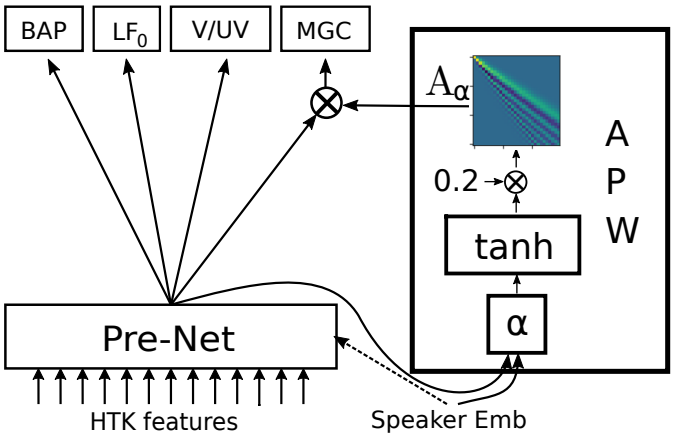


Figure 4.5 – Network structure with an APW layer. The α parameter is estimated per frame from the pre-network. The layer also has access to the speaker embedding.

neurons, three Bidirectional Long Short-Term Memory (BiLSTM) layers with 512 neurons, and a final 97 dimensional output layer (Fan et al., 2014). We train the baseline for 15 epochs, 0.05 dropout on all layers, a batch size of 32, and a learning rate of 0.001. We also tried training from scratch or fine-tuning with a batch size of two, but we did not see improvements in objective scores. The APW model uses the baseline architecture for its pre-network and also takes the speaker embedding as input to the APW layer (see Figure 4.5). The APW model is trained from scratch for 25 epochs, 0.05 dropout on all layers, a batch size of two, and a learning rate of 0.001. We also train a APW model with a pre-trained pre-network. The pre-network is trained in the same way as the baseline system and the APW model is trained for another 15 epochs with the same parameters.

Table 4.2 shows the objective scores of the baseline system, the proposed APW system trained from scratch, and the proposed APW system trained with the baseline used as pre-network initialization. Our model outperforms the baseline in this general multi-speaker speech synthesis task in objective scores. From the objective scores the initialization does not seem to have a great effect, however, perceptually we notice a higher quality. Therefore we use the initialized system in further experiments.

Table 4.2 – Objective scores of multi-speaker system trained with 29 speakers.

Model	**MCD [dB]**	**F_0 RMSE**	**V/UV [%]**	**BAP [dB]**
Baseline	6.1	17.6	12.2	21.3
APW scratch	5.3	16.3	11.6	17.9
APW	5.3	16.4	11.6	17.7

4.4.2 Speaker Adaptation

As a second step we test the two systems (baseline and APW with pre-network initialization) on a few-shot speaker adaptation task. We use the four speakers previously excluded from training (two male and two female). As some samples were excluded from the database after recording we make sure that we use only utterances which are available for all of the four speakers (exactly 400). Even though we randomly split the utterances we use the same utterances for all speakers in training, validation, and test set respectively. Both systems learn only the speaker embedding of the new speakers. We did not fine-tune the whole model as it was just recently proposed Arik et al. (2018) and shown to be beneficial Chen et al. (2019) after we did our research. We train both systems for 128 epochs with a learning rate of 0.01 and use early stopping to select the best model. For the objective scores we train once with 380 (~14 minutes) and once with only 10 utterances per speaker (~25 seconds).

Table 4.3 shows the average objective scores for the speaker adaptation task. We see that our model also outperforms the baseline in this task. The high APW pre-network MCD shows that our model makes heavy use of its warping ability and non-zero warping parameters are learned.[2] F_0 RMSE and BAP are better for male than female speakers revealing a shortcoming of both models for high pitched voices. More adaptation data does not lead to better objective scores. We hypothesise that the model either has learned the concept of speaker well and/or that the new speakers are close the known speakers so that ten utterances are enough to learn a proper embedding.

To evaluate the subjective quality of our model adapted with 10 utterances we conducted an ABX preference test (see 2.4.3) on speaker similarity with 46 participants where the original sample was given as reference. The listeners could select that they do not prefer any of the two systems. The results in Figure 4.6 show that our model is also subjectively superior to the baseline system. Half of the listeners preferred our model.

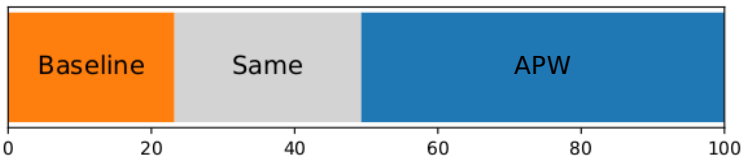


Figure 4.6 – Preference test: 23.2% prefer the baseline, 26.1% have no preference, 50.8% prefer the proposed APW system.

[2] We subsequently found a software error that caused this difference to be too large. The gap is still prominent. We did not repeat the experiments, as it does not change the conclusion we draw.

Table 4.3 – Objective scores of speaker adaptation task with four speakers (two male, two female). The scores are also separated into gender (*a*: all, *f*: female, *m*: male). APW pre-network is the score of the APW pre-network without warping.

Model	MCD [dB]			F_0 RMSE			V/UV [%]			BAP [dB]		
	a	f	m	a	f	m	a	f	m	a	f	m
Number of adaptation / test utterances per speaker: 380 / 10^3												
Baseline	**6.4**	6.4	6.4	21.0	26.6	15.4	13.4	13.3	13.5	20.9	22.1	19.7
APW	**5.7**	5.6	5.7	20.0	25.8	14.2	12.2	12.6	11.7	17.5	18.5	16.5
APW prenet[2]	12.6	12.7	12.5									
Number of adaptation / test utterances per speaker: 10 / 195												
Baseline	**6.4**	6.4	6.5	19.9	22.8	17.0	13.3	12.8	13.8	21.0	22.4	19.7
APW	**5.7**	5.6	5.8	18.8	21.1	16.4	12.9	13.1	12.8	17.7	18.6	16.9
APW prenet[2]	12.7	12.5	12.8									

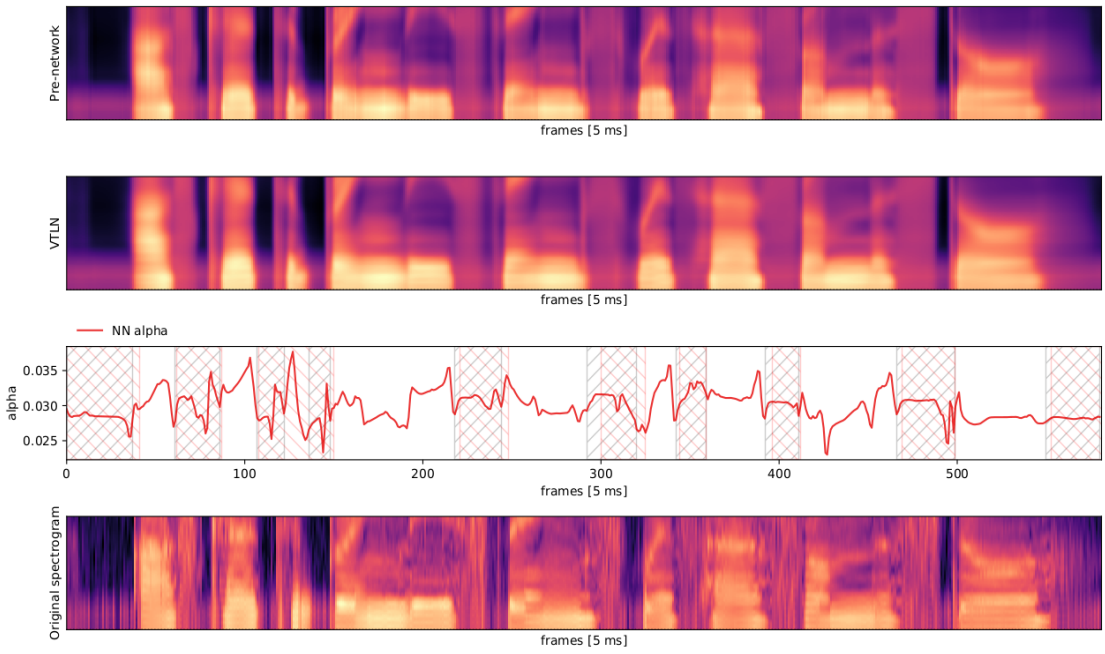


Figure 4.7 – First 150 bins of spectrogram of pre-network (first), final output after APW (second), warping parameter used (third), and original extracted from audio for utterance 002 of speaker p276 (female). The network is using only positive warpings for the female speaker. It is visible that formants are shifted slightly upwards.

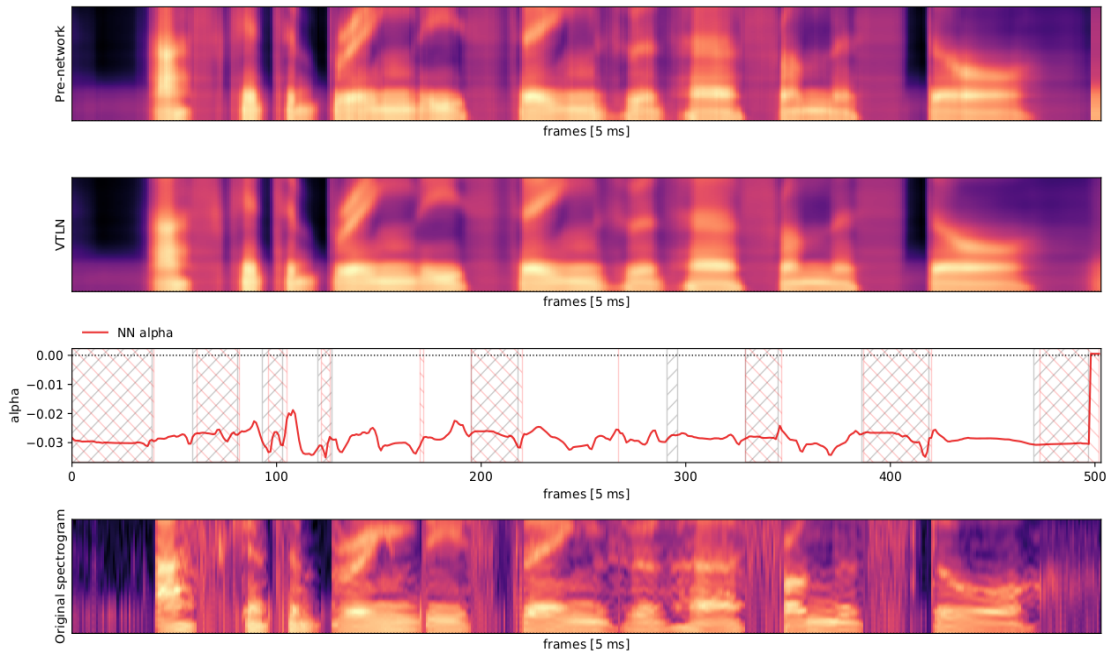


Figure 4.8 – Same as above but for a male speaker. The network is using only negative warpings for him. Formants are shifted slightly downwards.

4.5 Experiment 2: Zero-Shot Adaptation

In this experiment we investigate the use of an APW in a modern encoder-decoder model for zero-shot speaker adaptation on WSJCAM0 (database description in Section 2.5.3). We expect that the APW can play off its abilities in the zero-shot adaptation scenario, because it improves the generalisability of the TTS model and thus can lead to higher audio quality and/or speaker similarity for unseen speakers. At the same time we want to prove its effectiveness with modern encoder-decoder models of the new paradigm, which we were lacking in the experiment of the previous Section 4.4.2. We expect to see a positive effect in new paradigm encoder-decoder models as well, even though they itself have better speaker adaptation capabilities compared to the old paradigm models. We stress that, the system being constrained essentially to vocal tract normalisation, we do not expect it to outperform other more capable techniques. Rather, we use difficult speaker adaptation as a proof that the system is capable of doing what VTLN is known to do, before applying it to the core problem of emotion adaptation.

4.5.1 Model architecture

We use a state-of-the-art encoder-decoder architecture (see Section 2.3.4) inspired by Tacotron2 (Shen et al., 2018). It consists of a text-encoder, a reference encoder, an attention mechanism, and a decoder. We describe all modules in detail in the following. We use phonemes as inputs and WORLD acoustic features without deltas and double deltas as targets (a detailed description of the different features can be found in Section 2.2).

Text-Encoder

The text-encoder is the same as in Tacotron2 but its inputs are 128-dimensional phoneme embeddings. It consists of three convolutional layers each containing 512 filters with shape 5×1, followed by ReLU activation and batch normalisation. The last convolution is followed by a bi-directional LSTM with 128 units in each direction. This network should model the context embeddings labels of the old paradigm, thus providing context at each step.

Reference encoder

We use a similar reference encoder as in the Tacotron GST paper (Wang et al., 2018b) followed by a VAE as in (Battenberg et al., 2019). It consits of six CNN layers with a 3×1 kernel, 2×2 stride, ReLU non-linearity, and batch norm. The layers have 32, 32, 64, 64, 128, and 128 filters respectively. In contrast to other work we use 1D convolutions because the MGCs are already low dimensional and we do not want to blur frequencies together. The convolutional layers are followed by a unidirectional GRU where we take the last state as input to the VAE. A linear layer predicts the 128-dimensional mean μ and logarithmic variance $\log\sigma^2$ of a diagonal Gaussian posterior. The speaker embedding is produced by sampling from the posterior (reparametrization trick of Kingma and Welling (2014)).

Fixed Attention

A major difference of our model is that we used “Fixed Attention”, which means that we build the attention matrix from ground truth duration information generated in the forced-alignment step by HTK. Watts et al. (2019) have recently shown that this does not significantly deteriorate the overall synthesis quality. We mainly use it to speed up convergence and reduce the computational cost. We broadcast concatenate the speaker encodings from the reference encoder with the text-encoder outputs and use the fixed attention matrix to select an input for the decoder side. We use the word "select" here to emphasize that each row in the fixed attention matrix is one-hot.

Autoregressive Decoder

The autoregressive decoder-RNN consists of one fully connected layer of 512 ReLU units followed by a stack of two unidirectional LSTM layers with 1024 units each. Its output is projected through a linear transformation to predict the target acoustic features. As we use a fixed attention matrix we do not need a “stop token” prediction. The decoder-RNN predicts a chunk of five frames at a time. We found that this was necessary to achieve good audio quality. Its previous prediction is passed through an audio-encoder (often referred to as pre-net) containing two fully connected layers of 256 units with ReLU activation and a single unidirectional LSTM with 1024 units. We found that this additional LSTM layer (compared to Tacotron2) greatly improves the performance of our model. It can be seen as moving the

recurrent part of the attention network into the text-encoder. We also interpret the sequence of audio-encoder and decoder-RNN with linear projection as an auto-encoder (Figure 4.9), thus both networks (the decoder-RNN and the audio-encoder) should mirror each other or at least have similar capabilities.

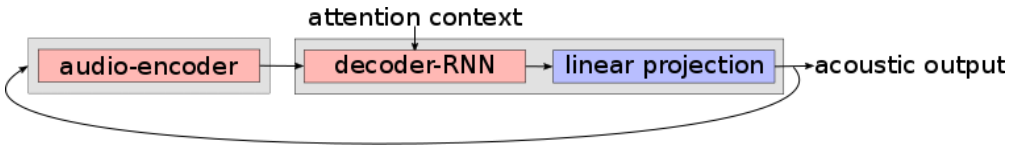


Figure 4.9 – Auto-encoder representation of the autoregressive decoder. The audio-encoder output can be seen as the bottleneck features, the decoder-RNN has external conditioning on the attention context.

Our preliminary experiments showed an incompatibility of the autoregressive decoder with the APW. The APW assumes that the pre-net outputs MGCs of an average voice which are warped by the APW to the target speaker identity. The autoregressive nature of the decoder requires it to feed the generated acoustic features back to the pre-net. As the model generates the next chunk of MGCs from the pre-net input, that input has to be the average voice. This is impossible during training because teacher forcing uses the target speaker features as input. We found that an autoregressive decoder trained in this configuration generates a warping with a "loading" phase before it reaches the desired warping over the chunk. In Figure 4.10 the autoregressive decoder outputs a chunk of five frames per decoder step. In each step the warping starts from near zero, then changes quickly for two frames, and then remains rather stable for the remaining two frames. We assume this is caused by the teacher forcing target which is very close to the target features in the next frame. Thus it does not require much warping to adapt it to the target voice in the first frame. Over the remaining frames of the chunk the network learned to predict an average voice and combine it with a warping. During inference the autoregressive input is not perfect and the loading phase deteriorates the audio quality. Instead we are using a parallel decoder (described in the following Section 4.5.1) for our experiments which still outperforms the old paradigm models (details in Section 4.6.2).

Post-net

The predicted acoustic features are passed through a post-net to predict an additive residual to smooth the overall reconstruction. We interpret it as the MLPG step when predicting Δ and $\Delta\Delta$ features. We use the same post-net architecture as in Tacotron2. Five convolutional layers with 512 filters with a shape of 5×1, followed by TanH activation on all but the last layer, and batch norm.

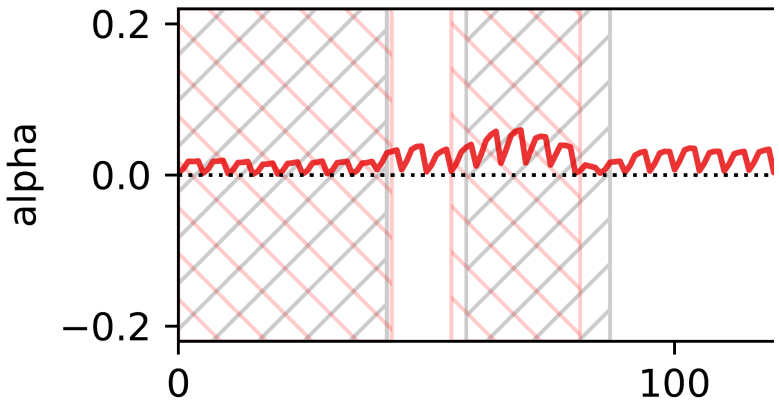


Figure 4.10 – Predicted warping value of an autoregressive decoder predicting a chunk of five frames per step. Each chunk shows a "loading" phase, where the first frame receives nearly no warping, then the warping raises quickly to a stable value for the last three frames. This behaviour shows the incompatibility of the APW with autoregressive decoders. The plot shows ground truth V/UV (grey, hatched upwards) and predicted V/UV (red, hatched downwards).

Parallel Decoder

Assuming an external duration model generating the correct alignments allows us to remove the iterative attention mechanism. This in turn opens up experiments with non-autoregressive models necessary because of the incompatibility of the APW with autoregressive models. We investigate a parallel decoder structure recently used by Karlapati et al. (2020a) and Qian et al. (2019). Details are not given by Karlapati et al. (2020a) thus we rely on the parameters in Qian et al. (2019). The parallel decoder consists of three 5×1 convolutional layers with 512 channels with ReLU activation followed by batch norm. Instead of three LSTM layers we use three bidirectional GRU layers with 1024 neurons to allow looking ahead. We use 50% dropout in the convolutional layers as in Tacotron and 10% dropout in the recurrent layers. As in the literature we do not use a post-net with this decoder.

APW model

We stack the APW with an alpha range of ± 0.2 on the parallel decoder similar to Figure 4.2. We pass the output of the last bidirectional GRU layer together with the speaker embedding to the APW layer. We compare this model (referred to as APW in the results) with the parallel decoder model without the APW (referred to as the baseline system) in the zero-shot speaker adaptation task.

4.5.2 Training

The model is trained with an L1 loss on the predicted acoustic features and a Kullback-Leibler (KL) term on the VAE parameters to push them towards a uniform Gaussian posterior. To

prevent posterior collapse we only take the KL term into account every 200 steps starting after a warmup phase of 25k steps. We train the model for 320 epochs (~160k steps) starting with a learning rate of 1E−4 in teacher forcing mode with the Adam optimiser ($\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 1\text{E}{-8}$, no weight decay). We use a plateau learning rate scheduler to reduce the learning rate by a factor of 0.1 on validation loss plateaus. Because we rely on fixed attention (see Section 4.5.1) the model generates features aligned with the target so that we can compute the validation loss without teacher forcing for more accurate results.

4.5.3 Zero-shot adaptation

For zero-shot adaptation we use speakers unseen during training from the test set of WSJCAM0. For each speaker we use the first sample (in alphabetic order) as input to the reference encoder. We assume that the reference sample is not transcribed, thus we do not apply any fine-tuning to the model. We then synthesise the remaining samples with oracle durations.

4.5.4 Subjective evaluations

To evaluate the impact of the APW we conduct two subjective listening tests. In the first test we ask listeners about their preference in terms of audio quality between the baseline and the APW model. In the second preference test listeners have to rate which of the models is closer to the same sample generated by copy synthesis in terms of speaker similarity. Both tests include a “no preference” option.

The WSJCAM0 test set contains five male and eight female speakers. We limit the listening test to samples between two and five seconds and a maximum of five samples per speaker (based on alphabetic order). Based on these two conditions we are left with two male speakers with four samples, one male speaker with two samples, and two male speakers with a single sample. We select a subset of five female speakers (again first in alphabetic order) with the same distribution. The resulting listening test consists of 12 male and 12 female samples from five different speakers each. 45 listeners rated nine randomly selected samples in each test.

The results show slight improvements in speaker similarity at the cost of audio quality (Table 4.4). The improvement in speaker similarity is more prominent for female speakers (+7.2%) which also show a smaller gap in audio quality (−1%). We found that the warping is especially used to generate female voices (Figure 4.11), which also shows that the warping is indeed used for speaker adaptation. For male speakers the improvement is smaller (+5.2%) and comes a the cost of a bigger audio quality drop (−4.1%). The drop in audio quality is not surprising. When the prediction comes closer to the target speaker it is moving further away from the training speakers and the exposure bias manifests itself in a drop of audio quality. The APW proves itself to increase the generalisability of the model in terms of speaker similarity.

Table 4.4 – Preference test on speaker similarity and audio quality for zero shot speaker adaptation on WSJCAM0 test speakers with 45 listeners and 9 samples per gender.

	Speaker similarity			Audio quality		
	Baseline	**APW**	**Same**	**Baseline**	**APW**	**Same**
female	22.1	29.3	48.6	29.3	28.3	42.4
male	25.7	30.9	43.4	36.1	32.0	31.9
combined	23.8	30.1	46.1	32.5	30.1	37.4

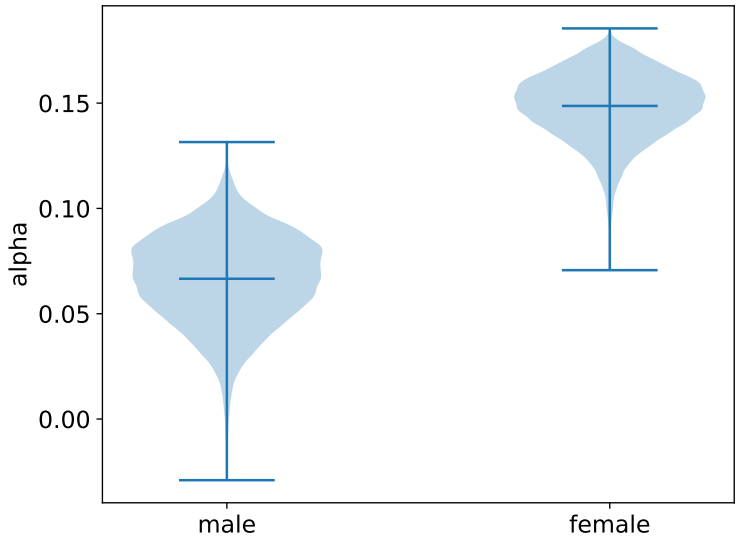


Figure 4.11 – Use of alpha per gender on the test test.

4.6 Experiment 3: Emotional TTS

This experiment is based on the observation that some emotions cause a shift of the first formant mean frequency. The emotion recognition community has shown that the analysis of vowel's formant frequency position allows detection of high arousal emotions in German (Vlasenko et al., 2011) and French (Bozkurt et al., 2011). We want to explicitly model this shift with the proposed APW, because it is an effective low-dimensional control for formant shifting. While we can offer the controllability through the APW, a-priori we do not know whether the model will be able to infer the correct locations to apply the formant shift from the textual input. If it can, we expect to improve the generalisability of emotional TTS models when trained on limited emotional data and thus improve audio quality and expressiveness.

4.6.1 Database Analysis

We use the SAVEE British English database (see Section 2.5.4) in this experiment. We work with phonemes as input and WORLD acoustic features with deltas and double deltas as targets (detailed feature description in 2.2). For mean/variance normalisation the parameters of the WSJCAM0 database are used, which facilitates transfer learning described below in Section 4.6.3.

As we base our work on the observation that some emotions cause a formant shift, we first analyse if this shift is also observable in the SAVEE database. We use the PRAAT speech analysis software (Boersma and Weenink, 2017)[4] to extract the first and second formant (F1 and F2) for vowel phonemes for the six different emotion (Figure 4.12). We draw the vowel triangle between the phonemes /ii/, /oo/, and /a/. The light grey triangle corresponds to neutral speech. One can see that for most emotions parts of the triangle were shifted. This is especially prominent for angry speech of speaker KL (Figure 4.13). We see the same F1 shift for angry speech as reported in Vlasenko et al. (2011). These observations show that emotions also cause a formant shift in English and that the SAVEE database is a suitable choice for our experiment.

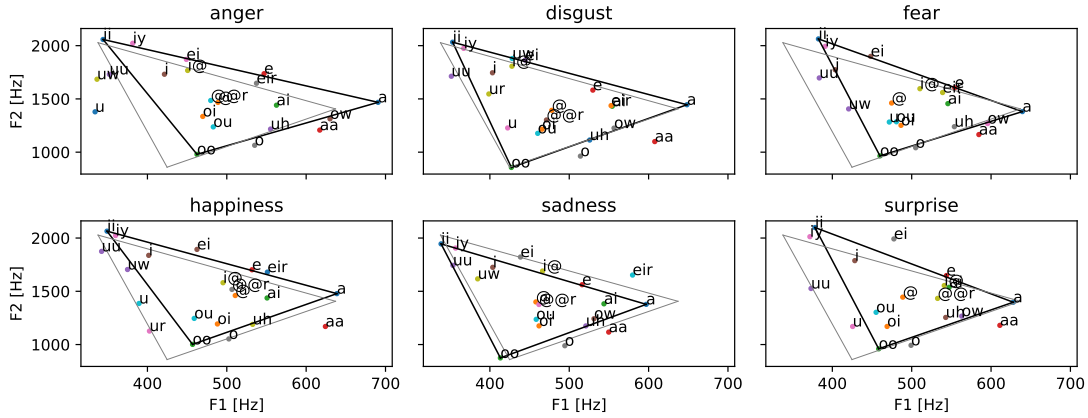


Figure 4.12 – Analysis of the first and second formant frequency in Hz of vowel phonemes for the six different emotions. Light grey corresponds to the vowel triangle of neutral speech.

4.6.2 Model architecture

Preliminary experiments showed that encoder-decoder models are not able to generate emotional speech from the limited amount of SAVEE data. Due to their high capacity they quickly overfit the training data before adapting to the new speaking styles. Thus we investigate only an RNN-baseline model of the old paradigm here. We use a commonly known RNN-based speech synthesis system (Zen et al., 2013) as the baseline system which has been used as well in recent studies of emotional speech synthesis (Lorenzo-Trueba et al., 2018; Henter et al., 2018). Henter et al. (2018) have compared supervised training of the baseline system with

[4] In combination with https://github.com/mwv/praat_formants_python.

unsupervised training of VQ-VAE-based (van den Oord et al., 2017) models on a Japanese single-speaker emotional database (Barra-Chicote et al., 2010). They found that the unsupervised learned representations achieve a slightly higher MOS of 0.13 in terms of perceived speech quality. Thus we believe it is still valuable to report results on the selected baseline system irrespective of the presence of newer VAE or encoder-decoder models. This RNN-baseline has two fully-connected layers with ReLU activation and 1024 neurons, three BiLSTM layers with 512 neurons, and a final 97 dimensional output layer. 5% dropout is applied in all but the final layer. All layers have a speaker and emotion embedding concatenated to their input.

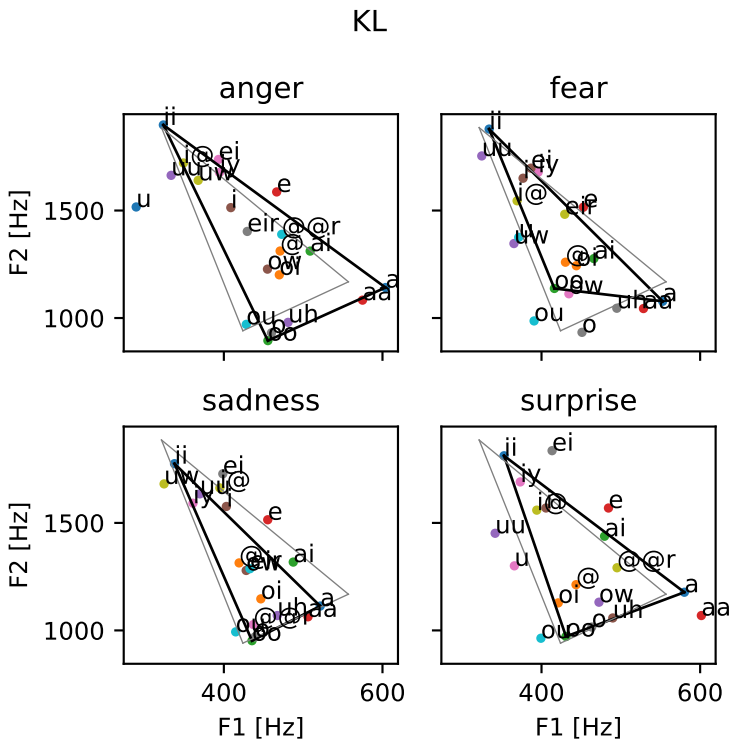


Figure 4.13 – Analysis of the first and second formant frequency in Hz of vowel phonemes for the four emotions most different from neutral of speaker KL.

We compare the RNN-baseline to the same model with an APW layer stacked on top. We will refer to this model as RNN-APW from here on. The RNN-APW architecture can be described well by Figure 4.2. All but the last 97 dimensional output layer are contained in the "Pre-Net" block, thus the last layer of the pre-net is a BiLSTM. The forward and backward output of the BiLSTM are concatenated with the emotion embedding and passed to the APW layer. Giving the emotion embedding only to the APW layer does not work because other features like LF_0 need to change with the emotion as well. Instead emotion and speaker embeddings are given to all layers in the pre-net. In contrast, the APW only receives the emotion embedding and the intermediate representation generated by the last layer of the pre-net. No speaker information is explicitly given to the APW to prevent it from speaker adaptation.

4.6.3 Training

To train a modern TTS system the SAVEE database does not provide a sufficient variety of words, i.e. it is too small. Thus we first pre-train on the WSJCAM0 database (database details in Section 2.5.3) to obtain a good TTS system. We train the model with a batch size of 16 for 35 epochs with early stopping and a learning rate of 0.001. The learning rate is reduced by a factor of 0.1 on validation loss plateaus.

We split the emotion adaptation into two steps: adaptation to SAVEE neutral and adaptation to SAVEE emotional. As we are only interested in the impact of the APW on emotional TTS, we add it only in the second step.

First, starting from a pre-trained model on WSJCAM0, we adapt only to the neutral part of the SAVEE database. This allows the model to learn the unseen speaker identities and differing environmental conditions. We follow a three step transfer learning procedure inspired by Chen et al. (2019). At first we train only the speaker embedding (10 epochs, lr=0.001), then we train the whole model (10 epochs, lr=0.001), at last we train the whole model with a reduced learning rate (10 epochs, lr=0.0001). In each step we use early stopping and continue with the best model. A batch size of 16 is used in all steps. Training only the speaker embedding does not give good results. We assume it is because the recording conditions of the two databases are different. Environmental conditions like microphone noise and reverberations are consistent throughout all speakers of the same database and thus can be encoded in the network weights. Adapting to a new database is therefore only possible by fine-tuning the whole model. The resulting model is capable of synthesising the four SAVEE speakers in neutral speech. This model forms the starting point for the second step: the adaptation to the emotional part of the SAVEE database. We compare two models on this task: 1) the unaltered RNN-baseline and 2) RNN-APW, where we add the APW with an alpha range of ± 0.1. We adapt both models with the same three stage transfer learning procedure as above, but this time using the entire SAVEE database.

4.6.4 Results

We find that the RNN-APW model gives slight improvements for a few samples, but in general does not outperform the RNN-baseline. We observe that the model is not making much use of the warping. We have tried to encourage the model to make better use of the warping with the following techniques:

- Different alpha ranges (± 0.02, ± 0.05, ± 0.2): Convergence might be better when the range matches the maximum warping useful for emotion adaptation so that the predictions are further away from the steep part of the TanH.

- Speaker embedding as additional input: With additional speaker information the APW should be able to predict a speaker and phoneme dependent warping.

- Scale alphas during inference: The predicted warping value gives an unprecedented control to change the cepstrum. To increase the effect of the warping we scaled it globally by up to 1000%. However, we found that the scaling did not result in more affective speech, but instead became unnatural after about 500% scaling. Sparser scaling might give the desired effect but we currently do not have a method to predict the right positions for it.

- Higher learning rate for APW layers: Directly after initialization, the APW brings only more distortion in the cepstrum for neutral speech. A faster training of the APW layers should make them useful much quicker so that the model does not converge to the "no warping" local optimum.

- Gradient scaling at alpha: By increasing the gradient at the alpha prediction stage the rest of the network receives more gradient from the APW branch, so that it adapts more to it.

- Use APW for speaker and emotion adaptation: The amount of emotional data might not be sufficient to learn to use the APW for formant shifting. Instead the APW can already be used for speaker adaptation (in the first adaptation step), then, when adding emotional samples (in the second adaptation step), the already known technique for speaker adaptation can be used in a smaller quantity for emotion adaptation.

However, none of the techniques has changed the converged model positively. Some have degraded the signal quality instead. Given the essentially negative result, we do not attempt to compare the adaptation performance to other techniques. Instead, we attempt to understand what the transform has and has not learned in order to know where to direct future research.

4.6.5 Statistical analysis of the warping per phoneme

Even though the APW does not improve the model, we found that the warping is still partly used. In this section we take a closer look on the statistics of the warping on a per-phoneme-basis. We found that even neutral samples receive some warping. From the phoneme alignments we can collect the warping values per phoneme and analyse them in a violin plot (Figure 4.14). The warping on neutral samples seems to position the phonemes within the vowel triangle (compare Figure 4.12). It can be observed that phonemes as /o/, /oi/, /oo/, and /ou/ are warped negatively, moving them down to the lower left corner of the triangle, while /a/, /aa/, and /ai/ receive positive values. This indicates that a part of the warping is used for the phoneme positioning within the vowel triangle.

However, we also observe emotion dependent patterns, which we analyse in the following. Emotions can be represented as categories, but also in a continuous space of valence and arousal. Arousal is often explained as alertness or "level of activity". Valence corresponds to the attractiveness/averseness of something. Thus high valence emotions are positive emotions,

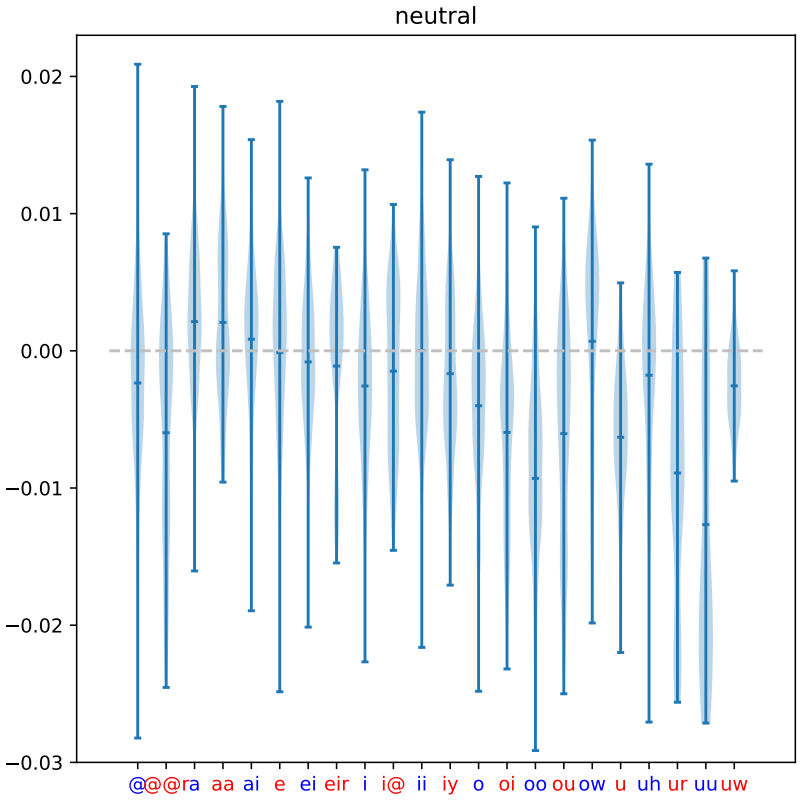


Figure 4.14 – Warping per vowel phoneme for neutral speech on SAVEE.

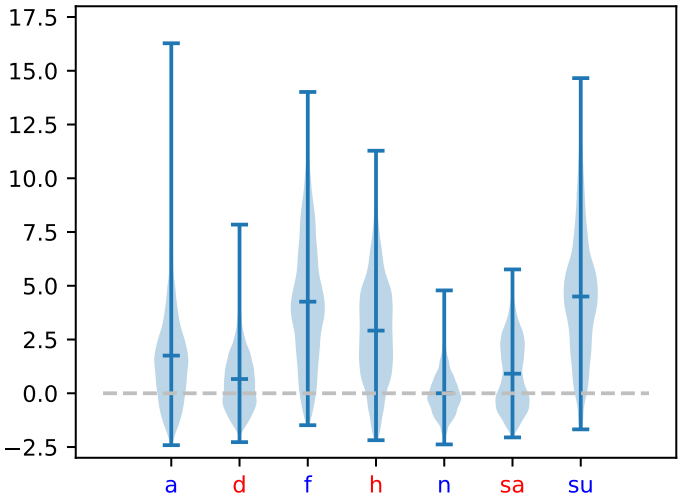


Figure 4.15 – F_0 statistics mean/variance normalised w.r.t. neutral for the six emotions (a: anger, d: disgust, f: fear, h: happiness, n: neutral, sa: sadness, su: surprise). High arousal emotions (anger, fear, happiness, surprise) show higher F0 mean, variance, and range.

e.g. happiness/joy. We do not find correlation between the level of arousal and level of warping. It is well known (Goudbeek et al., 2009; Banse and Scherer, 1996; Johnstone and Scherer, 2000) that arousal manifests itself primarily in a change of F0 mean, variance, and range. We compute the F0 statistics per emotion and mean-variance normalise them w.r.t. neutral (Figure 4.15). We see the expected higher mean, variance, and range for the high arousal emotions (anger, fear, happiness, surprise). This shows that arousal manifests itself in F0.

To investigate correlation between the warping an the level of valence in the emotion we group the categorical emotions into low (anger, disgust, sadness, fear) and high (happiness, surprise) valence emotions and compute how much vowel phonemes are affected in terms of mean F1 shift compared to neutral (Figure 4.16 left). We then compute for which phonemes the difference in mean F1 is the most between low and high valence emotions (Figure 4.16 right). From those phonemes we select the eight with the highest difference (/i@/, /@@r/, /aa/, /ei/, /@/, /oo/, /ow/) and study their received warping. If the warping correlates with valence, we expect to see the most warping difference on these eight phonemes.

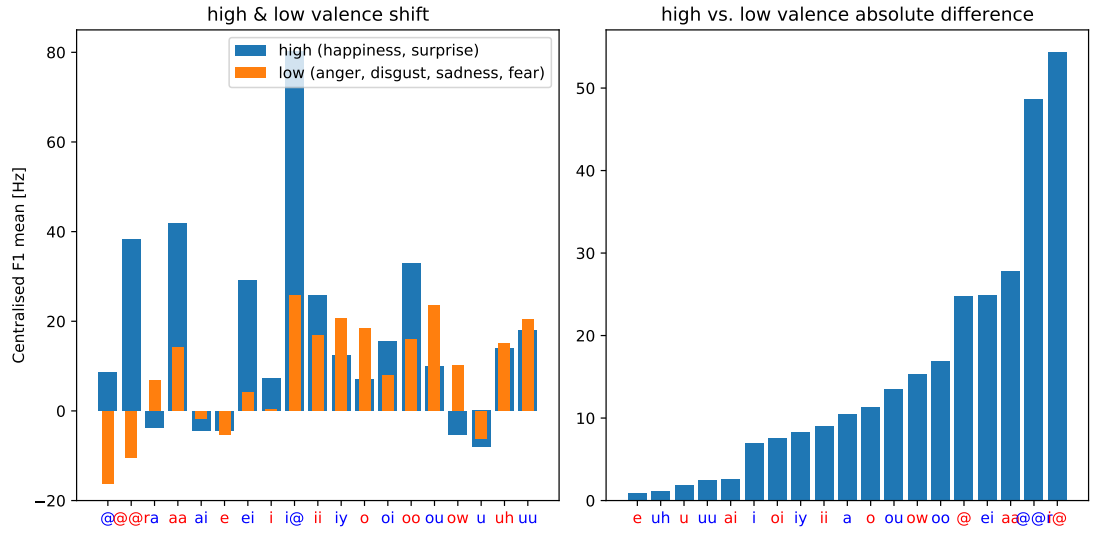


Figure 4.16 – Left: Average F1 shift between low/high valence emotions compared to neutral speech. Right: Phonemes ordered by absolute difference of the F1 shift between low and high valence emotions.

Indeed, we see that the average warping of all eight phonemes (red line in Figure 4.17) corresponds to the level of valence in the emotion. We observe high warpings for the high valence emotions, but small or negative values for the low valence emotions. The differences are statistically significant (the p-value is displayed in the figure if it exeeds 0.05 in a two-sided t-test). The warpings are normalised w.r.t. the warping on neutral, so that the phoneme positioning warping, described above, is not visible.

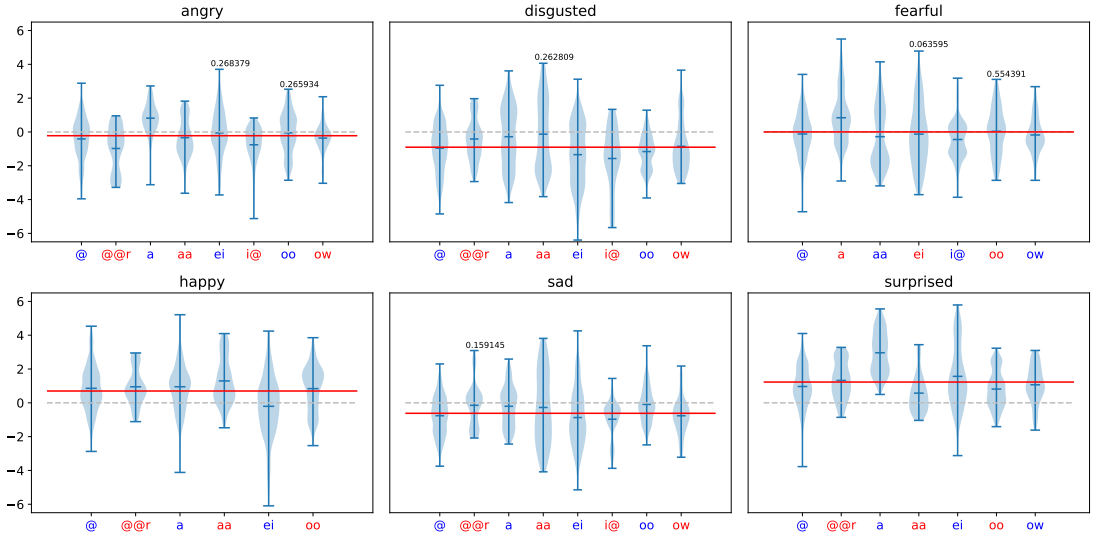


Figure 4.17 – Normalised warping of the eight phonemes, which are most affected by valence, for the different emotions. The warping is mean/variance normalised with the warping on neutral speech to remove the positioning effect described above. The red line indicates the average warping over all eight phonemes. p-values of a two-sided t-test are displayed when $p > 0.05$.

From our analysis we conclude that the warping is used to position the phonemes within the vowel triangle and also correlates with the level of valence in the emotion, even though it does not improve the overall model performance. From the analysis we develop the hypothesis that the APW can be used to increase the valence in an utterance, but the model is not able to predict it at the correct points in the utterance. This shortcoming is rooted in an independent problem. While the emotion of an utterance is not present in every word/phoneme, the database labels the whole utterance as one emotion. The model now has to infer which parts are actually emotional, which is impossible from the limited amount of data.

4.7 Conclusion

In this chapter we proposed a neural All Pass Warp (APW) inspired by VTLN, which allows efficient training and inference and is less data-hungry due to its small parameter space. On an artificial speaker with known warping parameters compared to a base speaker, we showed that the network can learn the ground truth time-dependent warping parameters. With the APW at hand we investigated three common scenarios of today's TTS research:

1. We showed that the technique improves objective scores of a **multi-speaker model** and objective and subjective scores on a **few-shot speaker adaptation** task. Even though we tested the proposed technique on an old-paradigm acoustic model we argue that it is applicable in the same way to state-of-the-art encoder-decoder models.

2. The APW by design generalises over different speakers, thus we expected that it would improve the generalisability of multi-speaker models, leading to improved speaker similarity and/or audio quality in a **zero-shot speaker adaptation** task. This hypothesis was demonstrated; listening tests showed superior speaker similarity at a small cost of audio quality.

3. Emotions cause a formant shift, which can be modelled explicitly with the APW. We expected to improve the expressiveness and audio quality in **emotional TTS**. This hypothesis was not demonstrated; the warping is not used much. However, our analysis shows that it correlates with the level of valence in the emotion, proving that the model learned what was intended. As other parts of the network can learn emotion as well, we assume that the APW gets swamped by their effect. Manual changing of the warping will alter the valence, but neither we nor the model, are currently able to infer the correct locations to do so. Rather, we assume that a dialogue or translation agent will be able to detect and reproduce them.

The somewhat negative results on emotional TTS suggest two future research directions: increasing 1) the quantity and 2) the quality of the emotional training data. On the quantity side the generation of synthetic data is a good candidate, this has already shown to be effective for expressive speaking styles (Huybrechts et al., 2020). We will investigate it in Chapter 6. On the quality side we will present a technique in Chapter 5 to infer additional localised features from the emotional data, which are then provided as additional inputs to the TTS model. During inference those features would need to come from a dialogue or translation agent. Although we cannot evaluate on translation directly, we mean to try to break the chicken-egg problem where the agent cannot be evaluated without the means to alter the acoustics, but the means cannot be evaluated without the agent. Rather, we will present the means, and characterise it, as we did in this work, in the hope that it may be used in work on agents.

5 Emotion Intensity

In the last chapter we found that, while it correlates with the level of valence in the emotion, the APW is not able to improve the emotional speech synthesis system. We hypothesise that the model is not able to properly identify the locations in the utterance to apply to. Many emotions are not displayed continuously in an otherwise emotional utterance; rather, the intensity varies with time. Emotional databases usually have a single emotion label for every recording. We argue that this generalisation is misleading and that the emotion is localised within the utterance. Depending on the context the localisation itself can vary despite having the same linguistic content and emotion. This kind of annotation can lead to different emotion labels on words with lower emotional strength like conjunctions, while their acoustic features only marginally differ. Obviously, this impedes the learning of the model. If we can provide this information to the network, we increase the quality of the training data, which should facilitate generalisability and thus increase emotion intensity and/or audio quality for low data regimes.

In this chapter we propose to add a frame-level single dimensional emotion intensity to every sample, which is used as additional input to the TTS model. The intensity provides a simple and understandable control to the network. We present two methods to extract this information from the recordings with pre-trained emotion recognisers. The simpler model contains a single attention layer, which allows use of the attention weights as emotion intensities. The other is a modern transformer model, where we exploit saliency maps to extract the intensity. We show that an emotion recogniser is capable of producing a measure of emotion intensity via attention or saliency; this measure is appropriate to label utterances subsequently used to train a speech synthesiser. We evaluate novel and published means to do this showing that, whilst it is no longer state of the art for emotion recognition, attention is a good way to indicate emotion intensity for speech synthesis. In this work we leave out the problem of generating the emotion intensity from text or extraction from a reference sample. Possible research directions to attack this problem are listed in the conclusions in 5.4.

The majority of the text in this chapter will appear at:

- Schnell, B. and Garner, P. N. (2021a). Improving emotional TTS with an emotion intensity input from unsupervised extraction. In *Proc. 11th ISCA Speech Synthesis Workshop*

5.1 Background

While TTS systems have mastered human performance for neutral speech, emotional speech synthesis is still a challenge. For neutral speech large and high quality databases exist, but emotional databases are rare and mostly of low quality and most were recorded with speech or emotion recognition tasks in mind. Thus their quality rarely fulfils the high requirements of modern TTS systems. It is certainly possible to record large amounts of a specific emotion and train the same systems as used for neutral speech. However, the range of emotions, varying intensities, the amount of languages, speaker variations, and the need to label each recording with the perceived emotion of multiple listeners makes recording alone a nearly infeasible task in terms of time and money requirements. Modern emotional TTS research has identified three possible directions to solve these problems. We will highlight some recent work for each direction:

1. Increase the generalisability of neural network architectures to increase performance on low data regimes.
2. Increase the quantity of emotional data by voice or emotion conversion.
3. Increase the quality of the emotional data.

Databases with more expressive speech than plain neutral exist, especially in the form of audio books. While those databases cover a wider range of speaking styles, they lack any annotation of the expressed emotion or style. The lack of these annotations spawned a wide range of recent works focusing on increased model *generalisability* by utilising unsupervised methods to extract style embeddings from reference audio on a global (Wang et al., 2018b; Skerry-Ryan et al., 2018; Cooper et al., 2020; Gururani et al., 2019; Bian et al., 2019), clustered (Klimkov et al., 2019; Sun et al., 2020), or frame level (Choi et al., 2020; Lee and Kim, 2019; Shechtman et al., 2021). Some have experimented with controlling the expressiveness through manipulation of the latent embeddings (Wang et al., 2018b; Um et al., 2020). A detailed review of all of these methods was already given in 4.1. However, controllability remains limited, especially for global style embeddings.

Some work has targeted increasing the *quantity* of the expressive training data. Huybrechts et al. (2020) have used voice conversion (CopyCat, detailed description in 6.1) to convert expressive (conversational and newscaster) recordings of different speakers to a target speaker. The source speaker provides large amounts of the desired style (7 hours conversational or 5

hours newscaster). The Voice Conversion (VC) model is trained with neutral and expressive recordings of the target speaker and one or multiple source speakers. Large amounts of neutral data are available for all speakers (~20 hours). The TTS model is then trained with the neutral, expressive, and converted samples of the target speaker. In a last step the TTS model is fine-tuned on the expressive recordings. The increased data quantity improved signal quality as well as style adequacy in subjective listening tests. This work assumes that large databases of the desired style of another same-gender speaker exist. In the next Chapter 6 we propose a neutral to emotional speech conversion model to address this limitation.

Xu et al. (2020) use an iterative process of TTS and ASR to improve both models on low resource languages. They start with pre-training on rich-resource languages and then fine-tune on the low-resource language with the phoneme/character and speaker embeddings all reinitialised. After the two steps both models still have issues. The authors propose to use unpaired data, i.e. text without speech (unpaired text) and speech without text (unpaired speech) of the target language, in a dual transformation between TTS and ASR to further improve the models. This data is usually easily obtained from the media. From unpaired text synthetic speech is generated. The ASR system is then used to filter out samples with word skipping and repeating issues. The resulting corpus is used to retrain the TTS model. The ASR system is also used to create text for the unpaired speech, the resulting paired corpus is combined with the synthetic speech generated from unpaired text and the low amount of paired data in the target language to retrain the ASR model. The dual transformation is iterative and applied on the fly so that it improves over time with advances in both models. The use of synthetic data improves intelligibility and audio quality of the TTS model, but also word error rate and character error rate of the ASR model.

We found a limited amount of work which attempts to increase the *quality* of the emotional data used during training. Emotional databases usually have a single emotion label for every recording. We argue that this generalisation is misleading and that the emotion is localised within the utterance. The closest work to ours is that of Zhu et al. (2019). They use relative attributes (Parikh and Grauman, 2011) to assign a level of emotional strength to each sample. In more recent work (Lei et al., 2020) they extended their method to phoneme level emotional strength. We will describe their method in the following.

5.1.1 Attribute Rank

Recent work (Zhu et al., 2019; Lei et al., 2020) has used attribute ranks (Parikh and Grauman, 2011) to compute emotion intensities. We include this work as a competitive method here and give a brief overview. For data of two categories the ranking function computes the ranking/order of the data w.r.t. to a certain attribute, here emotion intensity. Once the ranking function is learned, it can assign an emotion intensity level to unseen emotional data. For completeness we give an example closely following that in Lei et al. (2020).

We select all neutral N and happy H samples from the training set with acoustic features x_t with $t \in [1,\dots,T]$ with $T = |N \cup H|$. We then form an ordered set O and an unordered set S of pairs. In the ordered set we pair an emotional sample of H with a neutral sample from N, indicating that the emotion intensity is higher in the samples of H than in those of N. In the unordered set we randomly create pairs of neutral-neutral and happy-happy samples, indicating that their rank should be similar. The goal is to learn a ranking function

$$r(x_t) = wx_t \tag{5.1}$$

satisfying the following constraints as much as possible

$$\begin{aligned} &\forall (i,j) \in O : wx_i > wx_j \\ &\forall (i,j) \in S : wx_i = wx_j \end{aligned} \tag{5.2}$$

The problem can be relaxed with slack variables ξ_{ij} and γ_{ij} and a controllable trade-off C (Parikh and Grauman, 2011) and solved by Newton's method (Chapelle, 2007) similar to Support-Vector-Machines.

$$\begin{aligned} \min \quad & \left(\tfrac{1}{2}||w^T||_2^2 + C(\textstyle\sum \xi_{ij}^2 + \sum \gamma_{ij}^2)\right) \\ \text{s.t} \quad & w^T(x_i - X_j) \geq 1 - \xi_{ij}; \forall (i,j) \in O \\ & |w^T(x_i - x_j)| \leq \gamma_{ij}; \forall (i,j) \in S \\ & \xi_{ij} \geq 0; \gamma_{ij} \geq 0 \end{aligned} \tag{5.3}$$

In Lei et al. (2020) a single openSMILE (see Section 2.2) feature vector x_t is extracted for each utterance. Then the ranking function, i.e. the ranking vector w_m with $m \in [1,\dots,M]$, is learned for each combination of neutral with the other M emotions. To obtain phoneme-level rankings openSMILE features are extracted for the segments corresponding to each phoneme. Obviously this requires a forced-alignment step. We use the Montreal Forced Aligner (McAuliffe et al. (2017), Section 2.2) for it. In Lei et al. (2020) the authors use a 14-hour single speaker Chinese database, of which one third is emotional speech divided into six emotion categories. The remaining two thirds are neutral samples. To learn the ranking function they use only a subset of all samples.[1] To form set O they randomly select 300 neutral sentences and pair them with 300 emotional sentences, which allows that one emotional sentence is paired with multiple neutral ones. To form S 150 pairs are randomly selected in the neutral and emotional set each. We use a Python port[2] of the original code[3] of Parikh and Grauman (2011) with the default parameters for the Newton algorithm.

[1] We thank Shan Yang for the detailed description of the process.
[2] https://github.com/chaitanya100100/Relative-Attributes-Zero-Shot-Learning
[3] https://www.cc.gatech.edu/~parikh/relative.html

5.2 Emotion intensity extraction

In this section we present our two techniques to extract an emotion intensity from audio with the help of emotion recognisers.

5.2.1 Attention LSTM

We use a simple emotion recogniser mostly resembling previous research (Ramet et al., 2018) (Figure 5.1 left). It consists of a feature extraction part of three fully-connected layers with 256 neurons and a BiLSTM with 128 neurons per direction. We apply dropout with a probability of 0.1 after each layer. Additionally, it contains an attention block with a single BiLSTM with 128 neurons per direction and a fully-connected layer without bias with a single output neuron. The outputs of the BiLSTM in the attention block correspond to the keys in recent attention terminology, and the weights of the fully-connected layer represent the query, which acts as a task embedding. The output of the fully-connected layer respresents the unnormalised attention weights. As in the previous work (Ramet et al., 2018) we use a sigmoid activation, instead of the usual softmax, to obtain normalised attention weights. The reasoning was that a sigmoid activation ensures high activation levels over many frames, which leads to overall smoother attentions. This is especially desirable for our downstream task of emotional TTS. We use the predicted attention weights to compute a weighted sum over the outputs of the feature extraction part (values) to create a single utterance level embedding of size 256. We pass this vector through a single fully-connected layer with as many neurons as emotion classes. All parameters are initialised using Xavier initialisation (Glorot and Bengio, 2010) with a uniform distribution, with one exception: The weights of the fully-connected layer with single output neuron in the attention block are initialised with samples from $\mathcal{N}(0, 0.1^2)$.

The openSMILE toolkit (Eyben et al., 2013) is used to extract frame-level features with a sliding window of 25ms and a shift of 10ms. We use a 32-dimensional subset of the *IS09* feature subset composed of hand-crafted Low-Level Descriptors (pitch, energy, zero-crossing rate, voicing probability), 12 MCEP coefficients, and their first derivative. This subset is mean-variance normalised and forms the input to the emotion recogniser. To prevent overfitting we augment the input with random white noise with a standard deviation of 0.4. In contrast to previous research, this model is not limited to inputs with 500 frames; it accepts variable input lengths.

Training closely follows the procedure in previous work (Ramet et al., 2018). The model is trained with the Adam optimiser (Kingma and Ba, 2015) ($\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 1\text{E}{-8}$) with a learning rate of 3E−5 on mini-batches of 32 utterances for 200 epochs with the cross-entropy loss. To account for class-imbalance we weight the cross-entropy for each class c by a factor of $w_c = \frac{N_{tot}}{N_{classes} N_c}$, where N_{tot} is the total number of training utterances, $N_{classes}$ the number of different classes/emotions, and N_c the number of utterances of class c in the training set. All but the LSTM layers are regularised with l_2-regularisation with a factor of 5E−2. We select the best model based on the summed Weighted Accuracy (WA) and Unweighted Accuracy (UA).

We argue that this emotion recogniser, once trained, will attend to the emotional parts of the utterance to make a decision. Thus it is reasonable to assume that the attention weights over an utterance give a good approximation of the emotion intensity. By forwarding each utterance of the database through the model, we can obtain the emotion intensity value by saving the predicted attention weights.

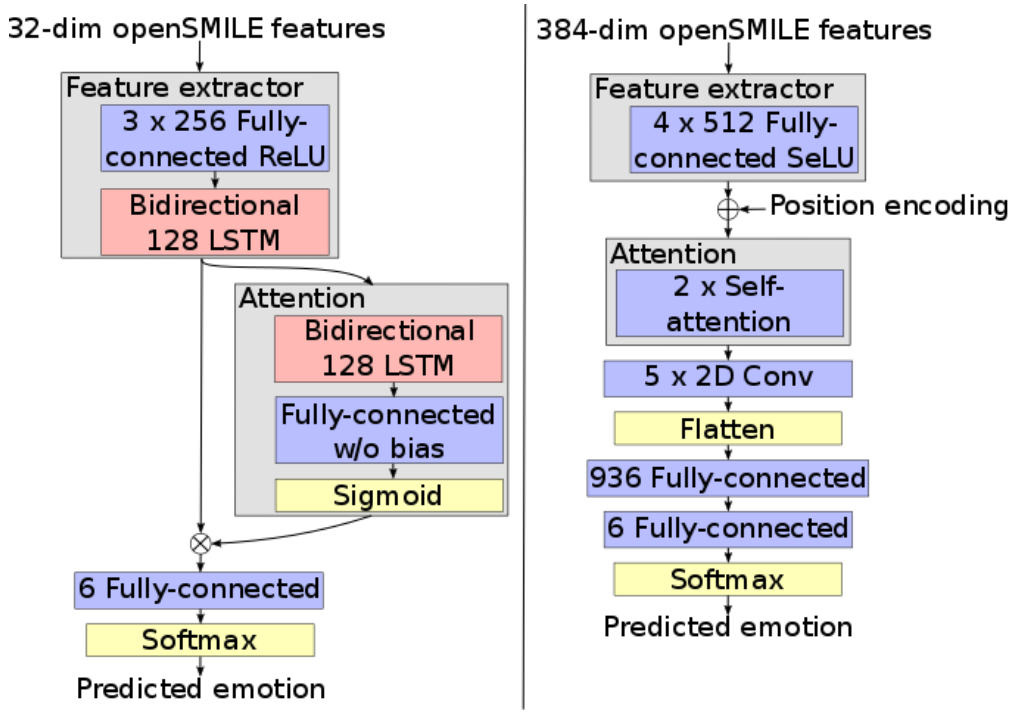


Figure 5.1 – Architectures of the emotion recognisers. Left: attention LSTM; right: transformer

5.2.2 Transformer

The above model is a very simple emotion recogniser and does not represent the state of the art. More complex architectures exist which do not allow a straight forward extraction of attention weights as described in Section 5.2.1. In this section we investigate a more recent transformer model (Tarantino et al., 2019). It consists of multiple self-attention blocks (Vaswani et al., 2017), which do not allow the extraction of attention weights in an obvious way. We make no claim that this model is the best emotion recogniser currently available (newer exist e.g. Latif et al. (2020)); rather, we present a technique representative of more complex models without restrictions to their architecture to extract emotion intensities.

The transformer model uses Scaled Exponential Linear Unit (SELU) activation. SELU activation is an attempt to omit the dead state problem in ReLUs, which appears when all activations are stuck in the negative orthant. A neuron affected by this will only receive zero gradients and is effectively dead. The SELU is the non-linearity of choice for a new kind of neural networks called *Self Normalizing Networks* (Klambauer et al., 2017). They are designed so

that for inputs sampled from a normal distribution, their output will also follow a normal distribution. This is achieved by using $\alpha =\sim 1.6732$ and $\lambda =\sim 1.0506$ in Equation 5.4. In that sense they self-normalise, which makes other normalisation like batch norm redundant, the second important property compared to other activation functions. While not standard in NN research we mainly use it to closely follow the previous work in Tarantino et al. (2019).

$$\text{SELU}(x) = \lambda \begin{cases} x & \text{if } x > 0 \\ \alpha e^x - \alpha & \text{if } x \leqslant 0 \end{cases} \tag{5.4}$$

The transformer (Figure 5.1 right) consists of a feature extraction block with four fully-connected layers with 512 neurons and SELU activation. Afterwards a positional encoding is added in form of a sinusoid with a large period. Dropout with 0.1 probability is applied on the latent feature representation, which is then fed to two self-attention blocks with 32 heads each. The resulting attention matrix is aggregated with five 2D convolutional layers with [30, 30, 30, 10, 6] output channels, a 5×5 kernel size, and a stride of 2×2. The flattened 936-dimensional output is projected with a fully-connected layer with 936 neurons and a final fully-connected output layer with as many neurons as emotion classes. After each but the last layer in the aggregation step, dropout with probability 0.2 and SELU activation is applied. All parameters are initialised using Xavier initialisation (Glorot and Bengio, 2010) with a uniform distribution.

As before we use the openSMILE toolkit to extract frame-level features of 25ms windows and 10ms shift. However, we use the entire 384-dimensional *IS09* features subset as input to the transformer model and we do not add any noise. The transformer model requires a fixed-length input. We use a sliding window of 500 frames with a step size of 50 frames previously found to perform best (Tarantino et al., 2019). At inference time the final prediction is made by applying a softmax on the predicted classes of each window and averaging the results. Sequences are zero padded to match the window and step size, no frames are dropped.

During training we randomly select 500 frames from each input in the batch. We use the Adam optimiser ($\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 1\text{E}{-8}$, no weight decay) with a learning rate of $1\text{E}{-5}$ for 170 epochs on a mini-batch size of 8 and PyTorch's *ReduceLROnPlateau* scheduler with default parameters.

To extract emotion intensities with the transformer model we propose to use saliency maps. Saliency maps are a common technique in vision-related machine learning tasks. They attempt to add interpretability to the neural network predictions. Saliency maps allow interpretation of inputs, but also of layers, or even neurons. In this work we are only interested in the interpretation of inputs. An increasing number of techniques exist with varying complexity : Input Gradients (Simonyan et al., 2013), Smoothgrad (Smilkov et al., 2017), Input X Gradient (Shrikumar et al., 2016), Integrated Gradients (Sundararajan et al., 2017), Grad-cam (Selvaraju et al., 2017), Full-Gradient (Srinivas and Fleuret, 2019), and more. Saliency maps compute the importance of each input to the network's output, thus each openSMILE feature in each frame receives a value. To compute a scalar emotion intensity value we investigate the aggregation

through *max* and *mean* operations. In the following we will give a high-level description of the techniques we use in our experiments (Section 5.3).

Saliency Maps

Input gradients (Simonyan et al., 2013) continues the backpropagation chain to the inputs and thus provides gradients of each input w.r.t. the correct class label. Theoretically one can compute the input gradients for any other class as well. The idea is that the gradients indicate how much the class prediction is affected by a change in each input, thus representing its importance.

Since *input gradients* produces relatively noisy saliency maps **Smoothgrad** (Smilkov et al., 2017) attempts to smooth them over multiple observations. It achieves this by adding white noise to the input multiple times and computes the average input gradients for all iterations. The idea of *Smoothgrad* can be applied in many other saliency map techniques.

Input X Gradient (Shrikumar et al., 2016) first computes the *input gradients* and then multiplies them with the input itself. The idea is that the gradient alone only indicates how important the feature is, but the input gives information on how strongly the feature is present. Together they provide a better abstraction of the feature importance.

Integrated Gradients (Sundararajan et al., 2017) is a more involved technique. Consider a specific input dimension, which has a major effect on the prediction once it reaches a threshold. When we compute the *input gradients* close to the threshold it will receive major importance. However, once the strength in this feature increases, the gradient on it will become small, because small pertubations do not cause a change of prediction anymore. However, the dimension is still of major importance for the predicted class. *Integrated Gradients* attacks this problem by considering the *input gradients* computed on a linear interpolation between a baseline and the actual input. The baseline in images is often a black or white image. In our case this simply corresponds to zeros in all inputs. If at any point between the baseline and the actual input the feature has contributed to the class prediction, *integrated gradients* captures it as it computes the integral over all steps.

5.3 Experiments

We compare all three methods (attention weights, saliency map, and attribute rank) and a baseline without intensity input on the task of emotional TTS. For our experiments we select the SAVEE database (see Section 2.5.4). To train emotion recognisers, the SAVEE database is rather limited. Thus we include the IEMOCAP database (see Section 2.5.5) in all strategies for emotion intensity extraction. It is a commonly used database for speech emotion recognition (Mirsamadi et al., 2017; Ramet et al., 2018; Tarantino et al., 2019). While still in the database we exclude samples where no majority label was found, additionally we exclude the 'disgusted' emotion from our experiments, as it is both very hard to express and very rare in the database.

5.3.1 Emotion Intensity

Emotion Recogniser

We train the attention LSTM (Section 5.2.1) and transformer (Section 5.2.2) emotion recogniser models on IEMOCAP with the parameters and inputs as defined in their respective section, using a random split of the 5th session for the validation and test set. We then fine-tune the models on SAVEE with the same parameters for 200 epochs and select the best model based on combined WA and UA on the validation set. The dataset is organised with numerical ids. For each emotion we select emotion specific utterances as test and validation set. Namely we use the 4th and 5th id as test set and the 6th and 7th id as validation set. We select the same ids for all speakers so that the content is unseen during training. Table 5.1 shows the metrics of the trained models on IEMOCAP and SAVEE. With the trained models we extract the emotion intensity. For the attention LSTM model these are simply the attention weights.

Table 5.1 – Weighted Accuracy (WA) and Unweighted Accuracy (UA) of the emotion recogniser models after pre-training on IEMOCAP and fine-tuning on SAVEE excluding the disgusted emotion class.

	IEMOCAP		**SAVEE**	
	WA	UA	WA	UA
Attention LSTM	54.7	40.3	62.5	60.4
Transformer	51.2	43.1	69.6	67.7

For the transformer model the variety of saliency maps (Section 5.2.2) allows multiple intensity curves (Figure 5.2). We extract emotion intensity using Input Gradients, Smoothgrad, Input X Gradient, and Integrated Gradients with *max* and *mean* aggregation. As the saliency maps can be very noisy, we also experiment with smoothed versions obtained by a simple convolution with an 11 frames wide Hanning window (Figure 5.3). More saliency map plots can be found in Appendix B. With informal listening we cannot select a clear best system. However, we find that a priori the intensity weights extracted with the attention LSTM model consistently produce more expressive speech than those extracted with saliency maps. To detect a difference between the two methods we decided to use the saliency map closest to the intensity weights extracted with the attention LSTM model, knowing they are the ones most difficult to distinguish. For that reason we extract the saliency maps on the attention LSTM model and compare them to the attention weights in terms of MSE. As can be seen in Table 5.2 the closest saliency map is smoothgrad with *mean* aggregation and smoothing.

Attribute Rank

While it would be possible to learn the ranking just on the SAVEE database, we also include the IEMOCAP database for a fair comparison. Indeed, we found that rankings extracted in combination with IEMOCAP outperform those learned only on SAVEE in informal listening

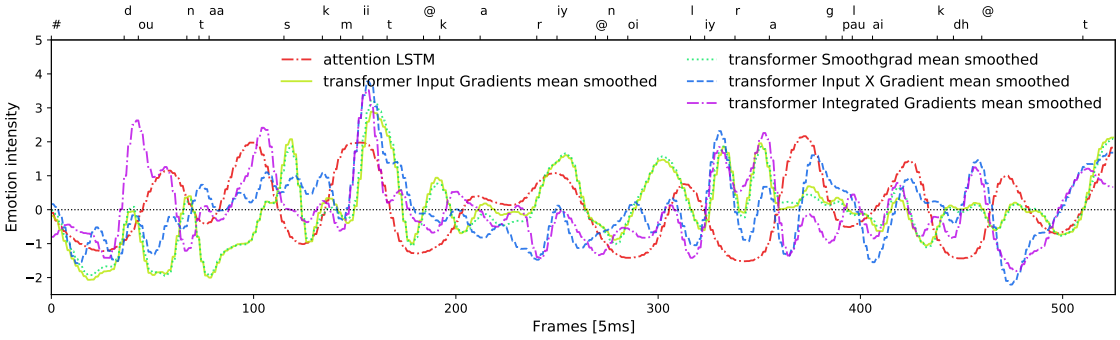


Figure 5.2 – Emotion intensities extracted with the attention LSTM model and different smoothed saliency maps for an angry utterance of speaker JK. For better comparison each intensity is mean-variance normalised based on its own statistics. The content is: “Don’t ask me to carry an oily rag like that.”

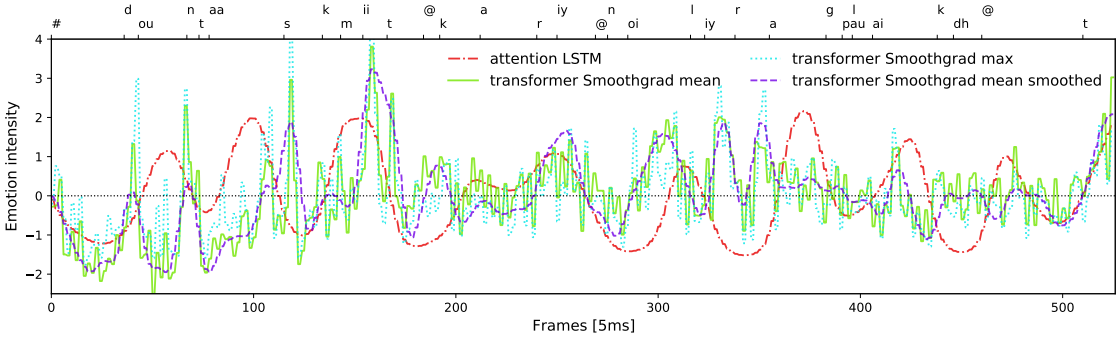


Figure 5.3 – Emotion intensities extracted with the attention LSTM model and with the Smoothgrad saliency map with max and mean aggregation as well as smoothed mean for the same utterance as in Figure 5.2.

Table 5.2 – MSE between saliency maps and attention weights extraction on the attention LSTM model on SAVEE.

Saliency map	Aggregation	Smoothed	MSE
Input Gradients	mean	No	1.46
Input Gradients	mean	Yes	0.625
Input Gradients	max	No	1.466
Input Gradients	max	Yes	0.665
Smoothgrad	mean	No	1.451
Smoothgrad	mean	Yes	**0.621**
Smoothgrad	max	No	1.461
Smoothgrad	max	Yes	0.664
Input x Gradient	mean	No	1.626
Input x Gradient	mean	Yes	1.179
Input x Gradient	max	No	1.5
Input x Gradient	max	Yes	0.835
Integrated Gradients	mean	No	1.62
Integrated Gradients	mean	Yes	1.312
Integrated Gradients	max	No	1.56
Integrated Gradients	max	Yes	0.984

tests. We exclude the SAVEE samples later used for validation and test set of the emotional TTS model (Section 5.3.2) when learning the ranking function. To form the unordered set we randomly form pairs for each sample in the neutral set of SAVEE. We then fill up the set with pairs from IEMOCAP (speaker independent selection) to reach 150 pairs. We perform the same with the respective emotion to obtain an unoredered set with 300 pairs. For the ordered set we randomly select a neutral SAVEE sample for each emotional SAVEE sample and again use IEMOCAP to fill up to 300 pairs. With the learned ranking function we compute phoneme-level rankings for all SAVEE samples.

5.3.2 Emotional TTS

Our goal is to train an emotional TTS system with emotion intensity input on the SAVEE database. Due to the small size of SAVEE we cannot train a modern encoder-decoder network on it, as it quickly overfits before adapting the new speaking styles. Instead we rely on a classical RNN-based network (Zen et al., 2013), which has also been used in recent studies on emotional speech synthesis (Lorenzo-Trueba et al., 2018; Henter et al., 2018). We use oracle durations in all our experiments, because duration prediction for emotional speech is a challenging problem on its own. The model consists of two fully-connected layers with ReLU activation and 1024 neurons, three BiLSTM layers with 512 neurons, and the final 97 dimensional output layer. 5% dropout is applied in all but the final layer. A 128-dimensional speaker and 64-dimensional emotion embedding is concatenated to the input of each layer.

Additionally, we concatenate the mean-variance normalised emotion intensity input in all layers, which gives better results than concatenating it only to the input. For all neutral samples we set the emotion intensity to zero for the entire sequence, indicating that there is no emotion present. At inference time we use the oracle emotion intensity extracted from the reference. We do not predict the emotion intensity internally, because we want to keep it as a tunable input. The inputs to the model are 425-dim context-embeddings (Section 2.2). The model predicts WORLD vocoder features with deltas and double deltas. The output is smoothed with the MLPG algorithm. The WORLD vocoder is used to generate the waveform.

Even for our model the SAVEE database is too small to train a TTS system, so we instead pre-train on the WSJCAM0 database (Section 2.5.3). The model is pre-trained for 35 epochs with a batch size of 16 and a learning rate of 0.001 and early stopping. We reduce the learning rate by a factor of 0.1 on validation loss plateaus. The adaptation to SAVEE is split into adaptation to the neutral subset of SAVEE first, and the entire database second. Each step is further divided into three phases. In the first phase only the speaker embedding is trained (10 epochs, lr=0.001), in the second phase the whole model is trained (10 epochs, lr=0.001), the last phase applies fine-tuning by repeating phase two with a smaller learning rate (10 epochs, lr=0.0001). The batch size in all phases is 16. In each phase early stopping is used and the best model is selected to continue with the next phase. This procedure is the same as in Section 4.6.3.

5.3.3 Subjective Results

In the subjective listening test we investigate how the TTS models compare in terms of perceived emotion and whether the audio quality is impacted. For the test we include five systems:

- **baseline**: TTS model without emotion intensity input
- **attention**: Attention weights extracted from the attention LSTM model
- **transformer**: Smoothgrad saliency map with mean aggregation and smoothing extracted with the transformer model
- **rank**: Phoneme-level rankings extracted with the competitive technique by Lei et al. (2020)
- **ref**: Copy synthesis of the recordings

The test set consists of the same two utterances recorded for every emotion (7, including neutral, excluding disgusted) and every speaker (4 males), i.e., the content is not emotion dependent. This makes 56 samples for each system. As we do not yet have a method to predict emotion intensity from text, we use the emotion intensity extracted from the reference audio by the respective technique. This gives an upper bound on the quality achievable with an emotion intensity input assuming that the prediction is perfect. We find that the emotion

intensity input does not increase the expressiveness of the speech much. However, it offers an unprecedented control to tune the emotion intensity. Informal listening shows a greatly increased expressiveness, while still sounding natural, when scaling the input with a factor of **7**. This factor is set ad-hoc; we currently do not have means to set it automatically. As we (as a research community) lack objective measures for the perceived emotion intensity and audio quality a systematic selection of the factor would require a grid search with subjective listening tests. However, the models have learned to connect certain speech properties with the intensity input, which allows scaling them in a natural way. In general higher intensities result in higher energy in the speech, which is desirable for all but the sad emotion. Thus all our tests use the scaled version except sadness. The emotion intensity for neutral is zero everywhere, so scaling has no effect.

36 listeners participated in the test. Each one rated 25 randomly selected samples in a 5-scale MOS test with 0.5 steps and also selected the emotion they perceived. Table 5.3 summarises the results. The *total* column includes the correct ratings on neutral. As many subtle emotions like fearful or surprised are rated as neutral, this number is biased. The *emotion* column indicates the accuracy on the emotional samples only. On both metrics the attention weight extracted with the attention LSTM model outperforms the other systems. It shows that an emotion intensity input increases the expressiveness of the speech, which is also perceivable by listeners. The *happy* emotion is almost never perceived. The low recognition rate of the reference samples indicates that it was not acted well enough. Providing a neutral reference during the listening test might facilitate its prediction.

Table 5.3 – Results of the subjective evaluation of perceived emotions in percentage. 'total' includes the neutral samples. Accuracy for each emotion is shown as well.

System	total	emotion	neutral	angry	fearful	happy	sad	surprised
baseline	25.3	17.6	**72**	28	15	3	33	12
attention	**35.5**	**28.9**	70	**54**	13	**6**	**55**	**21**
transformer	26.7	20.6	60	33	**25**	0	41	8
rank	25.3	19.0	69	30	14	**6**	40	8
ref	45.9	40.3	75	74	23	19	31	54

It also shows that the quality of the emotion intensity matters as the phoneme level rankings perform much worse, this might be due to the phoneme-level granularity. The key benefit of the ranking function is that it requires very little training data. It might perform best when we do not include any IEMOCAP data. It outperforms the baseline system in a similar manner to that reported in the related work (Lei et al., 2020).

Interestingly, the saliency map extracted from the transformer model performs worse then the simple attention weight, even though the model is much more complex and achieves higher emotion recognition scores. All the saliency map techniques are developed for the field of vision, focusing on convolutional layers. A different type of saliency map might be necessary

for speech tasks or more convolutional networks might allow better saliency maps. The benefit of the transformer model is that it will likely improve its emotion recognition performance with more training data compared to the attention LSTM model due to its small complexity. However, as long as no proper saliency map technique exists, we are limited to models that allow straight-forward extraction of emotion intensity.

Figure 5.4 shows the results of the MOS test. None of the differences in the results are statistically significant in a two-tailed paired t-test with a p-value < 0.05. This includes the copy synthesis reference, which has other quality issues that were rated low by listeners. We have no reason to believe that the proposed techniques deteriorate the audio quality.[4]

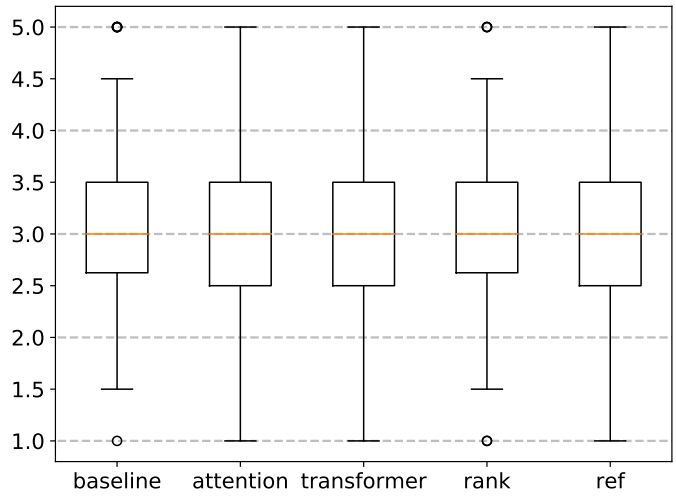


Figure 5.4 – Results of the 5-scale MOS test with 0.5 steps.

5.4 Conclusion and Future Work

In this chapter we present two techniques to extract an emotion intensity input from audio in an unsupervised way by utilising pre-trained emotion recognition networks. We do not require any emotion intensity labeling, but only emotion class labels to train the emotion recognisers. Thus one could also refer to it as weak supervision. From an emotion recognition network with a single attention layer we extract the attention weights as emotion intensity. From a transformer-based network we extract it using saliency map techniques. We show that the additional emotion intensity input improves an emotional TTS system; increasing the accuracy of which human listeners perceive the target emotion without degradation in signal quality. The simpler first method outperforms all others, including a recently published competitive method for emotion intensity extraction based on *relative attributes*.

For the tests we use oracle emotion intensity extracted from the reference audio, which gives an upper bound on the predicted quality. This way we try to break the chicken-egg

[4] Audio samples are available at https://www.idiap.ch/paper/ssw11_emotion_intensity/

problem assuming it will be possible to predict intensity in the future. As the results show great improvements with an emotion intensity input, it is reasonable that research on emotion intensity prediction is a promising future direction. Future work should focus on predicting emotion intensity from text or conversion in speech-to-speech translation. In both cases much more training data will be required, however, it is likely that emotion intensity is localised similarly across speakers, i.e. is speaker independent. Thus it should be possible to train on a large variety of speakers. It is also not necessary to use high quality speech as required for TTS, instead only an emotional class label per utterance is required. This allows the use of many speech emotion recognition databases.

6 Emotional Voice Conversion

In the last chapter we have improved the quality of the emotional training data to improve the generalisability of the model which has led to higher emotion intensity. In this chapter we instead investigate increasing the quantity of the emotional data. Creating voices in more expressive speaking styles usually requires recording large amounts of speech for the desired style. This is very time-consuming and costly. An alternative is the generation of synthetic data to satisfy the high data needs.

The conversion of speech is generally assumed to be easier than TTS, thus has lower data needs. Emotional Voice Conversion (EVC) is a subfield of VC which studies the transformation of a source audio signal into a different emotion while maintaining its linguistic content and speaker identity. Techniques applied in EVC are similar to VC and differ mostly in their feature selection (Kameoka et al., 2018; Rizos et al., 2020). EVC techniques working without hand-crafted features are applicable to other speaking styles as well. EVC is also applied to other tasks like film dubbing.

In this chapter, we aim to convert neutral to emotional speech in German. As we have only a limited amount of emotional German data available, we exploit emotional recordings in US English. We propose EmoCat, a language-agnostic EVC model trained jointly on German and US English working directly on mel-spectrograms. Compared to other works we use mel-spectrograms to leverage Amazon's high-quality universal vocoder (Lorenzo-Trueba et al., 2019) to keep a high bar on segmental quality. Our model adapts the CopyCat model (Karlapati et al., 2020b) (which is based on AutoVC (Qian et al., 2019)) for intra-speaker emotion conversion. CopyCat is a VC model which allows to convert the speech of unseen speakers to a set of target speakers. In contrast to the global speaker identity, emotion is a continuous component of speech. We use adversarial training to explicitly remove emotion leakage from the encoder, which encodes the neutral source spectrogram, to the decoder, which generates the converted emotional spectrogram. We propose a novel improvement to gradient reversal (Ganin and Lempitsky, 2015) to stabilise its gradients. We further investigate fine-tuning to improve naturalness. In an ablation study, we assess the effectiveness of each of the techniques. The proposed model is able to convert neutral German to two different emotions in three

intensities with the support of less than 45 minutes of German emotional data. To the best of our knowledge, no work exists on EVC with multi-lingual data or mel-spectrograms.

This work was performed during an internship in the TTS Research Group of Amazon in Cambridge. The majority of this text will appear at:

- Schnell, B., Huybrechts, G., Perz, B., Drugman, T., and Lorenzo-Trueba, J. (2021). EmoCat: Language-agnostic emotional voice conversion. In *Proc. 11th ISCA Speech Synthesis Workshop*

6.1 Background

6.1.1 Voice Conversion

Voice Conversion (VC) describes the process of converting the speech of some source speakers to speech of a set of target speakers, while retaining the content and style. In contrast to TTS, where audio is generated from a low dimensional sequence of phonemes/characters, VC starts with rich representations and fine details and only has to adapt the speech. This fundamental difference suggests that VC is a simpler task than TTS and thus requires less data. In the following we will give an overview of recent VC models. We will finish this list with the AutoVC and CopyCat model, which form the basis of our work. There are two main approaches to VC research. In the case of parallel training data the same utterances are spoken by different speakers. This allows to learn a direct mapping in a supervised way. For high quality speech generation the training data should contain a wide variety of speech, i.e. it needs a large database. When parallel data is required, the large database has to be recorded for every (new) speaker, which makes it impractical for most real world applications. Current research focuses mostly on non-parallel training data. It is still a challenging task because no direct supervision is possible. Following the chronological order we will start with parallel data works.

Abe et al. (1990) propose the non-parametric vector quantisation approach. It creates a codebook of codewords for each speaker, then the codeword closest to the feature vector is determined, and the corresponding codeword of the target codebook is used as converted vector. The method is applied independently on spectral features and quantised pitch and power. The method was later extended to represent the feature vector with a weighted sum of codewords (Shikano et al., 1991).

As a well known parametric technique GMMs were used by Stylianou et al. (1998). It models the source speaker space with a GMM, then a transformation function is learned, which minimises the total quadratic spectral distortion. The utterances of source and target speaker first need to be time aligned by Dynamic Time Warping (Berndt and Clifford, 1994) (DTW). This is necessary for most parallel data methods. The whole procedure is repeated with a new DTW alignment computed on the converted and target samples.

Dynamic kernel partial least squares (Helander et al., 2011) is a learned non-linear transformation of the source to the target frames. First k-means is applied to the source speaker samples to select a set of reference vectors. Then it learns a kernel transform between the average of the clusters and the target frames.

As discussed in a previous chapter VTLN-based approaches were also used in VC (Sundermann and Ney, 2003; Sundermann et al., 2003; Eichner et al., 2004). A thorough introduction to VTLN is given in Section 4.2.

Non-parallel VC only recently became an active research area. One of the first works is Hsu et al. (2016) which propose a VAE. The variational part accounts for the variants in speech and leads to a more understandable model and better regularisation. The sampled latent code is concatenated with a one-hot speaker embedding. Because the decoder has access to the speaker identity through the one-hot speaker embedding, a small enough VAE dimension will not be wasted to speaker information and will thus contain a speaker invariant representation of content and style and other latent features of the input speech. The model was later adapted with a GAN loss (Hsu et al., 2017a), where the decoder is treated as the generator. Instead of using an explicit discriminator they use a the Wasserstein distance between reconstructed and source spectrogram. The authors argue that the optimal transport solved to compute the Wasserstein distance suits the task of VC.

Gao et al. (2018) propose a CycleGAN for VC. The model consists of two generators and two discriminators. One generator for the conversion from male to female, and one for the converse. Additionally, one discriminator for female samples and one for male. The idea is to convert a source spectrogram to a target spectrogram and back to its source. This is performed for female-male-female and male-female-male. The respective discriminator then decides whether the spectrogram after each conversion is real or fake.

CycleGAN is limited to one-to-one-mappings and requires more generators and discriminator for different speakers, Kameoka et al. (2018) propose the StarGAN architecture instead. It consists of a single generator which converts a source spectrogram conditioned on a target class (a speaker representation in VC) to the spectogram of that target class. One discriminator tries to identify real and fake samples given the target class. Another domain discriminator predicts the class of the converted spectrogram, which should match the conditioning. The model outperformed the previous CycleGAN model. A newer version StarGAN-VC2 (Kaneko et al., 2019) exists. It conditions the first discriminator also on the source class, which ensures the converted data is close to the real data source- and target-wise. Also conditional instance normalisation (Dumoulin et al., 2017) is used instead of batch norm after each layer, which has been shown to be effective for style transfer in computer vision.

AutoVC (Qian et al., 2019) is an auto-encoder network with an information bottleneck and additional speaker encoder network. The model consists of a content encoder that encodes the content and style of a source sample followed by an information bottleneck. The bottleneck is not only low dimensional, to prevent copying from the encoder to the decoder side, a

temporal bottleneck is achieved by selecting only every N-th frame of the encoder output. To match the number of frames the down-sampled features are then copied N times each. The model also has a speaker encoder which extracts a global speaker encoding from a reference audio. During training the reference is another recording from the same speaker. This training procedure encourages the model to learn a content invariant speaker representation. The speaker encoder additionally allows to convert to unseen speakers, in contrast to learned speaker embeddings. The decoder generates the converted speech from the upsampled temporal bottleneck features and the global speaker encoding. A CNN postnet is used to restore fine details, similar to Tacotron. The bottleneck dimension as well as the bottleneck down-sampling frequency has to be chosen so that no speaker information leaks through the bottleneck, but as much content and style information as possible is preserved to facilitate the speech generation. The authors state that there is a balance between reconstruction quality and speaker disentanglement. The model outperformed state-of-the-art models of that time and also enabled zero-shot conversion (conversion to unseen speakers). The authors have extended their work (Qian et al., 2020) by splitting the encoder into three components responsible for the encoding of rhythm, content, and pitch respectively. This allowed disentangled control over each of the three aspects.

CopyCat (Karlapati et al. (2020b), Figure 6.1) is based on AutoVC. It extends the model with a phoneme encoder and instance norm is used instead of batch norm in the encoder. The phonemes need to be aligned and upsampled. The phoneme encodings serve as an additional conditioning on the encoder and decoder side. The model also makes use of speaker embeddings coming from a pre-trained speaker classifier (trained on 1000 LibriSpeech speakers). This allows conversion from unseen speakers to a set of seen speakers. The authors also include a GAN-like loss with a separate discriminator network to increase the audio quality.

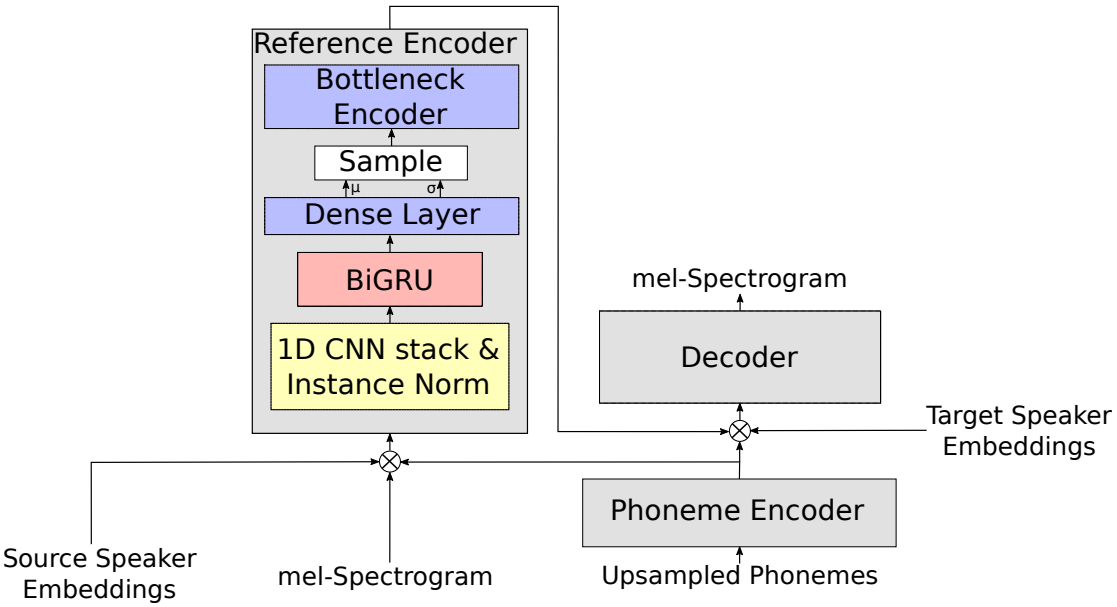


Figure 6.1 – CopyCat network architecture.

Some VC methods also operate directly on the waveform, e.g. Niwa et al. (2018) for parallel training data and Serrà et al. (2019) for non-parallel data. The latter is based on normalising flows, a novelty in VC literature.

6.1.2 Emotional Voice Conversion

Just like VC methods Emotional Voice Conversion (EVC) methods are split into two categories: parallel and non-parallel training data.

In the **parallel data** scenario the database contains the same utterance spoken by the same speaker in the different target emotions. This allows the network to directly learn the conversion. However, these databases are rare and typically small. It is expensive to record all the emotions of an utterance for a large phoneme coverage. Additionally, it is challenging for voice talents to act the target emotion when the linguistic content of the utterance does not match. This can lead to errors in emotion intensity and thus either lowers the quality of the database or requires the exclusion of some recordings. While parallel data methods are less relevant for this chapter, we nevertheless list some recent publications.

An early method (Aihara et al., 2012) uses a GMM for the spectrum and one for F_0. F_0 is approximated with the first N normalised Discrete Cosine Transform coefficients. The GMM models the joint source and target speaker space and its parameters are trained with the Expectation-Maximisation algorithm. Once trained the maximum Likelihood estimation is used to obtain the mixture component sequence for the source. The sequence is needed for the conversion from neutral to three emotions (each emotion has its own GMM).

In Inanoglu and Young (2007) F_0 is modelled by context-sensitive syllable HMMs, duration is modelled by phone-based relative decision trees conditioned on neutral duration and phonetic context, and segmental level spectral conversion is performed by codebook selection or GMMs. Subjective tests showed that the latter is significantly superior.

A BiLSTM-based model is used by Ming et al. (2016) to simultaneously convert spectrum, energy, and F_0. Energy is represented with a 10-scale and F_0 with a 5-scale continuous wavelet transform. For initialisation the model is trained as an average model on a large non-emotional parallel database.

Robinson et al. (2019) model F_0 and duration with an encoder-decoder model with attention. Given the neutral quantised F_0 contour and the syllable position information the model predicts emotional F_0 for all vowels. Thanks to the attention mechanism the model is also able to adapt the duration. The spectral features of the neutral speech are time-stretched to match the converted F_0.

Shankar et al. (2019) use a highway network to map spectrogram, F_0, and a gender embedding to pitch and energy of the target emotion. While the spectrogram is used to condition the network, it itself is not changed. Pitch and energy are modelled in separate networks. A smaller network for gender classification is used to predict the gender embedding.

In the **non-parallel data** scenario, the utterances for each emotion differ, meaning that the content can better match the emotion. This allows a wider variety of utterances and also simplifies acting for the voice talents. With the lack of parallel data, a model cannot be trained to do the conversion directly as the ground truth target is not available. The training can only be guided in an unsupervised way. GANs and cycle consistency losses are commonly used techniques.

In Gao et al. (2019) an encoder-decoder structure with a content and style encoder, which both generate a global code, is used to convert MCEP extracted by WORLD. The model is trained with three losses. First, the cepstrum is auto-encoded and an L1 reconstruction loss applied. Second, the model converts the source spectrogram to the target emotion and passes the generated spectrogram to the two encoders again. Semi-cycle consistency L1 losses force the generated global codes to match before and after conversion. Third, a GAN loss tries to discriminate generated from recorded samples. The decoder uses adaptive instance normalisation (Ulyanov et al., 2016) to add the target emotion. F_0 is converted by a linear transform (log Gaussian normalisation) to match the statistics of the target emotion domain. The band aperiodicities remain unchanged.

StarGAN is used in Rizos et al. (2020) on WORLD features with a reconstruction loss, an L1 cycle consistency loss, and a real/fake GAN loss. The model architecture is the same as in the original StarGAN-VC paper (Kameoka et al., 2018). An emotion recognition model (a variant of Zhang et al. (2019c)) was trained with the generated samples and evaluations show that its accuracy improved.

CycleGAN has also been used for emotion conversion (Zhou et al., 2020a). It is trained with three losses: 1) a reconstruction loss, 2) a cycle-consistency loss on a sample converted to another emotion and then back to the source emotion, and 3) the GAN loss for real/fake discrimination. The experiments show that separate CycleGANs for F_0 and MCEP outperform a joint model. Liu et al. (2020) follows a very similar approach but uses an additional emotion classification loss and no reconstruction loss.

A different approach is the variational auto-encoding Wasserstein GAN (VAW-GAN) for emotion conversion (Zhou et al., 2020b) (originally proposed for VC in Hsu et al. (2017a)). It consists of a VAE structure where the decoder is conditioned on an emotion embedding. The latent dimension is chosen to be small enough so that it will not contain emotion information. The model is trained with reconstruction loss, standard KL divergence on the VAE latent space, and a Wasserstein GAN loss.

Our approach is closest to the VAW-GAN by Zhou et al. (2020b) as it employs a similar encoder-decoder structure with a VAE encoder. However, the bottleneck used is temporal and drastically smaller, also we condition the decoder on the linguistic content. Our reference encoder is similar to the one in Skerry-Ryan et al. (2018). In contrast to all related work above, we operate on mel-spectrograms and train with multi-lingual data.

6.2 Model description

In this section we introduce EmoCat, a language-agnostic intra-speaker emotion conversion model. It aims to convert neutral speech to emotional speech of the same speaker[1]. EmoCat is based on CopyCat (Karlapati et al. (2020b), see Section 6.1.1) and inherits the same structure and hyper-parameters. We repeat the model structure here:

- The phoneme encoder is the same as in Tacotron 2 (Shen et al., 2018) consisting of three convolutional layers with 512 filters of shape 5×1 with batch normalisation and ReLU non-linearity, followed by a BiLSTM with 256 neurons per direction. The input phoneme embeddings are 512-dimensional.

- The reference encoder has three convolutional layers with instance normalisation (Dumoulin et al., 2017) followed by a BiLSTM. Instance normalisation works as batch normalisation, but applies the normalisation channel-wise independently on each element in the batch. The idea is that the constant part for each channel of each element in the batch is the speaker identity; normalising against it removes it. The last BiLSTM state is projected to form the parameters of a VAE. We sample from the VAE and pass it through a dimensional and temporal bottleneck, by selecting only every N-th frame. The down-sampled frames are then copied N times to match the former temporal resolution. This prevents the model to pass fine-grained information through the bottleneck and reduces leakage.

- The parallel decoder has a stack of three convolutional layers and a BiLSTM. It is conditioned on the bottleneck features, a target class embedding (speaker embedding for VC and emotion embedding for EVC), and the phoneme encodings.

- CopyCat also has a self-attention discriminator (Zhang et al., 2019a) used during GAN training. We omit it here because it is not used in EmoCat.

EmoCat differs from CopyCat in four aspects:

1. It uses 64-dim emotion embeddings instead of 128-dim speaker embeddings (see Section 6.2.1).

2. It uses a gradient inverter block to remove emotion leakage from the bottleneck embeddings (see Section 6.2.2).

3. It operates on multi-lingual data (see Section 6.3.1).

4. It does not pass the phoneme embeddings to the VAE reference encoder.

[1] We have informally verified that it also allows conversion between emotions.

Figure 6.2 shows the network structure. The VAE reference encoder encodes the mel-spectrograms and its emotion embedding. A dimensional and temporal bottleneck is applied by only selecting every N-th frame (Qian et al., 2019). Each selected frame is copied N times (to restore the sequence length). The bottleneck embeddings should contain as much information as possible to generate high-quality speech but no emotion information. This is ensured by passing them through the gradient inverter to the emotion classifier, which removes any leaking emotion information. Force-aligned upsampled phonemes are encoded by the phoneme encoder to produce phoneme embeddings. During inference, the bottleneck and phoneme embeddings are stacked with the target-emotion embedding centroid and consumed by the parallel decoder to produce the converted mel-spectrogram. During training, the oracle utterance-level emotion embedding is used on the encoder and decoder side. Source and target spectrograms are the same as well. The parallel decoder consists of a stack of three convolutional layers followed by a unidirectional LSTM. The model is trained with an L1 reconstruction loss and the KL-loss on the VAE latent space.

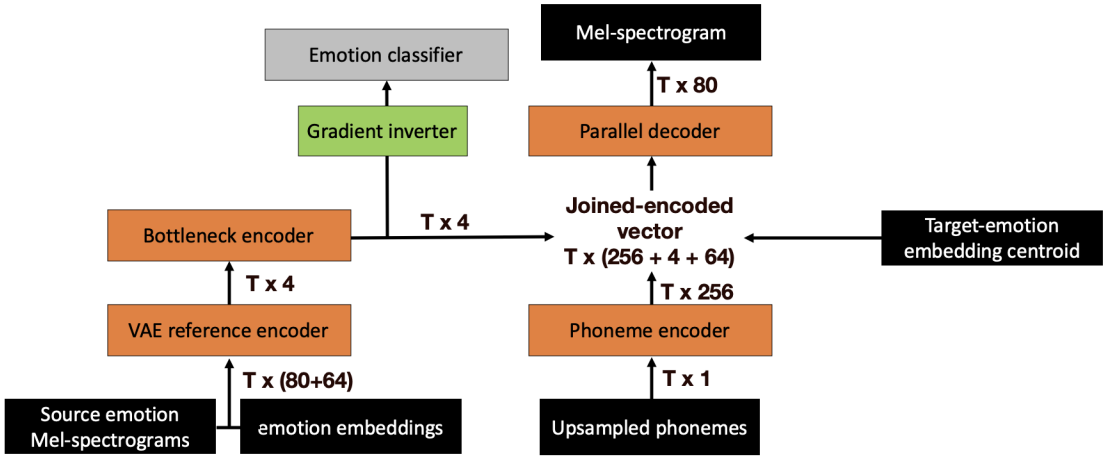


Figure 6.2 – The mel-spectrograms and emotion embedding are stacked, then given to the VAE reference encoder, and finally passed through a dimensional and temporal bottleneck. A gradient inverter followed by an emotion classifier is used to remove leaking emotion information. The centroid of the target emotion is used during inference. During training source and target spectrograms are the same as is the case for the emotion embedding.

6.2.1 Utterance-level emotion embeddings

During training, utterance-level emotion embeddings are fed to the VAE reference encoder and the parallel decoder. The emotion embeddings need to be organised language-independently by their style and other latent information to be beneficial to the model. This excludes simple embeddings per emotion class and suggests a learnable approach.

We obtain the emotion embeddings from a separate TTS model, which is pre-trained to do phoneme to mel-spectrogram conversion. The TTS model has a Tacotron-like architecture (Latorre et al., 2019) with the addition of two VAE reference encoders (Tyagi et al., 2020).

Within the reference encoder the last GRU state is projected to form the VAE parameters. The embedding is obtained by sampling from the VAE. One reference encoder captures the speaker information while the other captures the emotion. We use intercross training (Bian et al., 2019) to guide each encoder to encode only the speaker/emotion information and to be language-independent. We use the predicted embeddings from the emotion reference encoder as utterance-level emotion embeddings for the EmoCat training. We could learn the emotion embeddings in a similar fashion on-the-fly within the EmoCat model, but this would increase its training time, which is not desirable during research. We could also obtain them from a simple emotion recognition model, but we hypothesised that those embeddings might be more suited for recognition than generation, because they were trained with a loss based on recognition performance.

For the CopyCat model, robust speaker embeddings from a pre-trained speaker identification system are necessary, because the model also has to convert from unseen speakers. This is not the case for the EmoCat model, which only converts between seen emotions. Thus it requires less sophisticated emotion embeddings.

During inference, the utterance-level emotion embedding of the converted spectrograms is unknown. Instead we compute the centroid for each emotion over all emotion embeddings extracted from the training set and feed it to the decoder. The VAE reference encoder still uses the utterance-level emotion embedding of the input audio.

6.2.2 Gradient inverter

As emotion is a continuous and integral part of speech, it is necessary to explicitly prevent it from leaking from the encoder to the decoder side. With a pre-trained EmoCat with frozen weights we trained independent GRU emotion classifiers to predict the source emotion from the bottleneck embeddings, where the best achieved 64% overall accuracy. We found that heavy leakage resulted in low emotion intensity during conversion. Decreasing the bottleneck (as described by (Qian et al., 2019) in their AutoVC paper) led to heavy degradation in signal quality and intelligibility. With the reconstruction loss alone, we could not force the bottleneck embeddings to remove the undesired emotion information while keeping information needed for high signal quality.

Instead we used a gradient reversal block before the emotion classifier during training to actively remove emotion leakage from the bottleneck embeddings. The idea of gradient reversal is to reverse the gradients during back-propagation to remove any activation in the input that helps the following classifier. Gradient reversal achieves this by swapping the sign of the gradient Δ (Equation 6.1). It also applies a weight λ to control the impact of the gradient on the preceding layers. The weight greatly influences the performance of the final model.

$$\Delta' = -\lambda\Delta \tag{6.1}$$

We experimented with a feed-forward and a GRU based emotion classifier. Interestingly, EmoCat converged to a better model in terms of conversion ability with the feed-forward classifier than the GRU one. This suggests that with gradient reversal even a weak classifier gives sufficient gradients to lead to a better convergence point.

We again trained the same emotion classifier as above on the bottleneck embeddings of the model with gradient reversal. The classifier mainly predicted the majority class (95% of the time) showing that the majority of the emotion leakage was removed. Informal listening verified that the conversion ability of the model improved.

We argue that a simple swap of the sign (Equation 6.1) fulfils only half of the reversal purpose. Consider the following two scenarios:

1. Imagine there is **no leakage** in the input. As the classifier cannot rely on any information in the input, its prediction is random and the cross-entropy loss on its predictions is high. Thus the back-propagated gradients are large as well. Even though there is no leakage the preceding network receives a large reversed gradient.
2. Imagine there is **significant leakage** in the input and the classifier is already properly trained. Then its prediction is good, the cross-entropy loss is low, and the back-propagated gradients are small. Even though there is significant leakage the preceding network receives only a small reversed gradient.

The desired effect on the preceding network in both scenarios should be swapped. Without any leakage the received gradients should be small, while with significant leakage the gradients should be large. To address this issue, we present the **gradient inverter** block. Instead of only swapping the sign of the gradient, it performs a proper inversion by also converting small gradients to large ones and vice versa. We have experimented with two gradient inverter functions.

$$\Delta' = \frac{-\lambda\Delta}{||\Delta||_2^2} \qquad \text{Inverse square norm} \tag{6.2}$$

$$\Delta' = \frac{-\lambda\Delta}{\exp||\Delta||_2^2} \qquad \text{Inverse exp square norm} \tag{6.3}$$

Equation 6.2 implements directly what we want to achieve by scaling the gradient by its squared norm. Gradients with a norm smaller than one will become greater than one and vice versa. However, it might lead to unstable behaviour as gradients with a norm close to zero are scaled towards infinity. Equation 6.3 prevents this by bounding the denominator to less than one. In this variant, gradients with a small norm remain almost unchanged while big gradients are quickly faded out. We found that depending on the target emotion one of the proposed inverter functions performs better.

6.2.3 Fine-tuning

While the EmoCat model with the proposed gradient inverter achieved high emotion intensities, its signal quality left room for improvement. We investigated fine-tuning on a subset of the training data. First the model was trained with all data until convergence. Then we continued training on emotional and similar amounts of neutral data. This should compensate the averaging effect in the decoder introduced by the huge amount of neutral training data. We did not change any hyper-parameters, learning rates, or losses compared to the first training step. This approach outperforms a GAN-like loss (same as used for CopyCat), which strives for the generated spectrogram to be indistinguishable from the recordings.

6.3 Experiments

We aim at generating emotional German samples by converting from neutral using a model trained with a limited amount of emotional German data. We focused on two emotions: excited and disappointed, in three intensities: low, medium, high.

6.3.1 Database

We use two Amazon internal databases. For German, we use more than 20 h of neutral and 45 min of emotional single-speaker recordings of a female voice. 20 neutral samples are set aside as test set. We do not use a development set to guide the training because the L1 reconstruction loss does not match human perception. The 45 min of emotional data are split equally into excited and disappointed. 25% is low, 50% medium, and 25% high intensity. Excluding the test set, we have around 5 min for the most challenging intensity: high. As we do not have access to more emotional German data, we use recordings of a female US English voice as supporting speaker. From this speaker, we use more than 20 h of neutral and more than 10 h of emotional recordings of the same emotion categories. We found that including US English data greatly improved the conversion abilities of our model, despite the differences in language. This suggests that the production of emotion follows a similar behaviour in English and German, which thus makes it beneficial to include the English data during training. This might hold true for other emotions as well. 24 kHz recordings are used. We trim all silences to be maximum 100 ms and extract 80-dim mel-spectrogram. We use phonemes with fully disjoint sets for English and German, thus the speaker identity can directly be inferred and explicit speaker embeddings are unnecessary.

6.3.2 Models

We conduct an ablation study across three models. Each is trained for 100k steps on the combined two databases. The mel-spectrogram is synthesised with Amazon's universal vocoder (Lorenzo-Trueba et al., 2019).

1. **Grad. reversal** - This model uses the vanilla gradient reversal block (Equation 6.1) to remove leaking emotion information. In contrast to the following two models, we used a weighted cross entropy loss for the adversarial emotion classifier to compensate for the huge class imbalance in the training data. We chose the weights inverse proportional to the amount of the emotion in the total training data. We found that this improved the grad. reversal model.

2. **Grad. inverter** - This model replaces the gradient reversal block of model 1 with the improved gradient inverter block (see Section 6.2.2). We use two separate models for the conversion. The model to convert to the three excited emotions uses the inverse exp square norm function (Equation 6.3), while the one to convert to disappointed uses inverse square norm (Equation 6.2). This was selected based on a clear performance difference in informal listening.

3. **Fine-tuning** - This is model 2 fine-tuned for 2k steps as described in Section 6.2.3. The best results were obtained by fine-tuning on the emotional data of the target speaker with a similar amount of neutral data as for each emotion. The neutral data requirement is probably due to the adversarial training. This simple fine-tuning outperforms GAN fine-tuning.

We wanted to include a state-of-the-art baseline, however, we did not find any work on emotion conversion from spectrograms, which is required to use our high-quality neural vocoder. We adapted the work of Rizos et al. (2020) based on their StarGAN implementation[2] to use mel-spectrograms instead of WORLD vocoder features, but the quality of the synthesised speech was very low. It is likely that major adaptations to the model architecture are necessary to achieve competitive results. However, creating such a baseline system is out of scope for this work. A comparison with a WORLD vocoder-based model is superfluous (Wang et al., 2018a), therefore it was impossible for us to include a competitive state-of-the-art baseline model in our benchmark.

6.3.3 Evaluations

We randomly selected 10 neutral German samples from the held-out test set and converted them to each of the six emotion intensities. 24 native German listeners rated the samples in terms of emotion intensity and audio quality in a MUSHRA test from 0 to 100.

Emotion intensity

We asked listeners to rate the emotion intensity where we provided another neutral recording (different sentence) as a reference of 0. We also included another recoding of the same emotion of the target speaker as an upper anchor and the utterance generated by a neutral baseline

[2] https://github.com/glam-imperial/EmotionalConversionStarGAN

system. We see in Figure 6.3 that our gradient inverter model significantly worse for low excited, outperforms vanilla gradient reversal for medium excited, and is similar in high excited. With an intuitively large amount of ratings the *p-value* of a two-tailed t-test is still > 0.05 in the high excited case, which suggests that there is no statistical difference (denoted as a horizontal bar in the plots). The exp square norm function (Equation 6.3) only scales large gradients down which does not seem to be optimal for the excited intensities. For disappointed the gradient inverter model achieves more than 20 MUSHRA points higher score across all intensities, proving the improvement through the gradient inverter function. We either did not yet find a gradient inverter function which generalises to different emotions, or the function should be chosen depending on the use case. Fine-tuning lowers the emotion intensity for the medium and high emotions. This shows an averaging effect of the neutral and low intensity data. It should also be noted that we see a clear ascent from the low to the high intensity, but do not yet reach the emotion intensity of the recordings except for low disappointed. We were only able to partially address the averaging effect in the decoder, which might reveal a general shortcoming of current decoder architectures. Highly expressive data in another language seems to improve the system only to a certain point. More high expressive German recordings, even from other speakers, might push the emotion intensity further.

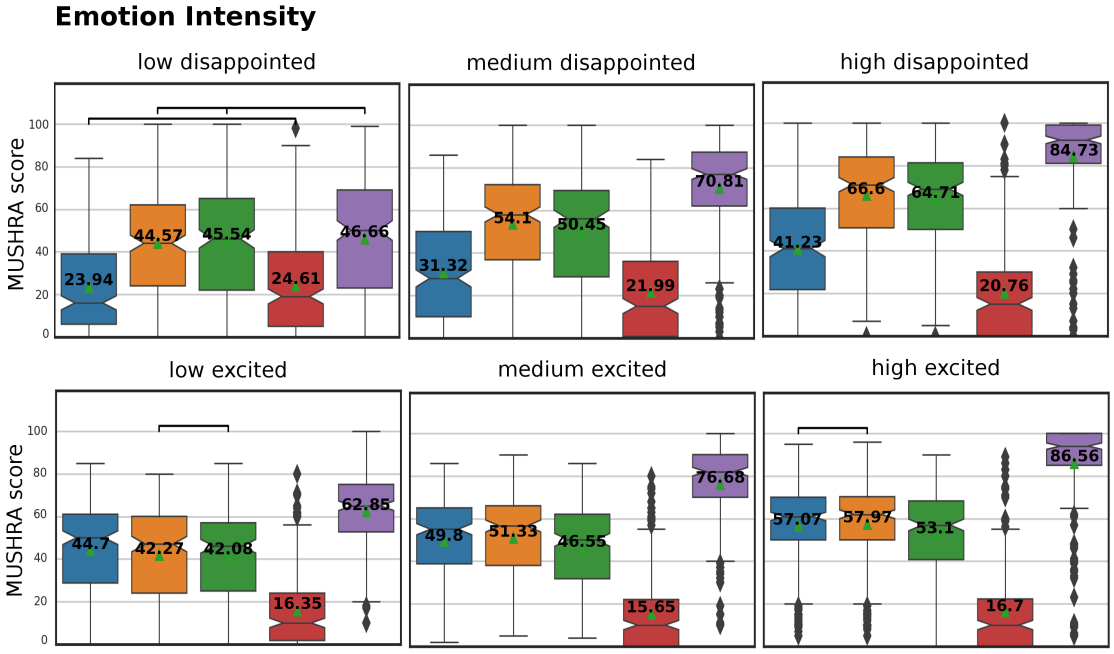


Figure 6.3 – System descriptions: blue: gradient reversal, orange: gradient inverter, green: gradient inverter fine-tuned, red: neutral baseline, purple: recordings. Black horizontal bars connecting systems denote no statistically significant difference between them (*p-value* > 0.05).

Audio quality

We compared the same systems as above but without a reference sample and asked the listeners to rate the audio quality (Figure 6.4). The test instruction was: "Please rate the systems in terms of audio quality. Try to ignore the content of the speech and the expressiveness and instead focus on the quality of the audio signal (e.g. glitches, clicks, noise, ...).". We do not see a statistical difference between all systems for medium and high excited. Vanilla gradient reversal outperforms both other techniques for low excited and all disappointed intensities, but at a lower emotion intensity which makes the comparison unfair. The other techniques are still at par with the recordings. We see a trade-off between emotion intensity and audio quality here. We usually found that higher emotion intensities suffer from reduced signal quality. Most likely because low intensities are close to neutral samples for which we have a lot of training data. This leads back to the averaging effect in the decoder. We suggest to explore different decoder architectures more suitable for highly expressive speaking styles. While we are not able to reach the emotion intensity of the recordings yet, we achieve high audio quality at a generally lower intensity level. Fine-tuning did not achieve the desired improvement in audio quality. Even though it increased the MUSHRA score in five out of six emotions the difference is only statistically significant for low disappointed. The increase in audio quality might be a consequence of the lower emotion intensity instead of fine-tuning. However, for low disappointed fine-tuning increased audio quality without reduced emotion intensity. Interestingly, listeners found the audio quality of the neutral baseline system to be significantly higher than the emotional recordings. Our current hypothesis is that the listeners indeed noticed that the recordings are acted emotions and thus found them slightly unnatural.

6.4 Conclusion

In this chapter we proposed EmoCat, a novel EVC model based on CopyCat, which operates directly on mel-spectrograms. It allows to convert neutral to emotional samples in German with less than 45 minutes of German emotional recordings. It achieves this by leveraging large amounts of emotional English data with the same emotions. While we expect the technique to be language-agnostic, we only demonstrate it for the rather similar languages German and US English. Even though the model is able to generate expressive speech at different intensities, we were not yet able to match the expressiveness of the recordings. Moreover, we presented the gradient inverter block, an improvement to gradient reversal. This showed statistically significant improvements in emotion intensity for four out of six emotions in subjective listening tests. We also found minor improvements in audio quality, at the cost of emotion intensity, through fine-tuning on the target emotional data.
Future work is required to investigate the influence of increasing the amount of emotional German data, testing on more dissimilar languages, and further improvements to the gradient inverter functions. It also remains to test how the synthetic data affects the affective TTS model in the downstream task. Recent work on newscaster style speech (Huybrechts et al., 2020) suggests that synthetic data can indeed improve the affect of a TTS model.

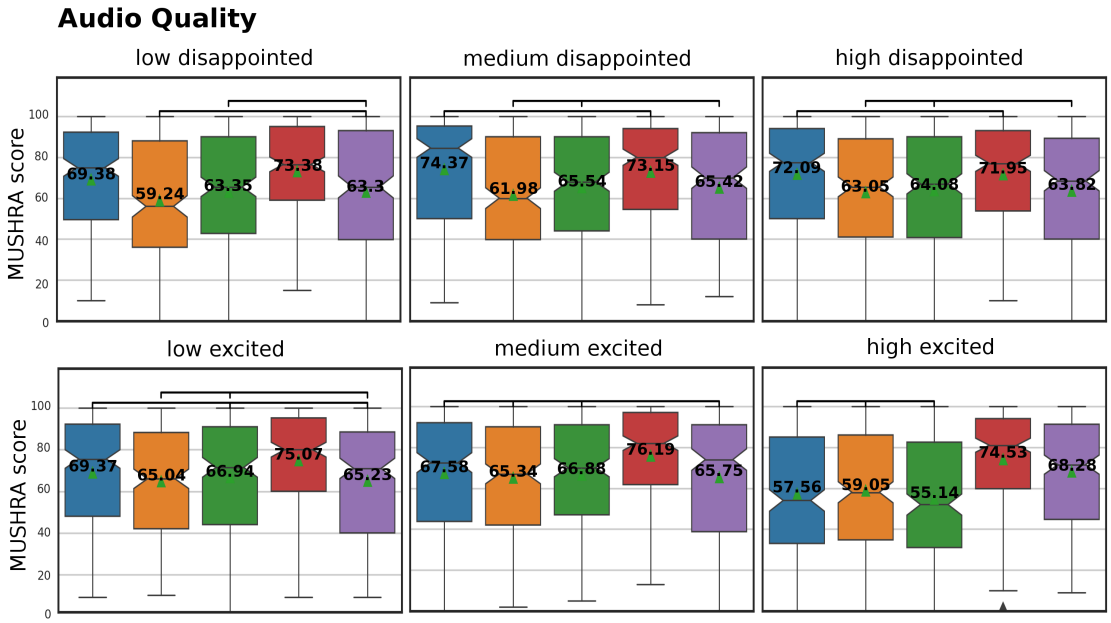


Figure 6.4 – System descriptions as before: blue: grad. reversal, orange: grad. inverter, green: grad. inverter fine-tuned, red: neutral baseline, purple: recordings. Black horizontal bars connecting systems denote no statistically significant difference between them (*p-value* > 0.05).

7 Conclusion

In this thesis we investigated techniques to increase the controllability and interpretability of speech synthesis systems. A special challenge was the integration within the *Deep Learning* framework as neural network components. Today's TTS systems still suffer from low variability in the generated speech and are incapable of generating affective speech. Due to the high variability in affective speech recording large databases is too costly. The techniques developed in this thesis aim to decrease the data needs of affective speech synthesis by incorporating well known signal processing techniques and prior knowledge about speech into the models.

We have developed an RNN which is capable of driving the GCR model to produce natural intonation contours while retaining its controllability and physiological plausibility. We have proposed a training algorithm inspired by the SPAN algorithm from the SNN literature to train it on the spiky command signals of the GCR model. We further extended the model with neural second-order linear all-pole filters to train directly on the LF_0 contour. This in turn allowed the model to learn the set of muscle responses and converge to a better set than a hand-crafted dictionary. We showed that an L1-regularisation term effectively ensures spiky latent representations, which can be interpreted as atom spikes. Retaining the analogy to the GCR model is desirable as previous work has shown that the GCR model is a strong candidate for the transplantation of emphasis in speech-to-speech translation.

Besides our work on pitch contours we developed a neural APW inspired by VTLN to adapt the spectral features. We proposed to split the computation into constant and varying parts to allow an efficient back-propagatable module. The APW is driven by a one-dimensional frame-level parameter, which directly maps to the warping applied on the spectral features, thus allowing direct interpretation and control. Experiments proved that the APW improves TTS systems on multi-speaker tasks like multi-speaker TTS and few- and zero-shot adaptation with limited data. The APW did not prove useful for emotional speech synthesis, even though observations in emotion recognition suggest that many emotions cause a formant shift. We hypothesised that localisation information of the emotion within the utterance would be required to apply the settle emotion dependent warping by the APW.

To add localisation we proposed to extract emotion intensity information from pre-trained emotion recognisers by two techniques in an unsupervised way (only the emotion label is needed). First, we used the attention weights of an emotion recogniser with a single attention network as indicator of the emotion intensity. Second, we investigated various saliency map techniques to extract emotion intensity from a transformer-based emotion recogniser. We showed that the additional emotion intensity input improves an emotional TTS model. We found that the attention weights of the simple attention-based emotion recogniser outperform the saliency maps extracted with the more complex transformer-based model.

As an alternative to explicit emotion intensity extraction we investigated the generation of artificial emotional data. The assumption is that a TTS model can infer the localisation from text with sufficient training data. We proposed a language-agnostic emotional voice conversion model which is able to convert neutral German to emotional German speech by exploiting large amounts of emotional English recordings. We developed an improvement to gradient reversal to truly reverse gradients, which enabled the model to benefit from English data without leakage. The model showed to be able to convert neutral to excited and disappointed emotion at varying intensities, but does not fully match the expressiveness of the recordings.

7.1 Recommendations

In this thesis a toolbox of controllable and interpretable techniques for (affective) speech synthesis was developed. This toolbox is not yet complete and some of the techniques could be tested in more broad scenarios. Completing the toolbox and combining the methods to form a unified framework is a major perspective. Then experiments on dialogue agents and integration in speech-to-speech translation systems are reasonable research directions to test the methods in realworld applications. Most experiments were conducted on English only and it remains to show that they generalise to other languages, especially to those significantly different such as tonal languages.

The controllable adaptation of durations was not investigated in this thesis due to time constraints, but duration plays an important rule in affective speech and deserves attention in the future. Duration prediction is different in the sense that it is very similar across speakers, which in turn allows pooling of emotional data of multiple speakers, and does not require high quality recordings in contrast to TTS, which allows to use emotion recognition databases.

As stated in Chapter 5 the oracle emotion intensity is used for all the experiments. How to obtain the emotion intensity is non-trivial and depends on the task. In speech-to-speech translation the model could also transfer the emotion intensity. As we do not have a speech-to-speech translation system available at Idiap testing this hypothesis would require a significant amount of time. Predicting the emotion intensity value from text is required in TTS systems and might benefit from the same relaxed constraints as duration prediction (speaker-independence and lower audio quality).

The emotion conversion system developed in Chapter 6 was not tested on the downstream task. It remains to show whether the artificial data improves the TTS system. We assume it does, because it has already been shown that artificial data helps TTS system on conversational and newscaster style. As the model was developed as part of an internship at Amazon access to the model and databases could not be granted afterwards and further research was not possible.

A Neural Generalised Command Response Model Results

Here we show a few more figures demonstrating the capability of the neural GCR model introduced in Section 3.2 to predict atom spikes. The caption is not repeated after the first figure.

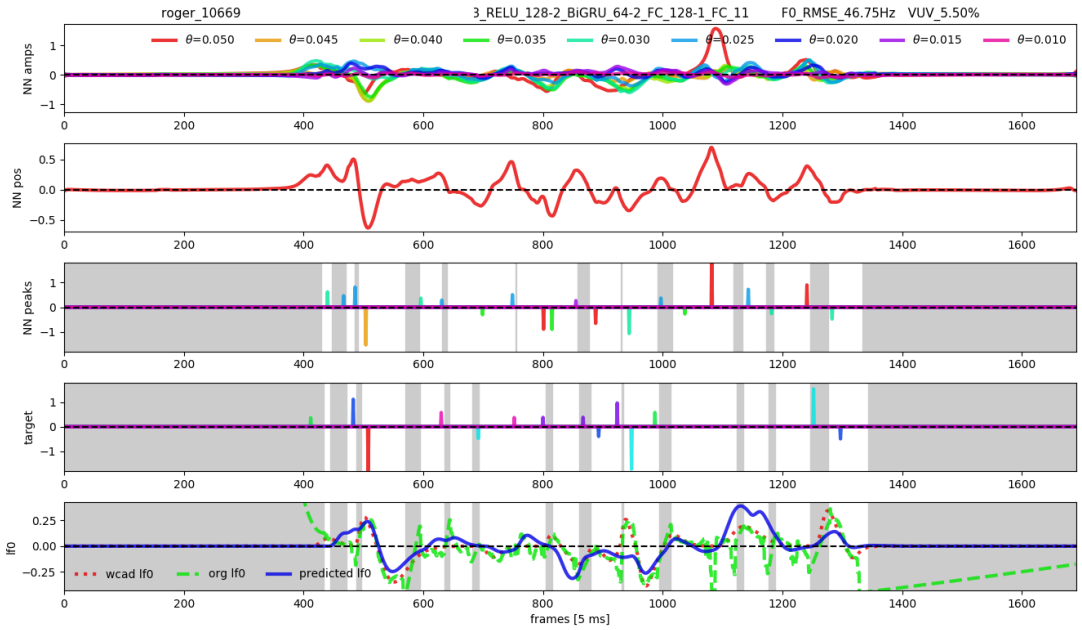
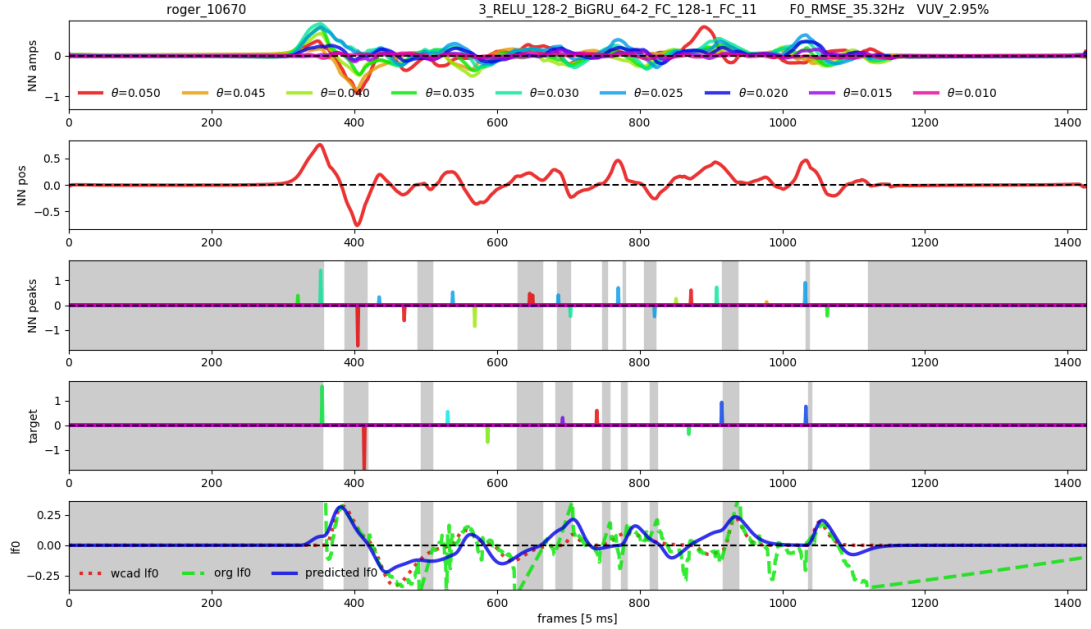
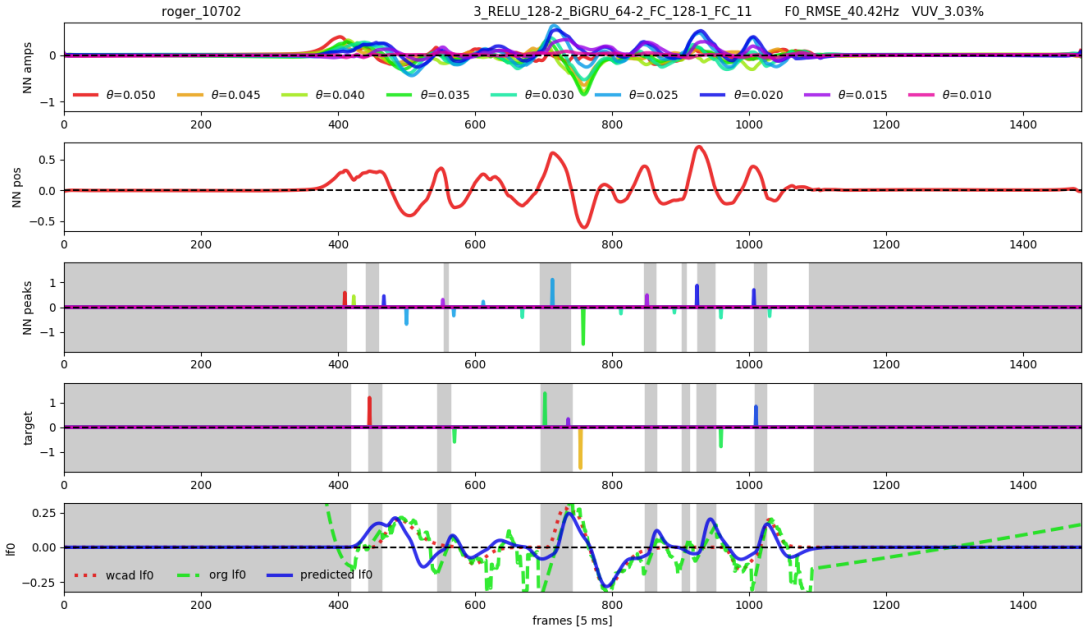


Figure A.1 – Synthetic features on a temporal scale of 5 ms per frame. Plot descriptions from top to bottom: **1:** Nine amplitude outputs (one per θ). **2:** Spike position flag before post-processing. **3:** Atom spikes generated from spike position and amplitude max/min, V/UV flag (unvoiced frames grey). **4:** Target atom spikes and target V/UV flag (unvoiced frames grey). **5:** LF_0 (without phrase component) NN reconstruction (blue, solid), target reconstruction (red, dotted), original (green, dashed) and target V/UV flag (unvoiced frames grey).


roger_10670
3_RELU_128-2_BiGRU_64-2_FC_128-1_FC_11
F0_RMSE_35.32Hz
VUV_2.95%
NN amps
θ=0.050
θ=0.045
θ=0.040
θ=0.035
θ=0.030
θ=0.025
θ=0.020
θ=0.015
θ=0.010
NN pos
NN peaks
target
lf0
wcad lf0
org lf0
predicted lf0
frames [5 ms]


roger_10702
3_RELU_128-2_BiGRU_64-2_FC_128-1_FC_11
F0_RMSE_40.42Hz
VUV_3.03%
NN amps
θ=0.050
θ=0.045
θ=0.040
θ=0.035
θ=0.030
θ=0.025
θ=0.020
θ=0.015
θ=0.010
NN pos
NN peaks
target
lf0
wcad lf0
org lf0
predicted lf0
frames [5 ms]

B Emotion Intensity Saliency Maps

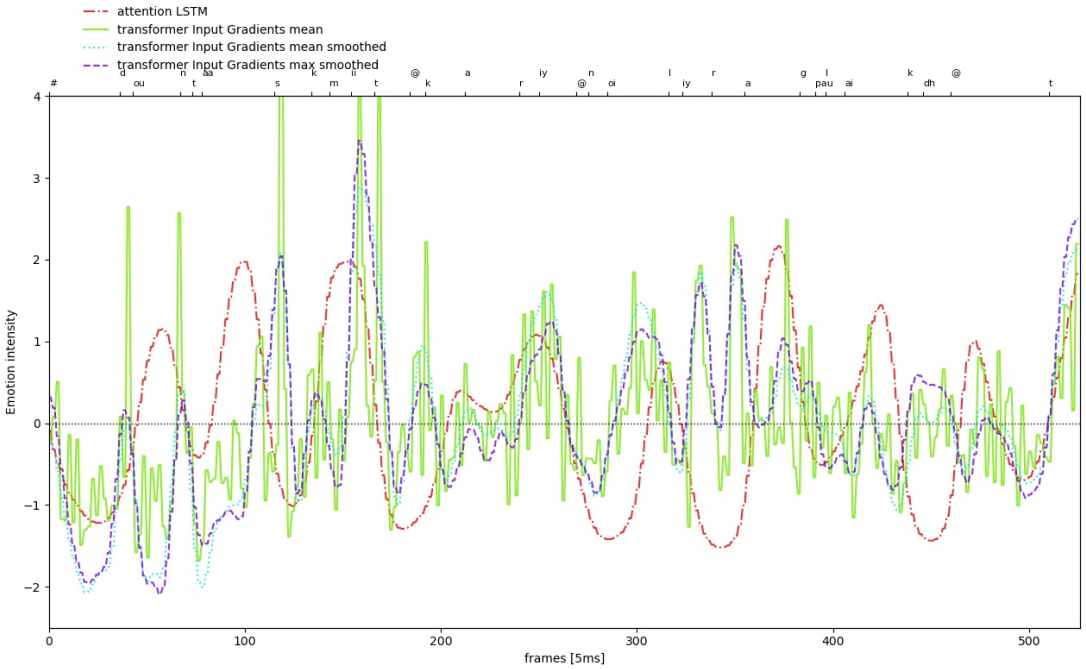


Figure B.1 – Emotion intensities extracted with the attention LSTM model and with the Input Gradients saliency map with smoothed max and mean aggregation as well as oracle mean aggregation. For better comparison each intensity is mean-variance normalised based on its own statistics. The content is: “Don’t ask me to carry an oily rag like that.”

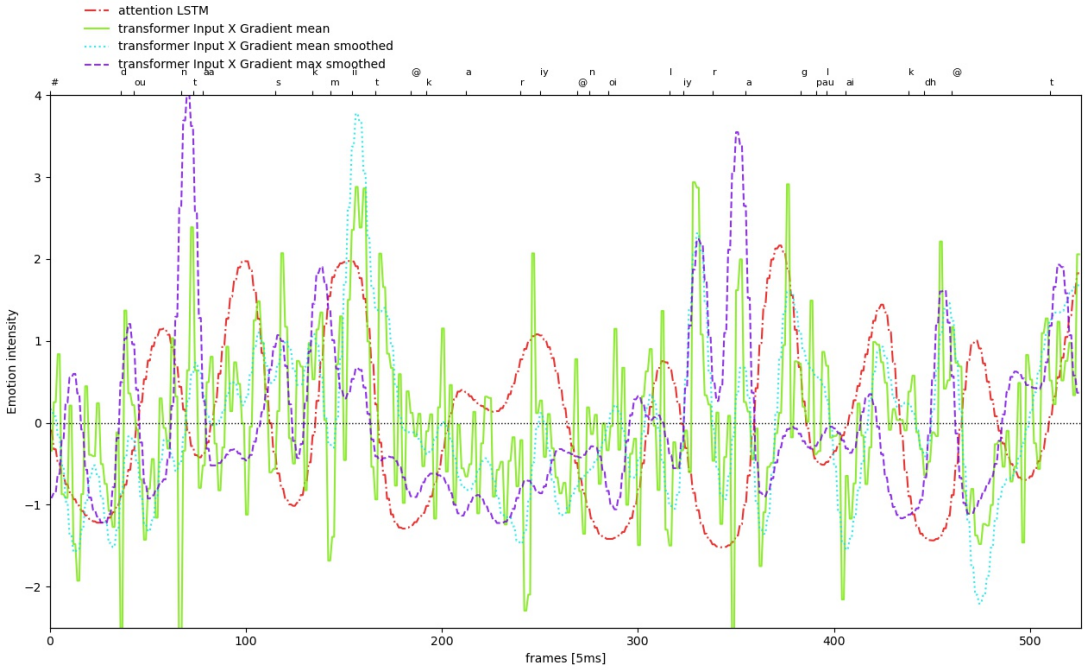


Figure B.2 – Emotion intensities extracted with the attention LSTM model and with the Input x Gradient saliency map with the same aggregations as above for the same utterance as in the figure above.

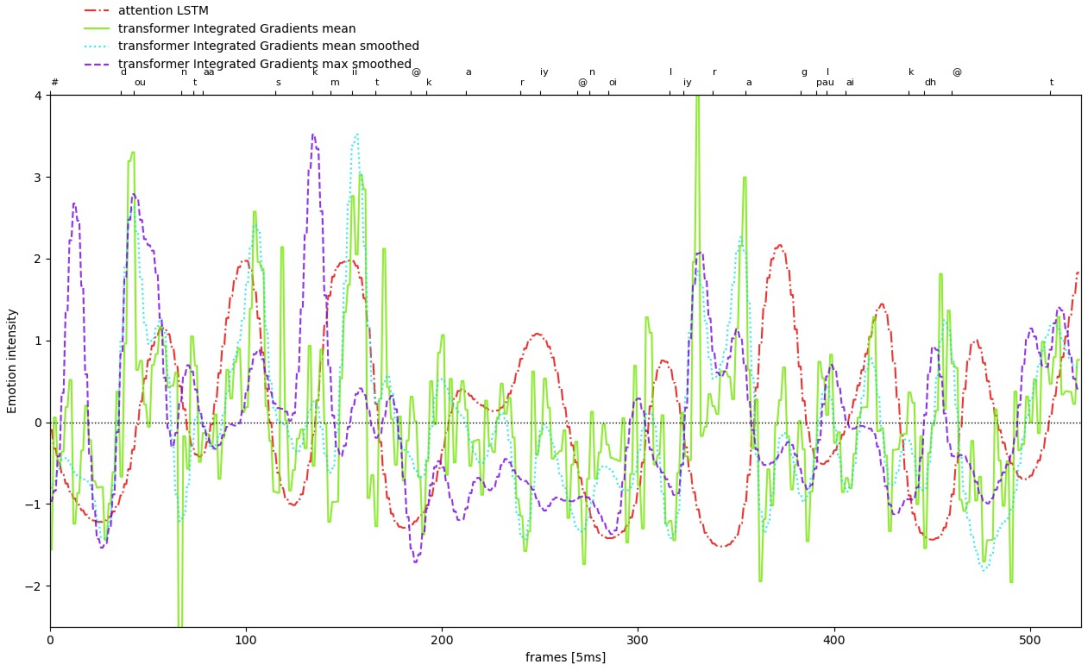


Figure B.3 – Emotion intensities extracted with the attention LSTM model and with the Integrated Gradients saliency map with the same aggregations as above for the same utterance as before.

Bibliography

Abe, M., Nakamura, S., Shikano, K., and Kuwabara, H. (1990). Voice conversion through vector quantization. *Journal of the Acoustical Society of Japan (E)*, 11(2):71–76.

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 *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6179–6183. IEEE.

Agüero, P. D. and Bonafonte, A. (2005). Consistent estimation of Fujisaki's intonation model parameters. In *Proc. SPECOM*. Citeseer.

Agüero, P. D., Wimmer, K., and Bonafonte, A. (2004). Automatic analysis and synthesis of Fujisaki's intonation model for TTS. In *Speech Prosody 2004, International Conference*.

Aihara, R., Takashima, R., Takiguchi, T., and Ariki, Y. (2012). GMM-based emotional voice conversion using spectrum and prosody features. *American Journal of Signal Processing*, 2(5):134–138.

Akuzawa, K., Iwasawa, Y., and Matsuo, Y. (2018). Expressive speech synthesis via modeling expressions with variational autoencoder. *Proc. Interspeech 2018*, pages 3067–3071.

Amodei, D., Ananthanarayanan, S., Anubhai, R., Bai, J., Battenberg, E., Case, C., Casper, J., Catanzaro, B., Cheng, Q., Chen, G., et al. (2016). Deep speech 2: End-to-end speech recognition in english and mandarin. In *International conference on machine learning*, pages 173–182. PMLR.

Arik, S., Chen, J., Peng, K., Ping, W., and Zhou, Y. (2018). Neural voice cloning with a few samples. In *Advances in Neural Information Processing Systems*, pages 10019–10029.

Back, A. D. and Tsoi, A. C. (1991). FIR and IIR synapses, a new neural network architecture for time series modeling. *Neural Computation*, 3(3):375–385.

Bailly, G. and Holm, B. (2005). SFC: a trainable prosodic model. *Speech communication*, 46(3-4):348–364.

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

Barra-Chicote, R., Yamagishi, J., King, S., Montero, J. M., and Macias-Guarasa, J. (2010). Analysis of statistical parametric and unit selection speech synthesis systems applied to emotional speech. *Speech communication*, 52(5):394–404.

Battenberg, E., Mariooryad, S., Stanton, D., Skerry-Ryan, R., Shannon, M., Kao, D., and Bagby, T. (2019). Effective use of variational embedding capacity in expressive end-to-end speech synthesis. *arXiv preprint arXiv:1906.03402*.

Bellec, G., Salaj, D., Subramoney, A., Legenstein, R. A., and Maass, W. (2018). Long short-term memory and learning-to-learn in networks of spiking neurons. In *NeurIPS*.

Bengio, Y., Lee, D.-H., Bornschein, J., Mesnard, T., and Lin, Z. (2015). Towards biologically plausible deep learning. *arXiv preprint arXiv:1502.04156*.

Berndt, D. J. and Clifford, J. (1994). Using dynamic time warping to find patterns in time series. In *KDD workshop*, volume 10, pages 359–370. Seattle, WA, USA:.

Bian, Y., Chen, C., Kang, Y., and Pan, Z. (2019). Multi-reference Tacotron by intercross training for style disentangling, transfer and control in speech synthesis. *arXiv preprint arXiv:1904.02373*.

Bilcu, E. B. (2008). *Text-to-phoneme mapping using neural networks*. Tampere University of Technology.

Binkowski, M., Donahue, J., Dieleman, S., Clark, A., Elsen, E., Casagrande, N., Cobo, L. C., and Simonyan, K. (2020). High fidelity speech synthesis with adversarial networks. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Bisani, M. and Ney, H. (2008). Joint-sequence models for grapheme-to-phoneme conversion. *Speech communication*, 50(5):434–451.

Bittar, A. and Garner, P. N. (2021). A Bayesian interpretation of the Light Gated Recurrent Unit. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, number CONF. IEEE.

Black, A., Taylor, P., Caley, R., and Clark, R. (1998). The festival speech synthesis system.

Bodyanskiy, Y. and Dolotov, A. (2013). A spiking neuron model based on the lambert w function. In *IJCCI*, pages 542–546.

Boersma, P. and Weenink, D. (2017). Praat: doing phonetics by computer [computer program].

Bourlard, H. and Kamp, Y. (1988). Auto-association by multilayer perceptrons and singular value decomposition. *Biological cybernetics*, 59(4):291–294.

Bozkurt, E., Erzin, E., Erdem, C. E., and Erdem, A. T. (2011). Formant position based weighted spectral features for emotion recognition. *Speech Communication*, 53(9-10):1186–1197.

Busso, C., Bulut, M., Lee, C.-C., Kazemzadeh, A., Mower, E., Kim, S., Chang, J. N., Lee, S., and Narayanan, S. S. (2008). IEMOCAP: Interactive emotional dyadic motion capture database. *Language resources and evaluation*, 42(4):335–359.

Campbell, N. and Black, A. W. (1997). Prosody and the selection of source units for concatenative synthesis. In *Progress in speech synthesis*, pages 279–292. Springer.

Campolucci, P. and Piazza, F. (2000). Intrinsic stability-control method for recursive filters and neural networks. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 47(8):797–802.

Cao, Z., Simon, T., Wei, S.-E., and Sheikh, Y. (2017). Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7291–7299.

Cauchi, B., Kodrasi, I., Rehr, R., Gerlach, S., Jukić, A., Gerkmann, T., Doclo, S., and Goetze, S. (2015). Combination of MVDR beamforming and single-channel spectral processing for enhancing noisy and reverberant speech. *EURASIP Journal on Advances in Signal Processing*, 2015(1):61.

Chapelle, O. (2007). Training a support vector machine in the primal. *Neural computation*, 19(5):1155–1178.

Chen, S.-H., Hwang, S.-H., and Wang, Y.-R. (1998). An RNN-based prosodic information synthesizer for mandarin text-to-speech. *IEEE transactions on speech and audio processing*, 6(3):226–239.

Chen, Y., Assael, Y., Shillingford, B., Budden, D., Reed, S., Zen, H., Wang, Q., Cobo, L. C., Trask, A., Laurie, B., et al. (2019). Sample efficient adaptive text-to-speech. In *International Conference on Learning Representations*.

Cho, K., van Merriënboer, B., Bahdanau, D., and Bengio, Y. (2014). On the properties of neural machine translation: Encoder–decoder approaches. *Syntax, Semantics and Structure in Statistical Translation*, page 103.

Choi, S., Han, S., Kim, D., and Ha, S. (2020). Attentron: Few-Shot Text-to-Speech Utilizing Attention-Based Variable-Length Embedding. In *Proc. Interspeech 2020*, pages 2007–2011.

Cohen, J., Kamm, T., and Andreou, A. G. (1995). Vocal tract normalization in speech recognition: Compensating for systematic speaker variability. *The Journal of the Acoustical Society of America*, 97(5):3246–3247.

Comsa, I. M., Fischbacher, T., Potempa, K., Gesmundo, A., Versari, L., and Alakuijala, J. (2020). Temporal coding in spiking neural networks with alpha synaptic function. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8529–8533. IEEE.

Cooper, E., Lai, C.-I., Yasuda, Y., Fang, F., Wang, X., Chen, N., and Yamagishi, J. (2020). Zero-shot multi-speaker text-to-speech with state-of-the-art neural speaker embeddings. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6184–6188. IEEE.

Crawford, K. (2021). Time to regulate AI that interprets human emotions. *Nature*, 592(7853):167.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314.

Dauwels, J., Vialatte, F., Rutkowski, T., and Cichocki, A. S. (2008). Measuring neural synchrony by message passing. In *Advances in neural information processing systems*, pages 361–368.

Donahue, C., Li, B., and Prabhavalkar, R. (2018). Exploring speech enhancement with generative adversarial networks for robust speech recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5024–5028. IEEE.

Dumoulin, V., Shlens, J., and Kudlur, M. (2017). A learned representation for artistic style. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Dziugaite, G. K., Roy, D. M., and Ghahramani, Z. (2015). Training generative neural networks via maximum mean discrepancy optimization. In Meila, M. and Heskes, T., editors, *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence, UAI 2015, July 12-16, 2015, Amsterdam, The Netherlands*, pages 258–267. AUAI Press.

Eichner, M., Wolff, M., and Hoffmann, R. (2004). Voice characteristics conversion for TTS using reverse VTLN. In *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages I–17. IEEE.

El Ayadi, M., Kamel, M. S., and Karray, F. (2011). Survey on speech emotion recognition: Features, classification schemes, and databases. *Pattern Recognition*, 44(3):572–587.

Eyben, F., Weninger, F., Gross, F., and Schuller, B. (2013). Recent developments in opensmile, the munich open-source multimedia feature extractor. In *Proceedings of the 21st ACM international conference on Multimedia*, pages 835–838.

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

Fernandez, R., Rendel, A., Ramabhadran, B., and Hoory, R. (2014). Prosody contour prediction with long short-term memory, bi-directional, deep recurrent neural networks. In *Fifteenth Annual Conference of the International Speech Communication Association*.

Fransen, J., Pye, D., Robinson, T., Woodland, P., and Young, S. (1994). WSJCAM0 corpus and recording description. *Cambridge University Engineering Department (CUED), Speech Group, Trumpington Street, Cambridge CB2 1PZ, UK, Tech. Rep. CUED/F-INFENG/TR*, 192.

Fujisaki, H. and Hirose, K. (1984). Analysis of voice fundamental frequency contours for declarative sentences of japanese. *Journal of the Acoustical Society of Japan (E)*, 5(4):233–242.

Fujisaki, H., Ohno, S., and Wang, C. (1998). A command-response model for F0 contour generation in multilingual speech synthesis. In *The Third ESCA/COCOSDA Workshop (ETRW) on Speech Synthesis*.

Ganin, Y. and Lempitsky, V. (2015). Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189. PMLR.

Gao, J., Chakraborty, D., Tembine, H., and Olaleye, O. (2019). Nonparallel emotional speech conversion. *Proc. Interspeech 2019*, pages 2858–2862.

Gao, Y., Singh, R., and Raj, B. (2018). Voice impersonation using generative adversarial networks. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2506–2510. IEEE.

Gardner, F. (1986). A transformation for digital simulation of analog filters. *IEEE transactions on communications*, 34(7):676–680.

Gerazov, B. and Garner, P. N. (2015). An investigation of muscle models for physiologically based intonation modelling. In *Telecommunications Forum Telfor (TELFOR), 2015 23rd*, pages 468–471. IEEE.

Gerazov, B. and Garner, P. N. (2016). An agonist-antagonist pitch production model. In *International Conference on Speech and Computer*, pages 84–91. Springer.

Gerazov, B., Honnet, P.-E., Gjoreski, A., and Garner, P. N. (2015). Weighted correlation based atom decomposition intonation modelling. In *Sixteenth Annual Conference of the International Speech Communication Association*.

Gers, F. A., Schraudolph, N. N., and Schmidhuber, J. (2002). Learning precise timing with LSTM recurrent networks. *Journal of machine learning research*, 3(Aug):115–143.

Ghahremani, P., BabaAli, B., Povey, D., Riedhammer, K., Trmal, J., and Khudanpur, S. (2014). A pitch extraction algorithm tuned for automatic speech recognition. In *2014 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 2494–2498. IEEE.

Giuliani, D. and Gerosa, M. (2003). Investigating recognition of children's speech. In *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03).*, volume 2, pages II–137. IEEE.

Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.

Goudbeek, M., Goldman, J. P., and Scherer, K. R. (2009). Emotion dimensions and formant position. In *Tenth Annual Conference of the International Speech Communication Association*.

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

Gururani, S., Gupta, K., Shah, D., Shakeri, Z., and Pinto, J. (2019). Prosody transfer in neural text to speech using global pitch and loudness features. *arXiv preprint arXiv:1911.09645*.

Haq, S., Jackson, P. J., and Edge, J. (2008). Audio-visual feature selection and reduction for emotion classification. In *Proc. Int. Conf. on Auditory-Visual Speech Processing (AVSP'08), Tangalooma, Australia*.

Helander, E., Silén, H., Virtanen, T., and Gabbouj, M. (2011). Voice conversion using dynamic kernel partial least squares regression. *IEEE transactions on audio, speech, and language processing*, 20(3):806–817.

Henter, G. E., Lorenzo-Trueba, J., Wang, X., and Yamagishi, J. (2018). Deep encoder-decoder models for unsupervised learning of controllable speech synthesis. *arXiv preprint arXiv:1807.11470*.

Hirst, D., Di Cristo, A., and Espesser, R. (2000). Levels of representation and levels of analysis for the description of intonation systems. In *Prosody: Theory and experiment*, pages 51–87. Springer.

Hirst, D. and Espesser, R. (1993). Automatic modelling of fundamental frequency using a quadratic spline function.

Hodari, Z., Moinet, A., Karlapati, S., Lorenzo-Trueba, J., Merritt, T., Joly, A., Abbas, A., Karanasou, P., and Drugman, T. (2021). Camp: a two-stage approach to modelling prosody in context. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6578–6582. IEEE.

Hodari, Z., Watts, O., and King, S. (2019). Using generative modelling to produce varied intonation for speech synthesis. In *Proc. 10th ISCA Speech Synthesis Workshop*, pages 239–244.

Hojo, N., Ohsugi, Y., Ijima, Y., and Kameoka, H. (2017). DNN-SPACE: DNN-HMM-based generative model of voice F0 contours for statistical phrase/accent command estimation. *Proc. Interspeech 2017*, pages 1074–1078.

Honnet, P.-E. and Garner, P. N. (2016). Emphasis recreation for TTS using intonation atoms. In *9th ISCA Speech Synthesis Workshop*, number EPFL-CONF-220902.

Honnet, P.-E., Gerazov, B., and Garner, P. N. (2015). Atom decomposition-based intonation modelling. In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, pages 4744–4748. IEEE.

Honnet, P.-E. J. C. (2017). *Intonation Modelling for Speech Synthesis and Emphasis Preservation*. PhD thesis, ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE.

Hornik, K., Stinchcombe, M., and White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366.

Howell, M. and Gordon, T. (2001). Continuous action reinforcement learning automata and their application to adaptive digital filter design. *Engineering Applications of Artificial Intelligence*, 14(5):549–561.

Hsu, C.-C., Hwang, H.-T., Wu, Y.-C., Tsao, Y., and Wang, H.-M. (2016). Voice conversion from non-parallel corpora using variational auto-encoder. In *2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, pages 1–6. IEEE.

Hsu, C.-C., Hwang, H.-T., Wu, Y.-C., Tsao, Y., and Wang, H.-M. (2017a). Voice conversion from unaligned corpora using variational autoencoding Wasserstein generative adversarial networks. *arXiv preprint arXiv:1704.00849*.

Hsu, W., Zhang, Y., and Glass, J. R. (2017b). Unsupervised learning of disentangled and interpretable representations from sequential data. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R., editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 1878–1889.

Hsu, W.-N., Zhang, Y., Weiss, R. J., Zen, H., Wu, Y., Wang, Y., Cao, Y., Jia, Y., Chen, Z., Shen, J., et al. (2019). Hierarchical generative modeling for controllable speech synthesis. In *International Conference on Learning Representations*.

Hunt, A. J. and Black, A. W. (1996). Unit selection in a concatenative speech synthesis system using a large speech database. In *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, volume 1, pages 373–376. IEEE.

Hunter, J. D. and Milton, J. G. (2003). Amplitude and frequency dependence of spike timing: implications for dynamic regulation. *Journal of neurophysiology*, 90(1):387–394.

Huybrechts, G., Merritt, T., Comini, G., Perz, B., Shah, R., and Lorenzo-Trueba, J. (2020). Low-resource expressive text-to-speech using data augmentation. *arXiv preprint arXiv:2011.05707*.

Inanoglu, Z. and Young, S. (2007). A system for transforming the emotion in speech: Combining data-driven conversion techniques for prosody and voice quality. In *Eighth annual conference of the international speech communication association*.

Jaitly, N. and Hinton, G. E. (2013). Vocal tract length perturbation (VTLP) improves speech recognition. In *Proc. ICML Workshop on Deep Learning for Audio, Speech and Language*, volume 117.

Jia, Y., Zhang, Y., Weiss, R., Wang, Q., Shen, J., Ren, F., Nguyen, P., Pang, R., Moreno, I. L., Wu, Y., et al. (2018). Transfer learning from speaker verification to multispeaker text-to-speech synthesis. In *Advances in neural information processing systems*, pages 4480–4490.

Jin, Z., Finkelstein, A., Mysore, G. J., and Lu, J. (2018). FFTNet: A real-time speaker-dependent neural vocoder. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2251–2255. IEEE.

Johnstone, T. and Scherer, K. R. (2000). Vocal communication of emotion. *Handbook of emotions*, 2:220–235.

Juvela, L., Bollepalli, B., Tsiaras, V., and Alku, P. (2019a). Glotnet—a raw waveform model for the glottal excitation in statistical parametric speech synthesis. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(6):1019–1030.

Juvela, L., Bollepalli, B., Yamagishi, J., Alku, P., et al. (2019b). Gelp: GAN-excited linear prediction for speech synthesis from mel-spectrogram. In *Interspeech*.

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

Kameoka, H., Kaneko, T., Tanaka, K., and Hojo, N. (2018). StarGAN-VC: Non-parallel many-to-many voice conversion using star generative adversarial networks. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 266–273. IEEE.

Kameoka, H., Roux, J. L., and Ohishi, Y. (2010). A statistical model of speech F0 contours. In *Statistical And Perceptual Audition 2010*.

Kameoka, H., Yoshizato, K., Ishihara, T., Kadowaki, K., Ohishi, Y., and Kashino, K. (2015). Generative modeling of voice fundamental frequency contours. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(6):1042–1053.

Kaneko, T., Kameoka, H., Tanaka, K., and Hojo, N. (2019). StarGAN-VC2: Rethinking conditional methods for stargan-based voice conversion. *Proc. Interspeech 2019*, pages 679–683.

Kaplan, R. M. and Kay, M. (1994). Regular models of phonological rule systems. *Computational linguistics*, 20(3):331–378.

Karaiskos, V., King, S., Clark, R. A., and Mayo, C. (2008). The blizzard challenge 2008. In *Proc. Blizzard Challenge Workshop, Brisbane, Australia*.

Karlapati, S., Moinet, A., Joly, A., Klimkov, V., Sáez-Trigueros, D., and Drugman, T. (2020a). CopyCat: Many-to-Many Fine-Grained Prosody Transfer for Neural Text-to-Speech. In *Proc. Interspeech 2020*, pages 4387–4391.

Karlapati, S., Moinet, A., Joly, A., Klimkov, V., Sáez-Trigueros, D., and Drugman, T. (2020b). CopyCat: Many-to-Many Fine-Grained Prosody Transfer for Neural Text-to-Speech. In *Proc. Interspeech 2020*, pages 4387–4391.

Kenter, T., Wan, V., Chan, C.-A., Clark, R., and Vit, J. (2019a). Chive: Varying prosody in speech synthesis with a linguistically driven dynamic hierarchical conditional variational network. In *International Conference on Machine Learning*, pages 3331–3340. PMLR.

Kenter, T., Wan, V., Chan, C.-A., Clark, R., and Vit, J. (2019b). CHiVE: Varying prosody in speech synthesis with a linguistically driven dynamic hierarchical conditional variational network. In *International Conference on Machine Learning*, pages 3331–3340. PMLR.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization.

Kingma, D. P., Salimans, T., Jozefowicz, R., Chen, X., Sutskever, I., and Welling, M. (2016). Improved variational inference with inverse autoregressive flow. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 4743–4751.

Kingma, D. P. and Welling, M. (2014). Auto-encoding variational bayes.

Klambauer, G., Unterthiner, T., Mayr, A., and Hochreiter, S. (2017). Self-normalizing neural networks. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Klimkov, V., Ronanki, S., Rohnke, J., and Drugman, T. (2019). Fine-grained robust prosody transfer for single-speaker neural text-to-speech. *Proc. Interspeech 2019*, pages 4440–4444.

Kominek, J., Black, A. W., and Ver, V. (2003). CMU ARCTIC databases for speech synthesis.

Kotani, G. and Saito, D. (2019). Voice conversion based on full-covariance mixture density networks for time-variant linear transformations. pages 75–80.

Kotani, G., Saito, D., and Minematsu, N. (2017). Voice conversion based on deep neural networks for time-variant linear transformations. In *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 1259–1262. IEEE.

Kouh, M. and Poggio, T. (2008). A canonical neural circuit for cortical nonlinear operations. *Neural computation*, 20(6):1427–1451.

Kraft, S. and Zölzer, U. (2014). BeaqleJS: HTML5 and javascript based framework for the subjective evaluation of audio quality. In *Linux Audio Conference, Karlsruhe, DE*.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105.

Latif, S., Rana, R., Khalifa, S., Jurdak, R., Epps, J., and Schuller, B. W. (2020). Multi-task semi-supervised adversarial autoencoding for speech emotion recognition. *IEEE Transactions on Affective Computing*, pages 1–1.

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

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–444.

Lee, Y. and Kim, T. (2019). Robust and fine-grained prosody control of end-to-end speech synthesis. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5911–5915. IEEE.

Lei, Y., Yang, S., and Xie, L. (2020). Fine-grained emotion strength transfer, control and prediction for emotional speech synthesis. *arXiv preprint arXiv:2011.08477*.

Lian, Z., Tao, J., Wen, Z., Liu, B., Zheng, Y., and Zhong, R. (2019). Towards fine-grained prosody control for voice conversion. *arXiv preprint arXiv:1910.11269*.

Liu, S., Cao, Y., and Meng, H. (2020). Emotional voice conversion with cycle-consistent adversarial network. *arXiv preprint arXiv:2004.03781*.

Lo, C.-C., Fu, S.-W., Huang, W.-C., Wang, X., Yamagishi, J., Tsao, Y., and Wang, H.-M. (2019). MOSNet: Deep Learning-Based Objective Assessment for Voice Conversion. In *Proc. Interspeech 2019*, pages 1541–1545.

Lorenzo-Trueba, J., Drugman, T., Latorre, J., Merritt, T., Putrycz, B., Barra-Chicote, R., Moinet, A., and Aggarwal, V. (2019). Towards achieving robust universal neural vocoding. *Proc. Interspeech 2019*, pages 181–185.

Lorenzo-Trueba, J., Henter, G. E., Takaki, S., Yamagishi, J., Morino, Y., and Ochiai, Y. (2018). Investigating different representations for modeling and controlling multiple emotions in DNN-based speech synthesis. *Speech Communication*.

Luong, H., Takaki, S., Henter, G. E., and Yamagishi, J. (2017). Adapting and controlling DNN-based speech synthesis using input codes. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4905–4909.

Ma, S., Mcduff, D., and Song, Y. (2018). Neural TTS stylization with adversarial and collaborative games. In *International Conference on Learning Representations*.

Mallat, S. G. and Zhang, Z. (1993). Matching pursuits with time-frequency dictionaries. *IEEE Transactions on signal processing*, 41(12):3397–3415.

Marelli, F. (2018). Designing second order recurrent neural networks for prosody modelling.

Marelli, F., Schnell, B., Bourlard, H., Dutoit, T., and Garner, P. N. (2019). An end-to-end network to synthesize intonation using a generalized command response model. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7040–7044. IEEE.

McAuliffe, M., Socolof, M., Mihuc, S., Wagner, M., and Sonderegger, M. (2017). Montreal forced aligner: Trainable text-speech alignment using kaldi. In *Interspeech*, volume 2017, pages 498–502.

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

Ming, H., Huang, D., Xie, L., Wu, J., Dong, M., and Li, H. (2016). Deep bidirectional LSTM modeling of timbre and prosody for emotional voice conversion. *Interspeech 2016*, pages 2453–2457.

Mirsamadi, S., Barsoum, E., and Zhang, C. (2017). Automatic speech emotion recognition using recurrent neural networks with local attention. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2227–2231.

Mixdorff, H. (2000). A novel approach to the fully automatic extraction of Fujisaki model parameters. In *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 00CH37100)*, volume 3, pages 1281–1284. IEEE.

Mohemmed, A., Schliebs, S., Matsuda, S., and Kasabov, N. (2013). Training spiking neural networks to associate spatio-temporal input–output spike patterns. *Neurocomputing*, 107:3–10.

Morise, M. (2016). D4C, a band-aperiodicity estimator for high-quality speech synthesis. *Speech Communication*, 84:57–65.

Morise, M., Yokomori, 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(7):1877–1884.

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

Narusawa, S., Minematsu, N., Hirose, K., and Fujisaki, H. (2002). A method for automatic extraction of model parameters from fundamental frequency contours of speech. In *2002 IEEE International conference on acoustics, speech, and signal processing*, volume 1, pages I–509. IEEE.

Niwa, J., Yoshimura, T., Hashimoto, K., Oura, K., Nankaku, Y., and Tokuda, K. (2018). Statistical voice conversion based on wavenet. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5289–5293. IEEE.

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

Oppenheim, A. V. and Johnson, D. H. (1972). Discrete representation of signals. *Proceedings of the IEEE*, 60(6):681–691.

Parikh, D. and Grauman, K. (2011). Relative attributes. In *2011 International Conference on Computer Vision*, pages 503–510. IEEE.

Pascanu, R., Mikolov, T., and Bengio, Y. (2012). Understanding the exploding gradient problem. *CoRR, abs/1211.5063*.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. (2017). Automatic differentiation in pytorch.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In Wallach, H. M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. B., and Garnett, R., editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 8024–8035.

Paul, D. B. and Baker, J. (1992). The design for the wall street journal-based CSR corpus. In *Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, February 23-26, 1992*.

Ping, W., Peng, K., Gibiansky, A., Arik, S. O., Kannan, A., Narang, S., Raiman, J., and Miller, J. (2018). Deep voice 3: Scaling text-to-speech with convolutional sequence learning. In *International Conference on Learning Representations*.

Pitz, M. and Ney, H. (2005). Vocal tract normalization equals linear transformation in cepstral space. *IEEE Transactions on Speech and Audio Processing*, 13(5):930–944.

Ponulak, F. and Kasiński, A. (2010). Supervised learning in spiking neural networks with ReSuMe: sequence learning, classification, and spike shifting. *Neural computation*, 22(2):467–510.

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

Prom-On, S., Xu, Y., and Thipakorn, B. (2009). Modeling tone and intonation in mandarin and english as a process of target approximation. *The Journal of the Acoustical Society of America*, 125(1):405–424.

Qian, K., Zhang, Y., Chang, S., Hasegawa-Johnson, M., and Cox, D. (2020). Unsupervised speech decomposition via triple information bottleneck. In *International Conference on Machine Learning*, pages 7836–7846. PMLR.

Qian, K., Zhang, Y., Chang, S., Yang, X., and Hasegawa-Johnson, M. (2019). AutoVC: Zero-shot voice style transfer with only autoencoder loss. volume 97 of *Proceedings of Machine Learning Research*, pages 5210–5219, Long Beach, California, USA. PMLR.

Quiroga, R. Q., Kreuz, T., and Grassberger, P. (2002). Event synchronization: a simple and fast method to measure synchronicity and time delay patterns. *Physical review E*, 66(4):041904.

Ramet, G., Garner, P. N., Baeriswyl, M., and Lazaridis, A. (2018). Context-aware attention mechanism for speech emotion recognition. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 126–131. IEEE.

Rao, K., Peng, F., Sak, H., and Beaufays, F. (2015). Grapheme-to-phoneme conversion using long short-term memory recurrent neural networks. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4225–4229. IEEE.

Ren, Y., Ruan, Y., Tan, X., Qin, T., Zhao, S., Zhao, Z., and Liu, T. (2019). Fastspeech: Fast, robust and controllable text to speech. In Wallach, H. M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. B., and Garnett, R., editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 3165–3174.

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 *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.01CH37221)*, volume 2, pages 749–752 vol.2.

Rizos, G., Baird, A., Elliott, M., and Schuller, B. (2020). StarGAN for emotional speech conversion: Validated by data augmentation of end-to-end emotion recognition. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3502–3506. IEEE.

Robinson, C., Obin, N., and Roebel, A. (2019). Sequence-to-sequence modelling of F0 for speech emotion conversion. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6830–6834. IEEE.

Ronanki, S., Watts, O., and King, S. (2017). A hierarchical encoder-decoder model for statistical parametric speech synthesis. *Proc. Interspeech 2017*, pages 1133–1137.

Rossum, M. v. (2001). A novel spike distance. *Neural computation*, 13(4):751–763.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1985). Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, 323(6088):533–536.

Rusu, C. V. and Florian, R. V. (2014). A new class of metrics for spike trains. *Neural Computation*, 26(2):306–348.

Saheer, L., Dines, J., and Garner, P. N. (2012). Vocal tract length normalization for statistical parametric speech synthesis. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(7):2134–2148.

Saheer, L., Dines, J., Garner, P. N., and Liang, H. (2010). Implementation of VTLN for statistical speech synthesis. In *SSW7*, Kyoto, Japan.

Salimans, T., Goodfellow, I. J., Zaremba, W., Cheung, V., Radford, A., and Chen, X. (2016). Improved techniques for training gans. In Lee, D. D., Sugiyama, M., von Luxburg, U., Guyon, I., and Garnett, R., editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 2226–2234.

Schnell, B. and Garner, P. N. (2018). A neural model to predict parameters for a generalized command response model of intonation. *Proc. Interspeech 2018*, pages 3147–3151.

Schnell, B. and Garner, P. N. (2019). Neural VTLN for speaker adaptation in TTS. In *Proc. 10th ISCA Speech Synthesis Workshop*, pages 29–34.

Schnell, B. and Garner, P. N. (2021a). Improving emotional TTS with an emotion intensity input from unsupervised extraction. In *Proc. 11th ISCA Speech Synthesis Workshop*.

Schnell, B. and Garner, P. N. (2021b). Investigating a neural all pass warp in modern TTS applications. *Speech Communication*.

Schnell, B., Huybrechts, G., Perz, B., Drugman, T., and Lorenzo-Trueba, J. (2021). EmoCat: Language-agnostic emotional voice conversion. In *Proc. 11th ISCA Speech Synthesis Workshop*.

Schreiber, S., Fellous, J.-M., Whitmer, D., Tiesinga, P., and Sejnowski, T. J. (2003). A new correlation-based measure of spike timing reliability. *Neurocomputing*, 52:925–931.

Schuller, B., Steidl, S., and Batliner, A. (2009). The interspeech 2009 emotion challenge. In *Tenth Annual Conference of the International Speech Communication Association*.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. (2017). Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626.

Series, B. (2014). Method for the subjective assessment of intermediate quality level of audio systems. *International Telecommunication Union Radiocommunication Assembly*.

Serrà, J., Pascual, S., and Segura Perales, C. (2019). Blow: a single-scale hyperconditioned flow for non-parallel raw-audio voice conversion. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Shah, N., Madhavi, M. C., and Patil, H. (2018). Unsupervised vocal tract length warped posterior features for non-parallel voice conversion. In *Proceedings of Interspeech*.

Shankar, R., Sager, J., and Venkataraman, A. (2019). A multi-speaker emotion morphing model using highway networks and maximum likelihood objective. In *INTERSPEECH*, pages 2848–2852.

Shechtman, S., Fernandez, R., and Haws, D. (2021). Supervised and unsupervised approaches for controlling narrow lexical focus in sequence-to-sequence speech synthesis. *arXiv preprint arXiv:2101.09940*.

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

Shikano, K., Nakamura, S., and Abe, M. (1991). Speaker adaptation and voice conversion by codebook mapping. In *1991 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 594–597. IEEE.

Shinoda, K. and Watanabe, T. (2000). MDL-based context-dependent subword modeling for speech recognition. *Acoustical Science and Technology*, 21(2):79–86.

Shrikumar, A., Greenside, P., Shcherbina, A., and Kundaje, A. (2016). Not just a black box: Learning important features through propagating activation differences. *arXiv preprint arXiv:1605.01713*.

Silverman, K., Beckman, M., Pitrelli, J., Ostendorf, M., Wightman, C., Price, P., Pierrehumbert, J., and Hirschberg, J. (1992). ToBI: A standard for labeling english prosody. In *Second international conference on spoken language processing*.

Simonyan, K., Vedaldi, A., and Zisserman, A. (2013). Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*.

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

Smilkov, D., Thorat, N., Kim, B., Viégas, F., and Wattenberg, M. (2017). Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*.

Srinivas, S. and Fleuret, F. (2019). Full-gradient representation for neural network visualization. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Stanton, D., Wang, Y., and Skerry-Ryan, R. (2018). Predicting expressive speaking style from text in end-to-end speech synthesis. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 595–602. IEEE.

Strom, V., Clark, R. A., and King, S. (2006). Expressive prosody for unit-selection speech synthesis. In *Ninth International Conference on Spoken Language Processing*.

Stylianou, Y., Cappé, O., and Moulines, E. (1998). Continuous probabilistic transform for voice conversion. *IEEE Transactions on speech and audio processing*, 6(2):131–142.

Sun, G., Zhang, Y., Weiss, R. J., Cao, Y., Zen, H., and Wu, Y. (2020). Fully-hierarchical fine-grained prosody modeling for interpretable speech synthesis. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6264–6268. IEEE.

Sundararajan, M., Taly, A., and Yan, Q. (2017). Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pages 3319–3328. PMLR.

Sundermann, D. and Ney, H. (2003). VTLN-based voice conversion. In *Proceedings of the 3rd IEEE International Symposium on Signal Processing and Information Technology (IEEE Cat. No. 03EX795)*, pages 556–559. IEEE.

Sundermann, D., Ney, H., and Hoge, H. (2003). VTLN-based cross-language voice conversion. In *2003 IEEE Workshop on Automatic Speech Recognition and Understanding (IEEE Cat. No. 03EX721)*, pages 676–681. IEEE.

Taigman, Y., Wolf, L., Polyak, A., and Nachmani, E. (2018). Voiceloop: Voice fitting and synthesis via a phonological loop. In *International Conference on Learning Representations*.

Taigman, Y., Yang, M., Ranzato, M., and Wolf, L. (2014). Deepface: Closing the gap to human-level performance in face verification. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1701–1708.

Tarantino, L., Garner, P. N., and Lazaridis, A. (2019). Self-attention for speech emotion recognition. In *Interspeech*, pages 2578–2582.

Taylor, P. (2000). Analysis and synthesis of intonation using the tilt model. *The Journal of the acoustical society of America*, 107(3):1697–1714.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288.

Tokuda, K., Kobayashi, T., Masuko, T., and Imai, S. (1994). Mel-generalized cepstral analysis-a unified approach to speech spectral estimation. In *Third International Conference on Spoken Language Processing*.

Tokuda, K., Yoshimura, T., Masuko, T., Kobayashi, T., and Kitamura, T. (2000). Speech parameter generation algorithms for HMM-based speech synthesis. In *Acoustics, Speech, and Signal Processing, 2000. ICASSP'00. Proceedings. 2000 IEEE International Conference on*, volume 3, pages 1315–1318. IEEE.

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

Torres, H. and Gurlekian, J. (2015). Novel estimation method for the superpositional intonation model. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(1):151–160.

Traber, C. (1991). F0 generation with a data base of natural F0 patterns and with a neural network. In *The ESCA Workshop on Speech Synthesis*.

Tyagi, S., Nicolis, M., Rohnke, J., Drugman, T., and Lorenzo-Trueba, J. (2020). Dynamic prosody generation for speech synthesis using linguistics-driven acoustic embedding selection. *Proc. Interspeech 2020*, pages 4407–4411.

Ulyanov, D., Vedaldi, A., and Lempitsky, V. (2016). Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*.

Um, S.-Y., Oh, S., Byun, K., Jang, I., Ahn, C., and Kang, H.-G. (2020). Emotional speech synthesis with rich and granularized control. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7254–7258. IEEE.

Valin, J.-M. and Skoglund, J. (2019). LPCNet: Improving neural speech synthesis through linear prediction. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5891–5895. IEEE.

van den 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 *9th ISCA Speech Synthesis Workshop*, pages 125–125.

van den Oord, A., Vinyals, O., et al. (2017). Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, pages 6306–6315.

Van Santen, J. P. and Möbius, B. (2000). A quantitative model of F0 generation and alignment. In *Intonation*, pages 269–288. Springer.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6000–6010.

Veaux, C., Yamagishi, J., MacDonald, K., et al. (2017). CSTR VCTK corpus: English multi-speaker corpus for CSTR voice cloning toolkit. *University of Edinburgh. The Centre for Speech Technology Research (CSTR)*.

Victor, J. D. and Purpura, K. P. (1997). Metric-space analysis of spike trains: theory, algorithms and application. *Network: computation in neural systems*, 8(2):127–164.

Vlasenko, B., Prylipko, D., Philippou-Hübner, D., and Wendemuth, A. (2011). Vowels formants analysis allows straightforward detection of high arousal acted and spontaneous emotions. In *Twelfth Annual Conference of the International Speech Communication Association*.

Wang, X., Lorenzo-Trueba, J., Takaki, S., Juvela, L., and Yamagishi, J. (2018a). A comparison of recent waveform generation and acoustic modeling methods for neural-network-based speech synthesis. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4804–4808. IEEE.

Wang, X., Takaki, S., and Yamagishi, J. (2017a). An RNN-based quantized F0 model with multi-tier feedback links for text-to-speech synthesis. In *INTERSPEECH*, pages 1059–1063.

Wang, X., Takaki, S., and Yamagishi, J. (2019a). Neural source-filter waveform models for statistical parametric speech synthesis. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:402–415.

Wang, X., Takaki, S., Yamagishi, J., King, S., and Tokuda, K. (2019b). A vector quantized variational autoencoder (VQ-VAE) autoregressive neural F0 model for statistical parametric speech synthesis. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:157–170.

Wang, Y., Skerry-Ryan, R., Stanton, D., Wu, Y., Weiss, R. J., Jaitly, N., Yang, Z., Xiao, Y., Chen, Z., Bengio, S., Le, Q., Agiomyrgiannakis, Y., Clark, R., and Saurous, R. A. (2017b). Tacotron: Towards end-to-end speech synthesis. In *Interspeech 2017, 18th Annual Conference of the International Speech Communication Association, Stockholm, Sweden, August 20-24, 2017*, pages 4006–4010. ISCA.

Wang, Y., Stanton, D., Zhang, Y., Ryan, R.-S., Battenberg, E., Shor, J., Xiao, Y., Jia, Y., Ren, F., and Saurous, R. A. (2018b). Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis. In *International Conference on Machine Learning*, pages 5180–5189. PMLR.

Watts, O., Henter, G. E., Fong, J., and Valentini-Botinhao, C. (2019). Where do the improvements come from in sequence-to-sequence neural TTS? In *10th ISCA Speech Synthesis Workshop. ISCA, Vienna, Austria (September 2019)*.

Werbos, P. J. (1990). Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560.

Whitehill, M., Ma, S., McDuff, D., and Song, Y. (2020). Multi-reference neural TTS stylization with adversarial cycle consistency. *Proc. Interspeech 2020*, pages 4442–4446.

Woodland, P. C., Odell, J. J., Valtchev, V., and Young, S. J. (1994). Large vocabulary continuous speech recognition using HTK. In *ICASSP (2)*, pages 125–128.

Wu, J., Yılmaz, E., Zhang, M., Li, H., and Tan, K. C. (2020). Deep spiking neural networks for large vocabulary automatic speech recognition. *Frontiers in neuroscience*, 14:199.

Wu, Z., Watts, O., and King, S. (2016). Merlin: An open source neural network speech synthesis system. *Proc. SSW, Sunnyvale, USA*.

Xu, J., Tan, X., Ren, Y., Qin, T., Li, J., Zhao, S., and Liu, T.-Y. (2020). Lrspeech: Extremely low-resource speech synthesis and recognition. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2802–2812.

Yao, K. and Zweig, G. (2015). Sequence-to-sequence neural net models for grapheme-to-phoneme conversion. In *Sixteenth Annual Conference of the International Speech Communication Association*.

Yasuda, Y., Wang, X., Takaki, S., and Yamagishi, J. (2019). Investigation of enhanced Tacotron text-to-speech synthesis systems with self-attention for pitch accent language. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6905–6909. IEEE.

Yoshimura, T., Tokuda, K., Masuko, T., Kobayashi, T., and Kitamura, T. (1999). Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis. In *Sixth European Conference on Speech Communication and Technology*.

Yu, C., Lu, H., Hu, N., Yu, M., Weng, C., Xu, K., Liu, P., Tuo, D., Kang, S., Lei, G., Su, D., and Yu, D. (2020). Durian: Duration informed attention network for speech synthesis. In Meng, H., Xu, B., and Zheng, T. F., editors, *Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, Virtual Event, Shanghai, China, 25-29 October 2020*, pages 2027–2031. ISCA.

Zen, H., Agiomyrgiannakis, Y., Egberts, N., Henderson, F., and Szczepaniak, P. (2016). Fast, compact, and high quality LSTM-RNN based statistical parametric speech synthesizers for mobile devices. *Interspeech 2016*, pages 2273–2277.

Zen, H., Senior, A., and Schuster, M. (2013). Statistical parametric speech synthesis using deep neural networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 7962–7966. IEEE.

Zhang, H., Goodfellow, I., Metaxas, D., and Odena, A. (2019a). Self-attention generative adversarial networks. In *International Conference on Machine Learning*, pages 7354–7363.

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

Zhang, Z., Wu, B., and Schuller, B. (2019c). Attention-augmented end-to-end multi-task learning for emotion prediction from speech. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6705–6709. IEEE.

Zhao, Y., Takaki, S., Luong, H.-T., Yamagishi, J., Saito, D., and Minematsu, N. (2018). Wasserstein GAN and waveform loss-based acoustic model training for multi-speaker text-to-speech synthesis systems using a wavenet vocoder. *IEEE access*, 6:60478–60488.

Zhou, K., Sisman, B., and Li, H. (2020a). Transforming spectrum and prosody for emotional voice conversion with non-parallel training data. In *Proc. Odyssey 2020 The Speaker and Language Recognition Workshop*, pages 230–237.

Zhou, K., Sisman, B., Zhang, M., and Li, H. (2020b). Converting anyone's emotion: Towards speaker-independent emotional voice conversion. *arXiv preprint arXiv:2005.07025*.

Zhu, X., Yang, S., Yang, G., and Xie, L. (2019). Controlling emotion strength with relative attribute for end-to-end speech synthesis. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 192–199. IEEE.

Zolnay, A., Schluter, R., and Ney, H. (2005). Acoustic feature combination for robust speech recognition. In *Proceedings.(ICASSP'05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, volume 1, pages I–457. IEEE.

Curriculum Vitae

Personal

Name:	Schnell
First name:	Bastian
Date of birth:	30/12/91
Nationality:	German
Address:	Heubnerweg 11a, 14059 Berlin, Germany
Telephone:	0049 163 9141 835
Email:	bastian.schnell@t-online.de

Education

Master of Science — Oct. 14 - Nov. 16

Hamburg University of Technology, Hamburg
Computer Science and Engineering
(grade 1.5, very good)

Exchange student for one semester at Linköping University, Sweden

Thesis: A hybrid architecture for path finding in dynamic environments

Adviser: Prof. Sybille Schupp

Bachelor of Science — Oct. 11 - Sep. 14

Hamburg University of Technology, Hamburg
Computer Science and Engineering
(grade 1.5, very good)

Thesis: Development of a Record/Replay-Interface for the Explorative Design of Driving Scenarios

Adviser: Prof. Thilo Pionteck

Work Experience

Amazon Development Center Germany GmbH, Berlin, Germany — Oct. 21 - now

Applied Scientist
Machine Learning at TTS Research for Alexa AI. Develop novel algorithms and modeling techniques to advance the state of the art in speech synthesis, build industry-leading Speech and Language technology for the next generation of synthetic voices

École polytechnique fédérale de Lausanne (EPFL), Idiap Research Institute Martigny, Switzerland — May 17 - Aug. 21

PhD Student
Research in multi-lingual affective speech synthesis; combining speech processing techniques and physiologically plausable models with deep learning to reduce the number of trainable parameters and thus the amount of required training data; also allow physiological interpretation

Thesis director: Prof. Hervé Bourlard

Adviser: Dr. Philip N. Garner

	Ski School Interski Nauders Nauders, Austria	Ski Instructor Group and private lessons on skiing for children, teenagers, and adults; obtained Tyrol instructor license (Anwärter Ski)	Dec. 16 - April 17
	German Aerospace Center (DLR), Institute of Transportation Systems Braunschweig, Germany	Student Assistant Development of a record/replay extension for the traffic scenario creation on a touch table as Bachelor Thesis; implementation in C++; documentation of created software; assistance with project work; literature researches	June 14 - Sep. 14
	YXLON International GmbH Hamburg, Germany	Intern Compulsory internship for Bachelor degree; basic metalworking; performance test of SPS Ethernet interface with TCP and UDP; conversion of SPS to PC communication from Profibus to Ethernet; creation of test scripts for software testing; simulation of SPS through C# implementation	April 14 - June 14
	ErholungsWerk Post Postbank Telekom e.V. Scheidegg, Germany	Alternative service in lieu of military service Versatile work in a holiday resort; entertainment, guide, welcome service, etc.	July 10 - Sep. 11
Teaching Experience	**Deep Learning** EPFL, Switzerland	Teaching assistant Helping students during the exercise and grading projects; the course gives a detailed introduction to deep-learning within the PyTorch framework	March 19 - July 19

Personal Skills

Computer Skills

Programming:	Python, C#, C++, C, Bash, Java, Javascript, LUA
Applications and Frameworks:	PyTorch, Torch, Git, PyCharm, Eclipse, Visual Studio, Sun Grid Engine
Platforms:	Windows, Linux, MAC

Languages

Mother tongue: German

Other languages:

	Understanding		Speaking		Writing
	Listening	Reading	Spoken interaction	Spoken production	
English	C1	C2	C1	C1	C1
French	A2	A2	A2	A2	A2

Projects

Main developer
A Python-based modular toolbox for building Deep Neural Network models (using PyTorch) for statistical parametric speech synthesis
(https://github.com/idiap/IdiapTTS)

Publications

Journal

Investigating a Neural All Pass Warp in Modern TTS Applications
Bastian Schnell, Philip N. Garner
Speech Communication

Papers

EmoCat: Language Agnostic Emotional Voice Conversion
Bastian Schnell, Goeric Hybrechts, Bartek Perz, Thomas Drugman, Jaime Lorenzo-Trueba
Proc. 11th ISCA Speech Synthesis Workshop, 2021

Improving Emotional TTS with an Emotion Intensity Input from Unsupervised Extraction
Bastian Schnell, Philip N. Garner
Proc. 11th ISCA Speech Synthesis Workshop, 2021

Neural VTLN for Speaker Adaptation in TTS
Bastian Schnell, Philip N. Garner
Proc. 10th ISCA Speech Synthesis Workshop, 2019

An End-to-end Network to Synthesize Intonation Using a Generalized Command Response Model
François Marelli, Bastian Schnell, Hervé Bourlard, Thierry Dutoit, Philip N Garner
IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2019

A Neural Model to Predict Parameters for a Generalized Command Response Model of Intonation
Bastian Schnell, Philip N. Garner
Proc. Interspeech 2018, 2018