Investigating a Neural All Pass Warp in Modern TTS Applications[☆]

Bastian Schnell[a,b,*], Philip N. Garner[b]

[a]*Idiap Research Institute, Centre du Parc, Rue Marconi 19, PO Box 592, Martigny, Switzerland*
[b]*École Polytechnique Fédérale de Lausanne (EPFL), Route Cantonale, 1015 Lausanne, Switzerland*

Abstract

We present a neural implementation of the all pass warp (APW) previously used for vocal tract length normalisation. This includes an efficient back-propagation, which can easily be integrated in modern neural network frameworks. The APW offers a low-dimensional control to alter the spectrum, which by design generalises over different speakers. We investigate the APW in two tasks required for future dialogue or translation agents, and provide a fairly thorough literature review for both: 1) Zero-shot speaker adaptation to allow keeping the source speaker identity with very small amounts of data. Experiments show increased speaker similarity and prove that the APW increases the generalisability of a multi-speaker model. 2) Emotional speech synthesis to translate or produce affective cues. To the best of our knowledge this is the first attempt on emotional speech synthesis with an APW. While the APW is not able to increase expressiveness or audio quality, our analysis shows that the warping correlates with the level of valence in the emotion. This work should enable future research on emotion translation during machine translation.

Keywords: All pass warp, VTLN, zero-shot speaker adaptation, emotional TTS, VAE

1. Introduction

Text to speech synthesis (TTS) has become ubiquitous in recent years, mainly in the context of agents associated with mobile telephony. Coupled with (automatic) speech recognition (ASR), the technology enables hands free access to information, often including translation between languages. As the underlying dialogue managers evolve, we are interested in general in enabling an appropriate evolution of TTS to enable higher level functions. In particular, we want to enable *affective* TTS, where expression and emotion can be conveyed. There are at least two clear cases where this is desirable: The simplest is in speech to speech translation, where one would like the translation of an utterance in L2 to reflect the emotion and expression that were conveyed in L1. This involves the ability to recognise, translate and reproduce such affective cues. The translation case also requires that the synthetic voice be adapted to match the identity of the L1 speaker. The second, more difficult, case is that of dialogue agents. For instance, if either party has misunderstood the other, it may be appropriate to repeat with carefully chosen emphasis; if a human participant becomes angry or frustrated, it may be appropriate for an agent to react more sympathetically. This all requires control of affect.

Affect is certainly influenced by prosody. Normally taken to be pitch, energy and duration, each of these elements is one-dimensional. Certainly a translation agent can detect and reproduce them; it is reasonable to suppose that a dialogue agent could also produce these cues. However, it is known (see, e.g., Vlasenko et al., 2011) that spectral features are also warped when conveying certain emotions. Spectra are multi-dimensional and hence more difficult to handle.

Our goal in this paper is to show that an all-pass warp (APW) is a strong candidate to provide the spectral warping required in emotional speech synthesis. The APW is known in the field for its application to vocal tract length normalization (VTLN). VTLN originated in ASR to compensate for the fact that the vocal tract length of different speakers varies (it is longer in males than in females) with an inversely proportional effect on the formant frequencies (Cohen et al., 1995; Zolnay et al., 2005; Giuliani and Gerosa, 2003; Jaitly and Hinton, 2013). Reciprocally, the warp can be used to synthesise speech of different speakers (Sundermann and Ney, 2003; Saheer et al., 2012). The APW arises because a spectral warp can be cast as a linear transformation in the cepstrum (Pitz and Ney, 2005).

In principle, then, the APW is capable of implementing the speaker-dependent warping required by speech to speech translation, and the affect-dependent warping required for dialogue. The former case was demonstrated in a preliminary version of this paper (Schnell and Garner, 2019) at SSW, some content of which is repeated here.

[☆]This work has been conducted with the support of the Swiss NSF under grant number 165545: Multilingual Affective Speech Synthesis (MASS); http://p3.snf.ch/Project-165545 and 185010: Neural Architectures for Speech Technology (NAST); http://p3.snf.ch/Project-185010

[*]Corresponding author
Email addresses: bastian.schnell@idiap.ch (Bastian Schnell), phil.garner@idiap.ch (Philip N. Garner)

The contributions of this work are

1. an implementation of the APW as a neural network component. The component can be added to a neural network as easily as adding a linear layer. It allows efficient forward and backward propagation. The input to the component is the explicit warping value, which allows external control during inference. The additional control is lacking in related work using speaker embeddings extracted by reference encoder networks.
2. a completion of the study of utilizing the APW in multi-speaker systems. In the preliminary version of this paper (Schnell and Garner, 2019) we have already investigated multi-speaker TTS and few-shot speaker adaptation. Here we investigate the APW in zero-shot speaker adaptation to complete the study.
3. an investigation of using an APW for emotional TTS. To the best of our knowledge this has not been done before.

We will explain the tasks of contribution 2. and 3. in detail in the following.

In a first group on **Zero-shot speaker adaptation**, we aim to develop TTS systems which are able to synthesise speech of a speaker unseen during training with very small amounts of data from that target speaker. They are of special interest when designing personalised voice assistants with data recorded by non-experts. We had previously demonstrated that a time-dependent APW layer leads to better speaker-adaptation performance for small amounts of adaptation data (~25 seconds, also called few-shot adaptation). We hence expect it to perform equally well in the zero-shot scenario. In this context we investigate encoder-decoder models (new paradigm) which form the current state-of-the-art in TTS.

In a second group on **Emotional TTS**, we attempt to synthesise speech with different emotions. Today's voice assistants are still mostly limited to neutral speech and more expressive styles have only been explored recently (Aggarwal et al., 2020; Skerry-Ryan et al., 2018). While it is possible to generate speech of a specific style with a sufficient amount of recordings of that style, it is too expensive and time consuming to record a database for each style and multiple languages. Techniques to improve the generalisability of models trained on limited data are needed. Inspired by research on emotion recognition we show that some emotions cause a formant shift on the investigated emotional database. We expect the APW to be helpful because it is an efficient control for formant shifting which by design generalises over all speakers.

The paper is presented as four main parts: We begin in section 2 with a literature review on the different aspects related to this work covering all-pass warp methods in TTS, and notably style transfer. In sections 3 and 4 we detail a common VTLN implementation in form of a bilinear transformation (i.e. a non-complex all pass warp) and show how it can be integrated as a back-propagatable component in a modern neural network framework. In section 5, we present the first group of experiments above. This is a low-risk experiment in that a-priori we know that the APW is capable of doing speaker adaptation; here we attempt a "difficult" scenario. The second group of experiments on emotional TTS is presented in section 6. This is a higher risk case; we are not aware of previous attempts to do emotion adaptation using an APW.

2. Related Work

This section covers related work in zero-shot speaker adaptation, affective speech synthesis, and the all pass warp transformation in a three-fold way. In the first part we describe zero-shot and few-shot speaker adaptation methods for adapation data with and without transcriptions. The second part gives a list of recent unsupervised methods to achieve affective speech synthesis. The last part describes work related to our proposed all pass warp layer, which has been used only for multi-speaker systems so far.

2.1. Few- & Zero-shot Speaker Adaptation

Few- & zero-shot speaker adaptation means to create a TTS system which sounds like one or multiple target speakers unseen during training, while using only a very short amount of adaptation data for each of the target speakers. The boarder 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 adapation 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 adapation data through a reference encoder network. Acoustic features are extracted from the adapation 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.

2.1.1. Not transcribed adapation 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 not transcribed 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 less 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.

Cooper et al. (2020) use a speaker encoder network pre-trained on a speaker verification task in combination with a Tacotron (Wang et al., 2017) 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 most performant.

In contrast to both works, we train a reference encoder network to produce speaker embeddings and do not use a pre-trained model. We also only input the reference embeddings to the attention, because we use a parallel decoder without pre-net (see section 5.3.4). However, the method we propose would likely benefit from speaker-embeddings from a pre-trained model as well, but its proof lies beyond the scope of this work.

2.1.2. Transcribed adapation data

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. We did not experiment with the WaveNet architecture in this work.

2.2. Affective Speech Synthesis

Since the publication of Tacotron with Global Style Tokens (GST) (Wang et al., 2018) 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 work we do not rely on any of those methods but on simple trained 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.

2.2.1. Global Style Tokens

Tacotron with Global Style Tokens (GST) (Wang et al., 2018) uses a reference encoder to compress the prosody of a variable length audio signal into a fixed-length vector which is called 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.

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 simply 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.

Reference encoders to extract speaker or emotion embeddings have been proposed in various other works as well (Arik et al., 2018; Nachmani et al., 2018; Choi et al., 2020; Skerry-Ryan et al., 2018; Lian et al., 2019; Klimkov et al., 2019; Gururani et al., 2019; Battenberg et al., 2019; Bian et al., 2019; Whitehill et al., 2020). While all of them (the ones above included) achieve good results in multi-speaker scenarios and also allow zero-shot adaptation, 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 speech to full emotional speech.

The APW we propose here is an alternative to the techniques listed in this section 2.2.1. While we also use a reference encoder similar to Tacotron GST to approach the zero-shot scenario, the proposed APW does not rely on it, but instead provides a low-dimensional control (the level of warping). In contrast to the reference embeddings, which do not reveal an obvious structure, the control of the APW is interpretable and allows linear interpolation with guaranteed high audio quality by design.

2.2.2. 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. (2019) 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. While we do use a similar VAE reference encoder in our model, we do not aim to use it as a control of speaker or style.

2.3. 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).

Shah et al. (2018) trained two deep neural networks (DNNs) to imitade 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 our work they only train a DNN to behave like a VTLN (implicit), but they do not provide a neural network component that explicitly implements it. It also means that 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 hidden Markov models. 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.

3. Preliminaries

The all pass warp technique we present is inspired by a well known technique for speaker adaptation in ASR and TTS: vocal tract length normalization (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. This shift equals a linear transformation, i.e. a warping, in the cepstral domain (Pitz and Ney, 2005). Usually an $N \times N$ warping matrix $\boldsymbol{A}$ pre-multiplies N mel-cepstral coefficients to produce their warped representation. As in previous work we use a bilinear transform to generate $\boldsymbol{A}_\alpha$ (it only depends on a single warping parameter α). The APW is also key to the mel generalised cepstrum (MGC) of 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

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

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

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

where we extended the formula by the part marked *added* to get results valid for negative alphas as well. A qualitative representation of the warping matrix $\boldsymbol{A}_\alpha$ can be seen in Figure 1.

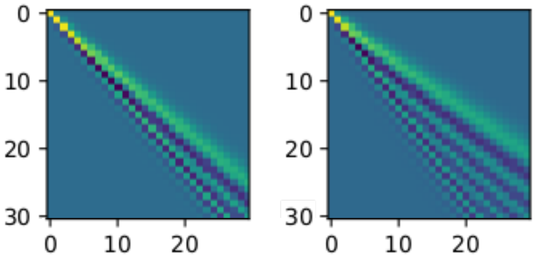

Figure 1: Qualitative representation of a VTLN warping matrix for a bilinear transform (left: $\alpha = 0.1$, right: $\alpha = 0.2$).

Warping a mel-cepstral coefficient vector $x = (c_1, \dots, c_N)^T$, or its extended version with deltas (Δ) and double deltas (Δ^2) of a single frame is as simple as equation 3 and 4.

$$\boldsymbol{x}_\alpha = \boldsymbol{A}_\alpha \boldsymbol{x}, \tag{3}$$

$$\begin{bmatrix} \boldsymbol{x}_\alpha \\ \Delta \boldsymbol{x}_\alpha \\ \Delta^2 \boldsymbol{x}_\alpha \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_\alpha & 0 & 0 \\ 0 & \boldsymbol{A}_\alpha & 0 \\ 0 & 0 & \boldsymbol{A}_\alpha \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \Delta \boldsymbol{x} \\ \Delta^2 \boldsymbol{x} \end{bmatrix}. \tag{4}$$

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. Neural Network Implementation

In preliminary work (Schnell and Garner, 2019) we have developed an implementation[1] in the PyTorch framework which we will describe again here. On the one hand, we found that any recursive structure based on equation 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 $\boldsymbol{A}_{\boldsymbol{\alpha}}$ directly with equation 2 recomputes many factorials each time. We proposed an efficient implementation that splits the constant and variable parts of equation 2. Equation 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).

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

$$\boldsymbol{A}^{3D}_{k,l,2n+k-l} = \begin{cases} \frac{1}{(l-1)!} \binom{l}{n} \frac{(k+n-1)!}{(k+n-l)!} (-1)^{n+l+k} & \text{if } l\text{-}k \le n \le l, \\ & n \ge 0 \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

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 polynomials in $\boldsymbol{\alpha}$. A comparison with a matrix computed recursively with equation 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 advise using double precision computation for $N > 35$.

4.1. 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:

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

Step 3

$$\frac{\partial L}{\partial \boldsymbol{x}} = \frac{\partial L}{\partial \tilde{\boldsymbol{x}}}\frac{\partial \tilde{\boldsymbol{x}}}{\partial \boldsymbol{x}} = \Delta_{\tilde{\boldsymbol{x}}} * \boldsymbol{A}_{\alpha}^{T}, \qquad (6)$$

$$\frac{\partial L}{\partial \boldsymbol{A}_{\alpha}} = \Delta_{\tilde{\boldsymbol{x}}} * \boldsymbol{x} = \Delta_{\boldsymbol{A}_{\alpha}}, \qquad (7)$$

Step 2

$$\frac{\partial L}{\partial \boldsymbol{\alpha}} = \Delta_{\boldsymbol{A}_{\alpha}} * \boldsymbol{A}^{3D} = \Delta_{\boldsymbol{\alpha}}, \qquad (8)$$

Step 1

$$\frac{\partial L}{\partial \alpha} = \Delta_{\boldsymbol{\alpha}} \cdot \begin{pmatrix} 0 & 1 & 2\alpha & \dots & (2N-1)\alpha^{2N-2} \end{pmatrix}. \qquad (9)$$

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. In previous work (Schnell and Garner, 2019) we reported memory problems when using our all pass warp layer. We have since improved our implementation to achieve minimal memory overhead.

4.2. Model Integration

We integrate our proposed all pass warp layer into TTS neural network architectures by simply stacking it on top (Figure 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 our previous work we have used speaker embeddings concatenated with the penultimate pre-net layer output to use the all pass warping layer for speaker adaptation. We will use the same configuration in the zero-shot speaker adaptation experiments. Additionally, we will use it with emotion embeddings to perform the emotion adaptation experiments. The activations are passed through a *tanh* non-linearity and scaled to be in a range that makes sense for the task. We have used a scale of 0.2 in previous work because any higher warpings result in very unnatural speech. 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

$$\alpha = \frac{\alpha_i + \alpha_j}{1 + \alpha_i \alpha_j}. \qquad (10)$$

Our implementation combines multiple warping factors with equation 10 first and then builds a single warping matrix to minimize computation. However, the experiments in this work do not include multiple warpings.

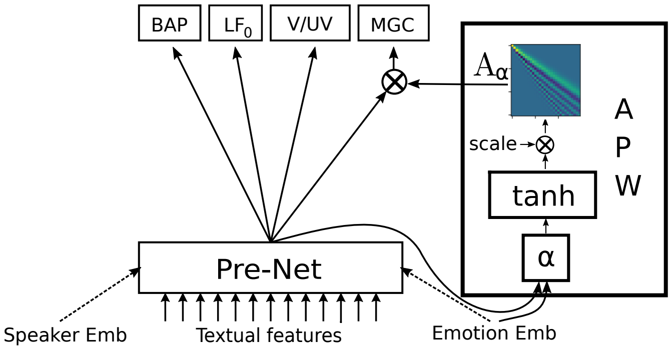

Figure 2: Network structure with an APW layer. The α parameter is estimated per frame from the pre-network. The layer also has access to some embeddings that influence the warping. The figure shows the embedding for a multi-speaker emotional TTS system.

5. Experiment: Zero-shot speaker adaptation

In this experiment we investigate the use of an APW in a modern encoder-decoder model for zero-shot speaker adapation on WSJCAM0. Our hypothesis is that the APW can play off its abilities in the zero-shot adaptation scenario, because it generalises over speakers by design. At the same time we want to prove its effectiveness with modern encoder-decoder models of the new paradigm, which we were lacking in our preliminary work (Schnell and Garner, 2019). 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 have no hypothesis that it will 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.

5.1. WSJCAM0 database

We use the big British English database WSJCAM0 (Fransen et al., 1994), because it resembles the same style as the database we use in the other experiment (section 6). We use only the head-mounted close-talking microphone recordings of the training set consisting of 92 speakers with 90 utterances each. The audio was recorded at 16 kHz. To compensate for loudness differences we use a loudness normalization technique (Equation 11) to normalize all samples to an average root-mean squared value of $RMS = 0.1$.

$$\tilde{x} = x * \sqrt{\frac{T * RMS^2}{\sum^{T}(x - x_{mean})^2}}. \qquad (11)$$

We also found background noise to degrade performance 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.

5.2. Features

We use Festival (Black et al., 1998) to extract phone sequences from text, and HTK (Woodland et al., 1994) to compute forced-alignments with context-independent Hidden-Markov-Models. From the aligned phoneme sequences we generate question labels with 425 text-derived binary and numerical features normalized to [0.01, 0.99] and duration labels where we sum the duration of the five states per phoneme to get phoneme durations. We use the WORLD vocoder (Morise et al., 2016) (D4C edition (Morise, 2016)) for the extraction of log F_0 (LF_0), 30-dimensional MGC, and one Band Aperiodicity (BAP) at 5 ms frame step. We interpolate LF_0 before training and add a binary V/UV flag to represent voicing information. We perform mean/variance normalization for all but V/UV. In all experiments waveforms are generated with the WORLD vocoder. We were not able to produce better waveforms with a WaveNet vocoder (van den Oord et al., 2016) based on synthetic WORLD vocoder features.

5.3. Model architecture

We use a state-of-the-art encoder-decoder architecture 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.

5.3.1. 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 Rectified Linear Unit (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 question labels of the old paradigm, thus providing context information at each step.

5.3.2. Reference encoder

We use a similar reference encoder as in the Tacotron GST paper (Wang et al., 2018) 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)).

5.3.3. 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. 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.

5.3.4. Autoregressive Decoder

The autoregressive decoder-RNN consists of one fully connected layer of 512 ReLU units followed by a stack of two uni-directional 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 hidden ReLU units and a single uni-directional 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 loop of *acoustic output* $\xrightarrow{\text{audio-encoder}}$ *hidden representation* $\xrightarrow{\text{decoder-RNN}}$ $\xrightarrow{\text{linear projection}}$ *acoustic output* as an auto-encoder, thus both networks (the decoder-RNN and the audio-encoder) should mirror each other or at least have similar capabilities.

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 3 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 5.3.6) for our experiments which still outperforms the old paradigm models (details in section 6.2).

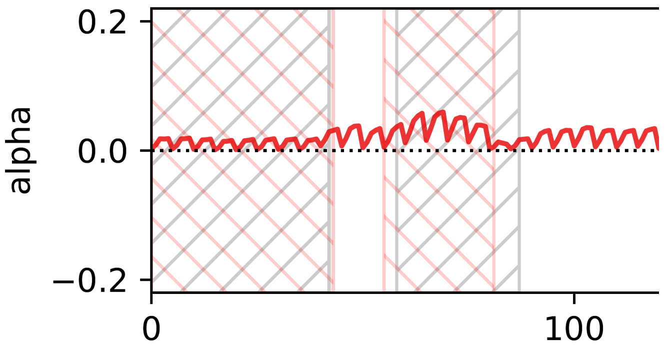

Figure 3: Predicted warping value of an autoregressive decoder which predicts a chunk of five frames per step. Each chunk shows a "loading" phase, where the first frame receives nearly no warping, then the warping raises quickly to a stable value for the last three frames. This behaviour shows the incompatibility of the APW with autoregressive decoders. Additionally the plot shows ground truth V/UV (grey, hatched upwards) and predicted V/UV (red, hatched downwards).

5.3.5. 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 Maximum Likelihood Parameter Estimation (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.

5.3.6. 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 in Karlapati et al. (2020); Qian et al. (2019). Details are not given in Karlapati et al. (2020) 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.

5.3.7. APW model

We stack the APW with an alpha range of ± 0.2 on the parallel decoder similar to Figure 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.

5.4. Training

The model is trained with an L1 loss on the predicted acoustic features and a 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 5.3.3) the model generates features aligned with the target so that we can compute the validation loss without teacher forcing for more accurate results.

5.5. 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.

5.6. 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 of the tests.

The results show slight improvements in speaker similarity at the cost of audio quality (Table 1). 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), 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 1: 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		
	Baseline	**APW**	**Same**
female	22.1	29.3	48.6
male	25.7	30.9	43.4
combined	23.8	30.1	46.1

	Audio quality		
	Baseline	**APW**	**Same**
female	29.3	28.3	42.4
male	36.1	32.0	31.9
combined	32.5	30.1	37.4

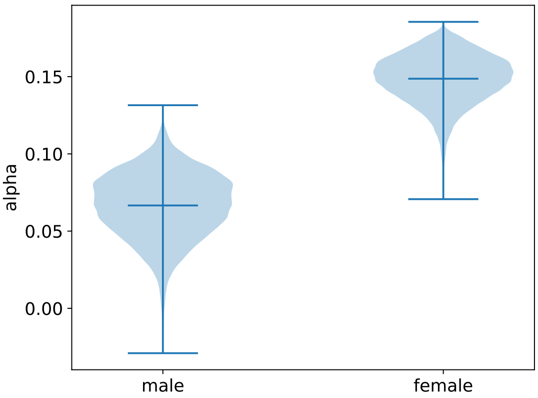

Figure 4: Use of alpha per gender on the test test.

6. Experiment: Emotion Adaptation

This work 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. Our hypothesis is that if it can, we expect it to improve the generalisability of emotional TTS models when trained on limited emotional data and thus improve audio quality and expressiveness.

6.1. Database 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. We do not use the visual information of the database. In SAVEE 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 5.1). We work with the same input and output features as described in section 5.2, but we also compute dynamic features. For mean/variance normalisation the parameters of the WSJCAM0 database are used, which facilitates transfer learning described below.

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)[2] to extract the first and second formant (F1 and F2) for vowel phonemes for the six different emotion (Figure 5). 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. We see the same F1 shift for angry speech as reported in Vlasenko et al. (2011).

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

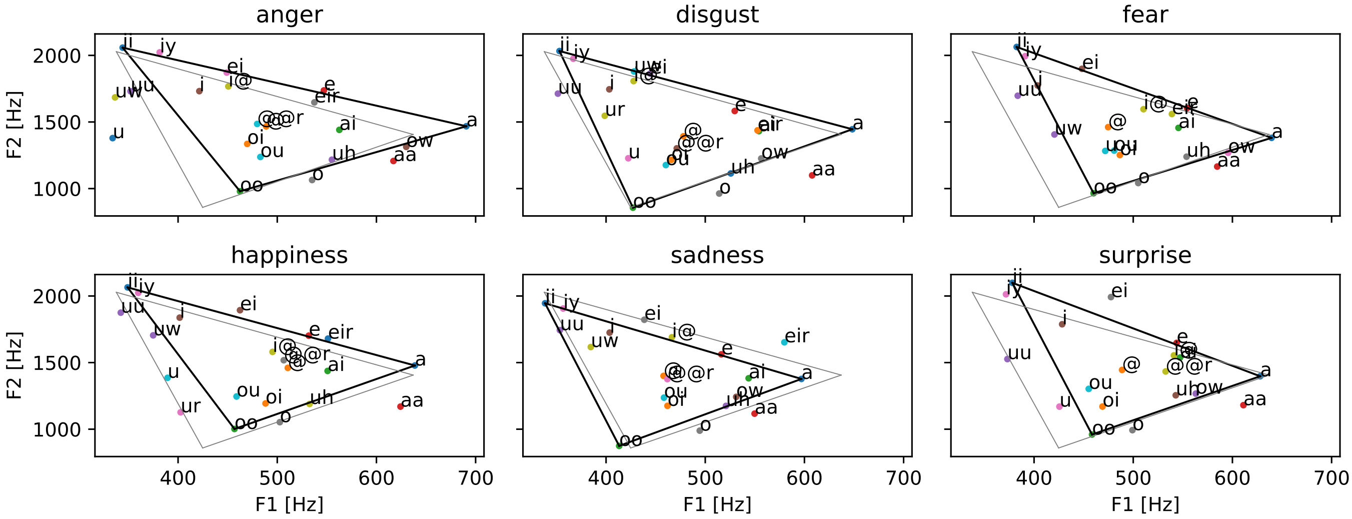

Figure 5: Analysis of the first and second formant frequency in Hz of vowel phonemes for the six different emotions. Light grey corresponds to the vowel triangle of neutral speech.

These observations show that emotions also caused a formant shift in English and that the SAVEE database is a suitable choice for our experiment.

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 are investigating 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) has compared supervised training of the baseline system with 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 Mean-Opinion-Score (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.

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 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 an LSTM. The output of the LSTM is 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 LF0 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.

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 5.1) 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

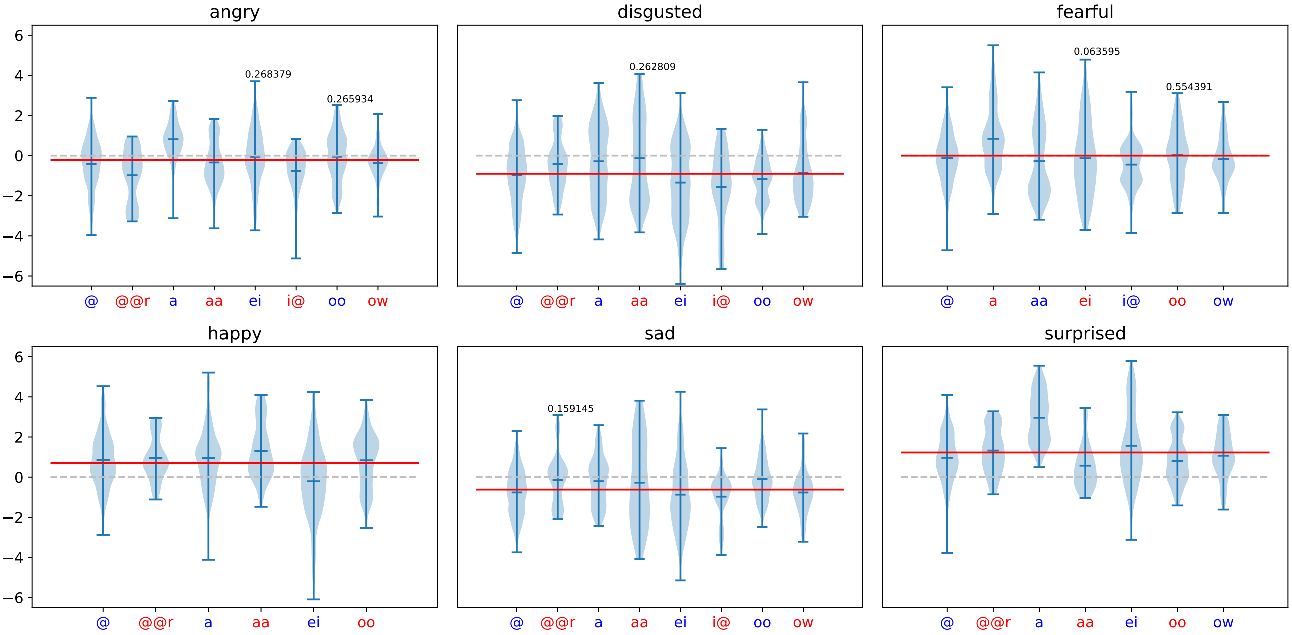

Figure 6: Normalised warping of the eight phonemes, which are most affected by valence, for the different emotions. The warping is mean/variance normalised with the warping on neutral speech to remove the positioning effect described above. The red line indicates the average warping over all eight phonemes. p-values of a two-sided t-test are displayed when p > 0.05.

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.

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 have a hypothesis that our model would outperform other techniques in adaptation performance. Instead, we attempt to understand what the transform has and has not learned in order to charactersise the technique and to know where to direct future research.

6.5. Statistic 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 7). The warping on neutral samples seems to position the phonemes within the vowel triangle (compare Figure 5). 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.

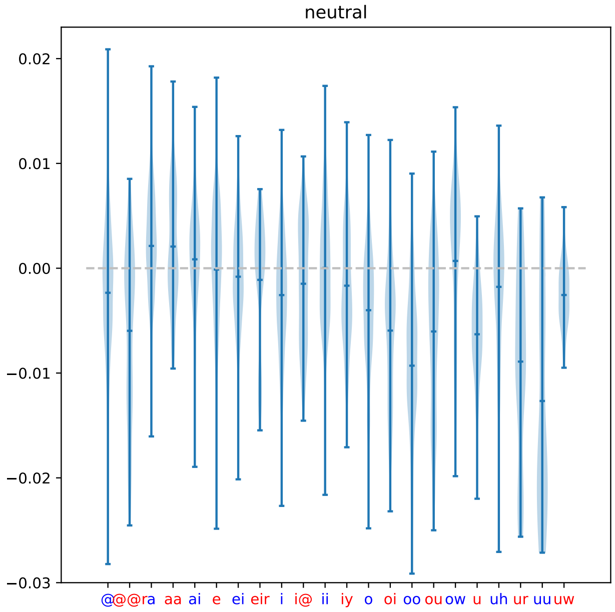

Figure 7: Warping per vowel phoneme for neutral speech on SAVEE.

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, 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 8). 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.

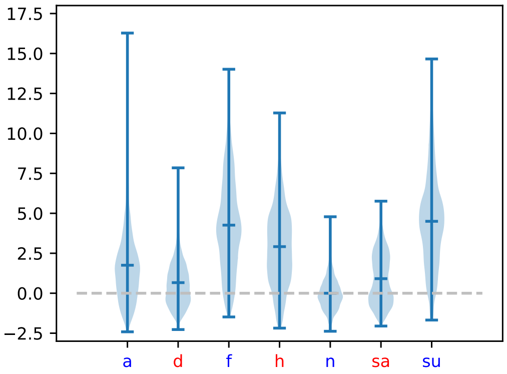

Figure 8: F0 statistics mean/variance normalised w.r.t. neutral for the six emotions (a: anger, d: disgust, f: fear, h: happiness, n: neutral, sa: sadness, su: surprise). High arousal emotions (anger, fear, happiness, surprise) show higher F0 mean, variance, and range.

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 9 left). We then compute for which phonemes the difference in mean F1 is the most between low and high valence emotions (Figure 9 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.

Indeed, we see that the average warping of all eight phonemes (red line in Figure 6) 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.

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

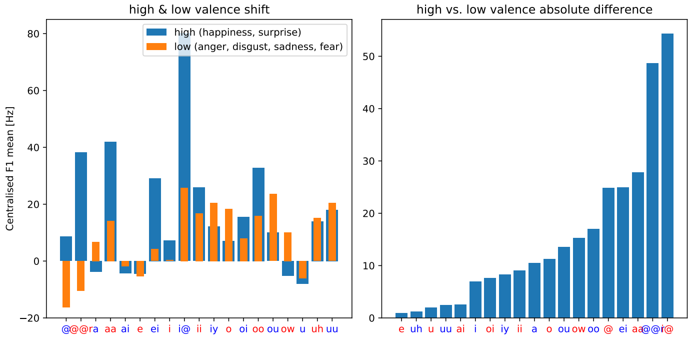

Figure 9: Left: Average F1 shift between low/high valence emotions compared to neutral speech. Right: Phonemes ordered by absolute difference of the F1 shift between low and high valence emotions.

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.

7. Conclusion

We set out to characterise a neural all pass warp (APW). We made two hypotheses for that matter: 1) 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. 2) 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; Schnell et al., 2021). On the quality side we intend 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.

Aggarwal, V., Cotescu, M., Prateek, N., Lorenzo-Trueba, J., 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). IEEE, pp. 6179–6183.

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

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

Banse, R., 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., 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., Bagby, T., 2019. Effective use of variational embedding capacity in expressive end-to-end speech synthesis. arXiv preprint arXiv:1906.03402.

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

Black, A., Taylor, P., Caley, R., Clark, R., 1998. The festival speech synthesis system.
URL `http://www.cstr.ed.ac.uk/projects/festival/`

Boersma, P., Weenink, D., 2017. Praat: doing phonetics by computer [computer program].
URL `http://www.praat.org/`

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

Cauchi, B., Kodrasi, I., Rehr, R., Gerlach, S., Jukić, A., Gerkmann, T., Doclo, S., 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.

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.

Choi, S., Han, S., Kim, D., Ha, S., 2020. Attentron: Few-Shot Text-to-Speech Utilizing Attention-Based Variable-Length Embedding. In: Proc. Interspeech 2020. pp. 2007–2011.
URL `http://dx.doi.org/10.21437/Interspeech.2020-2096`

Cohen, J., Kamm, T., 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.

Cooper, E., Lai, C.-I., Yasuda, Y., Fang, F., Wang, X., Chen, N., 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). IEEE, pp. 6184–6188.

Eichner, M., Wolff, M., Hoffmann, R., 2004. Voice characteristics conversion for TTS using reverse VTLN. In: 2004 IEEE International Conference on Acoustics, Speech, and Signal Processing. Vol. 1. IEEE, pp. I–17.

Fransen, J., Pye, D., Robinson, T., Woodland, P., 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.

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

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

Gururani, S., Gupta, K., Shah, D., Shakeri, Z., 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., Edge, J., 2008. Audio-visual feature selection and reduction for emotion classification. In: Proc. Int. Conf. on Auditory-Visual Speech Processing (AVSP08), Tangalooma, Australia.

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

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

Jaitly, N., Hinton, G. E., 2013. Vocal tract length perturbation (VTLP) improves speech recognition. In: Proc. ICML Workshop on Deep Learning for Audio, Speech and Language. Vol. 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. pp. 4480–4490.

Johnstone, T., Scherer, K. R., 2000. Vocal communication of emotion. Handbook of emotions 2, 220–235.

Karlapati, S., Moinet, A., Joly, A., Klimkov, V., Sez-Trigueros, D., Drugman, T., 2020. CopyCat: Many-to-Many Fine-Grained Prosody Transfer for Neural Text-to-Speech. In: Proc. Interspeech 2020. pp. 4387–4391.
URL http://dx.doi.org/10.21437/Interspeech.2020-1251

Kingma, D. P., Welling, M., 2014. Auto-encoding variational bayes.
URL http://arxiv.org/abs/1312.6114

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

Kotani, G., Saito, D., 09 2019. Voice conversion based on full-covariance mixture density networks for time-variant linear transformations. pp. 75–80.

Kotani, G., Saito, D., 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). IEEE, pp. 1259–1262.

Lee, Y., 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). IEEE, pp. 5911–5915.

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

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

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

Morise, M., Yokomori, F., 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., Wolf, L., 2018. Fitting new speakers based on a short untranscribed sample. In: International Conference on Machine Learning. pp. 3683–3691.

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

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

Qian, K., Zhang, Y., Chang, S., Yang, X., Hasegawa-Johnson, M., 09–15 Jun 2019. AutoVC: Zero-shot voice style transfer with only autoencoder loss. Vol. 97 of Proceedings of Machine Learning Research. PMLR, Long Beach, California, USA, pp. 5210–5219.
URL http://proceedings.mlr.press/v97/qian19c.html

Saheer, L., Dines, J., 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., Liang, H., September 2010. Implementation of VTLN for statistical speech synthesis. In: SSW7. Kyoto, Japan.

Schnell, B., Garner, P. N., 2019. Neural VTLN for speaker adaptation in TTS. In: Proc. 10th ISCA Speech Synthesis Workshop. pp. 29–34.

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

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

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). IEEE, pp. 4779–4783.

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

Sundermann, D., 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). IEEE, pp. 556–559.

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

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

Um, S.-Y., Oh, S., Byun, K., Jang, I., Ahn, C., 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). IEEE, pp. 7254–7258.

van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., Kavukcuoglu, K., 2016. Wavenet: A generative model for raw audio. In: 9th ISCA Speech Synthesis Workshop. pp. 125–125.

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

Vlasenko, B., Prylipko, D., Philippou-Hübner, D., 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, 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., Saurous, R. A., 2017. 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. ISCA, pp. 4006–

4010.
URL http://www.isca-speech.org/archive/Interspeech_2017/abstracts/1452.html

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

Watts, O., Henter, G. E., Fong, J., 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).

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

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

Yasuda, Y., Wang, X., Takaki, S., 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). IEEE, pp. 6905–6909.

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

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

Zhang, Y.-J., Pan, S., He, L., Ling, Z.-H., 2019. 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). IEEE, pp. 6945–6949.

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