AI-SYNTHESIZED SPEECH

GENERATION AND DETECTION

by

Ehab Alsayed Albadawy Abdrabuh

A Dissertation

Submitted to the University at Albany, State University of New York

in Partial Fulfillment of

the Requirements for the Degree of

Doctor of Philosophy

College of Engineering and Applied Sciences

Department of Electrical and Computer Engineering

Spring 2022

Dedicated to my parents, sisters, and brothers for their continuous love and support.

ABSTRACT

From speech to images, and videos, advances in machine learning have led to dramatic improvements in the quality and realism of so-called AI-synthesized content. While there are many exciting and interesting applications, this type of content can also be used to create convincing and dangerous fakes. We seek to develop forensic techniques that can distinguish a real human voice from a synthesized voice. We observe that deep neural networks used to synthesize speech introduce specific and unusual artifacts not typically found in human speech. Although not necessarily audible, we develop various detection algorithms to measure these artifacts and be able to differentiate between human and synthesized speech.

ACKNOWLEDGMENT

I would like to thank my advisors Prof. Ming-Ching Chang and Prof. Siwei Lyu for their help and continuous support. I am grateful for the opportunity they gave me to pursue my graduate studies. I have been through critical times through my Ph.D., without the patients and guidance from my advisors, my dissertation would not have been possible. I would also like to thank my graduate committee members, Prof. Daphney-Stavroula Zois and Prof. Hany Elgala for their insightful comments and encouragement. I would like to extend my gratitude to Prof. Yelin Kim and Prof. Hany Farid for their great mentoring and kind support.

My sincere thanks to Prof. Maciej Mazurowski for giving me the chance to work on my first real research problem. With his guidance, I learned to think critically and find solutions independently. I had a great time working in his lab where I got to work with amazing colleagues, Ashirbani Saha and Mateusz Buda.

I would also like to thank my wonderful teachers and mentors back in Egypt. I have been fortunate enough to meet such great minds in my early career life.

My sincere thanks also go to my wonderful colleagues and friends. In particular, I would like to thank:

My friends and colleagues at University at Albany: Yuezun Li, Lipeng Ke, Angeliki Kapodistria, Miley Yao, and Yasitha Warahena Liyanage.

My mentors and colleagues at Meta AI: Xin Lei, Andrew Gibiansky, Qing He, and Jilong Wu.

My friends: Abduallah Mohamed, Ali Mohamed, and Alaaeldin El-Nouby.

Finally, and most immortally, I would like to thank my family for their endless love and support.

CONTENTS

LIST OF FIGURES

CHAPTER 1

Introduction

The advancements of AI-generated audio often referred to as “deepfake audio” have introduced a growing threat of impersonation and disinformation. It is therefore of practical importance to develop detection methods for deepfake audios. We first start to explore the different techniques of Deep learning method to be used on speech signals. More specifically, we work on the emotion recognition problem where we use the state-of-the-art deep BLSTM network on the RECOLA dataset. Our results demonstrate three contributions: (i) the joint representation outperforms both individual representation baselines and the state-of-the-art speech-based results on RECOLA, validating the assumption that combining continuous and discretized emotion representations yields better performance in emotion prediction; and (ii) the joint representation can help to accelerate convergence, particularly for valence prediction. Our work provides insights into joint discrete and continuous emotion representation and its efficacy for describing dynamically changing affective behavior in valence and activation prediction.

We explore the generation process of AI-synthesized speech to build a generalizable classifier for synthesized speech. We consider building our own dataset using state-of-the art algorithms for speech synthesis. Additionally, we introduce our own deep learning-based approach for voice conversion with speech style transfer across different speakers. In our work, we use a combination of Variational Auto-Encoder (VAE) and Generative Adversarial Network (GAN) as the main components of our proposed model followed by a WaveNet-based vocoder. We develop three objective metrics to evaluate our model using the ASVspoof 2019 dataset for three tasks: 1) measuring the difficulty of differentiating between human and synthesized samples, 2) content verification for transcription accuracy, and 3) speaker encoding for identity verification.

Secondly, we conduct a deeper analysis on neural vocdoers. Neural vocoders, used for converting the spectral representations of an audio signal to the waveforms, are a commonly used component in speech synthesis pipelines. It focuses on synthesizing waveforms from low-dimensional representation, such as Mel-Spectrograms. In recent years, different approaches have been introduced to develop such vocoders. However, it becomes more challenging to assess

these new vocoders and compare their performance to previous ones. To address this problem, we present VocBench, a framework that benchmark the performance of state-of-the-art neural vocoders. VocBench uses a systematic study to evaluate different neural vocoders in a shared environment that enables a fair comparison between them. In our experiments, we use the same setup for datasets, training pipeline, and evaluation metrics for all neural vocoders. We perform a subjective and objective evaluation to compare the performance of each vocoder along a different axis. Our results demonstrate that the framework can show competitive efficacy and quality of the synthesized samples for each vocoder. VocBench framework is available at:
https://github.com/facebookresearch/vocoder-benchmark.

Thirdly, we have developed a forensic technique that can distinguish human from synthesized speech. This technique is based on the observation that current speech-synthesis algorithms introduce specific and unusual higher-order bispectral correlations that are not typically found in human speech. We have provided preliminary evidence that these correlations are the result of the long-range correlations introduced by the underlying network architectures used to synthesize speech. This bodes well for us in the forensic community as it appears that these network architectures are also what is giving rise to more realistic sounding speech (despite the unusual bispectral correlations). More work, however, remains to be done to more precisely understand the specific source of the unusual bispectral correlations.

Lastly, for a more sophisticated analysis of high level artifacts on the synthesised speech. We build a large-scale dataset named LibriVoc that includes samples generated by six different vocoder systems. Experimental results show that this approach is effective in vocoder identification. We further introduce a new method to detect deepfake audios by identifying the neural vocoders that are widely used in the generation process. The neural vocoder is a core component in most deepfake audio synthesis models, hence the identification of neural vocoder processing implies that an audio sample may be synthesized. We develop a vocoder identification method based on the RawNet2 model.

CHAPTER 2

Application of Deep Learning on Speech: Joint Discrete and Continuous Emotion Prediction

2.1 Introduction

Emotion recognition has gained a great interest in the multimodal interaction community [12, 64, 126]. Audio-visual expressive behavior includes salient information to understand the overall tone and affective context of interaction [89, 68, 10]. Continuous emotion prediction, a process of estimating emotion in continuous time, uses dimensional representation of emotion (e.g., arousal and valence) [31, 83]. The continuous-time prediction allows us to better understand the dynamic behavior of affective interaction [71] and generate more natural and timely responses in interactive systems [31, 93, 94]; however, the main challenge is that continuous emotion labels often contain inherent noise that result in negative consequences in training and prediction. In this paper, we explore the benefit of jointly modeling continuous and discrete emotion representation and investigate the optimal trade-off between the two representations.

Previous studies on continuous emotion prediction either focused on (i) regression approaches that directly used continuous emotion representation (e.g., valence: 0.386) [22, 71, 100, 124, 16, 77, 96, 39, 59, 67] or (ii) classification approaches that used discretized representation that quantized continuous labels into a discrete set of categories (e.g., valence: ”high” category) [123, 137, 125, 131, 70, 136]. The regression approaches retain the full label information during training; however, inherent noise in emotion labels [32, 135] may present negative effects in training, such as decreased accuracy or increased model complexity [21]. In contrast, the classification approaches attempted to reduce label noise using various quantization methods, such as Affinity Propagation-based clustering [137], binarization based on the mean (low and high classes) [70], label modeling over grid cells [131, 136], and k-means clustering [58]. These approaches has shown to be effective in emotion prediction at the expense of a label quantization error [58, 22]. The previous studies demonstrate the pros and cons of regression and classification approaches; however, an open question remains how we can find the optimal trade-off between the two approaches, and whether we can jointly use continuous and discrete emotion representation.

This gap led us to explore new joint modeling methods for combining the regression and classification approaches: *ensemble* and *end-to-end* models. Each model combines the two tasks at the prediction (decision) and representation levels, respectively. First, the ensemble model combines the predictions from two separate models that are optimized independently for regression and classification tasks. This model can make the optimal prediction for each task, and previous research has shown the efficacy of ensemble approaches in multiple domains, such as speech recognition [17], image recognition [34], and emotion recognition [75, 95]. Next, the proposed end-to-end model is trained to simultaneously optimize regression, classification, and the final combination between these two, in an end-to-end manner. The advantage of the end-to-end model is that it does not require any manually designed intermediary algorithms (e.g. averaging in ensemble models). For the end-to-end model, we further investigate how the classification and regression losses should be combined by introducing a new total loss function. We also explore the benefit of our proposed joint representation compared to a fully-connected neural network that does not exploit this representation.

We use the benchmark Remote Collaborative and Affective Interactions (RECOLA) dataset [89] and Concordance Correlation Coefficient (CCC) performance to examine the proposed methods in the state-of-the-art context. We build strong baselines using deep bidirectional long short-term memory (BLSTM) for individual classification and regression tasks. Our two proposed models use deep BLSTMs that combine these two tasks in ensemble and end-to-end approaches. Since the previous state-of-the-art work that used a discretization method on RECOLA focused on speech emotion recognition [58], we first use speech to explore different joint modeling methods. We then investigate how our proposed joint modeling work in audio-visual environment.

Our experimental results suggest that joint modeling of discrete and continuous emotion labels results in more accurate emotion prediction. Our work can increase the understanding of the representation of multimodal interaction in continuous time. In summary, the main novelty of this work includes: (i) the design and development of new joint modeling methods using deep BLSTMs, (ii) the new insight into a high-level joint representation of discrete and continuous emotion, and (iii) the investigation of the trade-off between classification and regression tasks for emotion prediction.

2.2 Background

2.2.1 Continuous emotion prediction

Previous research has demonstrated that continuous emotion predictions systems can model the dynamics of the emotion fluctuation.

Khorram et al. [49] used two Convolutional Neural Network (CNN) architectures to capture long-term temporal dependencies in speech emotion. The first architecture uses a stack of dilated convolutions that has been shown to improve image segmentation, speech synthesis, and automatic speech recognition. To handle unstable predictions of the first architecture, they proposed a downsampling/upsampling network that downsamples the input signal and upsamples the generated predictions. The results showed that this network can not only improve the emotion prediction, but also generate smoothed predictions. Soleymani et al. [99] used linear ridge regression algorithm to map the extracted features to the given rating for each video clip. They used arousal, valence, dominance, and liking ratings. For their features, they used recorded physiological responses of 32 participants while watching collected emotional music videos. To validate their results, they used leave-one-out cross-validation strategy for each participant. They got significant improvement for the continuous emotion detections in comparison to random estimation. Tzirakis et al. [108] used a CNN-Recurrent Neural Network (RNN) model that operates on the raw signal of speech and visual data to get the continuous emotion predictions. To handle the speech and visual data, they used two type of different models. The first model is Visual Network, where they used a deep residual network (ResNet) of 50 layers [34]. They used the pixel intensities from the cropped faces of the subject's video as input to ResNet. The second model is Speech Network, where they used 2-layers CNN model on the raw audio signal. Both models are followed by Long Short-Term Memory (LSTM) layer to handle the temporal dependency. The results showed that their proposed models perform significantly better in the test set in comparison to other models using RECOLA database. The winning submission to the AVEC 2016 challenge, Brady et al. [9] used a multi-modal system for the continuous predictions. For the audio sequence, they used sparse coding to learn the higher representation of the extracted features. They used Mel-frequency cepstral coefficients (MFCC), shifted delta cepstral (SDC), and prosody features. They got a significant improvement in arousal compared to the baseline scores. For the Video sequence, they used CNN with 3 convolution layers on the detected face in each frame. Additionally, they proposed a fusion approach that enabled the multi-sensor fusion of emotional state while maintaining the emotional states variation over time. However, these works didn't take into consideration the nosiness in the ground truth while tackling the continuous emotion prediction task as a regression problem.

Recently, Le et al. [58] proposed a novel discretization method of continuous emotion labels. This method can do the label discretization by using K-means clustering to divide the output range into discretized regions. Choosing the number of the clusters for the k-means model is very crucial, as it affects both how precise the conversion will be from the discretized signal to the original one, and the difficulty level for the network with the target labels. In order to achieve the balance between both the precision and difficulty level, they used 4 different k-means models with 4, 6, 8 and 10 cluster numbers. To have more robust prediction, they introduced a decoding framework of a Hidden Markov Model (HMM)-based language model. The results showed that discretization-based classification outperforms a traditional regression model, and achieved the discretization approach achieved the state-of-the-art performance in RECOLA dataset. This work motivated our investigation of joint modeling approach; however, our work differs from [58] in that we focus on combining classification and regression tasks to simultaneously reduce the label noise and regress its prediction. We also provide new insights into how the two tasks contribute differently to final emotion prediction.

2.2.2 Bidirectional LSTM

Previous research has demonstrated the efficacy of bidirectional Long Short-Term Memory (BLSTM) models in emotion prediction. For example, Metallinou et al. [71] proposed a hierarchical approach using BLSTM and HMM classifiers to model emotion state for each utterance, and demonstrated the effiacy of using a hybrid HMM/BLSTM model. He et al. [35] used deep BLSTM based fusion architecture between different modalities (e.g., audio and video) for continuous emotions prediction. They smoothed the initial predictions from a unimodal DBLSTM with Gaussian smoothing, and input these predictions into a second layer of DBLSTM for the final prediction.

BLSTM is a standard LSTM cell that process the input sequence in both the forward and the backward directions. An LSTM cell [38] has internal cell state (c_τ) at time τ that is computed based on the current input (x_τ) and the previous cell state ($c_{\tau-1}$). Combination of input (i_τ) and forget (f_τ) gates determines how much the previous cell state $c_{\tau-1}$ and the current input x_τ contribute into the current cell state c_τ. The activation function for both forget f_τ in input i_τ gates is sigmoid

σ that outputs values between 0 and 1. Specifically, the current cell state c_τ is computed as follows:

$$i_\tau = \sigma(W_{xi}x_\tau + W_{hi}h_{\tau-1} + W_{ci}c_{\tau-1} + b_i) \tag{2.1}$$

$$f_\tau = \sigma(W_{xf}x_\tau + W_{hf}h_{\tau-1} + W_{cf}c_{\tau-1} + b_f) \tag{2.2}$$

$$\tilde{c}_\tau = tanh(W_{xc}x_\tau + W_{hc}h_{\tau-1} + b_c) \tag{2.3}$$

$$c_\tau = f_\tau * c_{\tau-1} + i_\tau * \tilde{c}_\tau \tag{2.4}$$

The current cell output for time τ is computed based on the current cell state c_τ:

$$o_\tau = \sigma(W_{xo}x_\tau + W_{ho}h_{\tau-1} + W_{co}c_{\tau-1} + b_o) \tag{2.5}$$

$$h_\tau = o_\tau * tanh(c_\tau) \tag{2.6}$$

where o_τ is the output gate that determines the current cell state c_τ contribution to the current output h_τ. We can also re-write h_τ and c_τ as follows:

$$(h_\tau, c_\tau) = \mathcal{F}(x_\tau, h_{\tau-1}, c_{\tau-1})$$

where $\mathcal{F}$ is the LSTM activation function. For BLSTM, h_τ will be a composite of the forward and backward directions $h_\tau = [\overrightarrow{h_\tau}; \overleftarrow{h_\tau}]$ and it is defined as follows:

$$(\overrightarrow{h}_\tau, \overrightarrow{c}_\tau) = \overrightarrow{\mathcal{F}}(x_\tau, \overrightarrow{h}_{\tau-1}, \overrightarrow{c}_{\tau-1}) \tag{2.7}$$

$$(\overleftarrow{h}_\tau, \overleftarrow{c}_\tau) = \overleftarrow{\mathcal{F}}(x_\tau, \overleftarrow{h}_{\tau-1}, \overleftarrow{c}_{\tau-1}) \tag{2.8}$$

2.3 Audio-Visual Data and Features

In this work, we use the RECOLA dataset [89]. This dataset has been used as a benchmark dataset in continuous emotion prediction [12, 107]. Particularly, the dataset has been used in the Audio/Visual Emotion Challenge (AVEC) [111] and had resulted in numerous advancements in emotion recognition [58, 49, 9, 66, 48]. Hence, this dataset allows us to evaluate and compare our proposed methods in the state-of-the-art context.

RECOLA contains spontaneous and naturalistic interactions of 27 French-speaking subjects, each with five-minute recordings. All the subjects were recorded in dyads during a video conference while completing a task requiring collaboration. The corpus contains multimodal cues, including audio, video, electrocardiogram (ECG), and electrodermal activity (EDA). The dataset is divided into train, development and test sets, where 9 different subjects present in each set. Ground truth emotion labels, arousal and valence, are obtained from six gender-balanced French-speaking annotators as continuous-time ratings (40 ms binned frames).

In this paper, we use Kaldi toolkit [84] to extract 40-dimensional log Mel filter bank coefficient with 25ms window size and 10ms frame shift to be consistent with previous work [58, 49] To have the same number of input and output frames, we concatenate every four consecutive frames of the input, resulting in 160 feature vector for each frame.

To explore joint modeling of emotion representation in a multimodal environment, we also use visual appearance and geometric features provided in AVEC 2015 that used the RECOLA dataset [88]. The use of the challenge baseline features allows us to directly compare our modeling with baselines and foster reproducibility of our work. The appearance features were calculated using Local Gabor Binary Patterns from Three Orthogonal Planes for each frame. The dimensionality of the features was reduced to 84 after Principal Component Analysis-based feature selection. The geometric features were computed using the Euclidean distances between 49 facial landmarks [88], resulting in 316 features. The missing frames of these features were interpolated. In total, the dimensionality of visual features is 400.

We perform z-normalization per each session on both audio and visual features.

Evaluation Strategy. To be consistent with previous work using RECOLA [9, 58, 111], we use Concordance Correlation Coefficient (CCC) [57] which is defined by:

$$CCC = \frac{2 * cov(y, \hat{y})}{var(y) + var(\hat{y}) + \left(\mathbb{E}[y] - \mathbb{E}[\hat{y}]\right)^2} \tag{2.9}$$

where $cov(y, \hat{y})$ is the covariance matrix, $var(y)$ is the variance, and $\mathbb{E}[y]$ is the expected value for every given ground truth label $y = [y_1, y_2, y_3, ..., y_n]$ and predicted label $\hat{y} = [\hat{y}_1, \hat{y}_2, \hat{y}_3, ..., \hat{y}_n]$, where n is total number of frames that are evaluated.

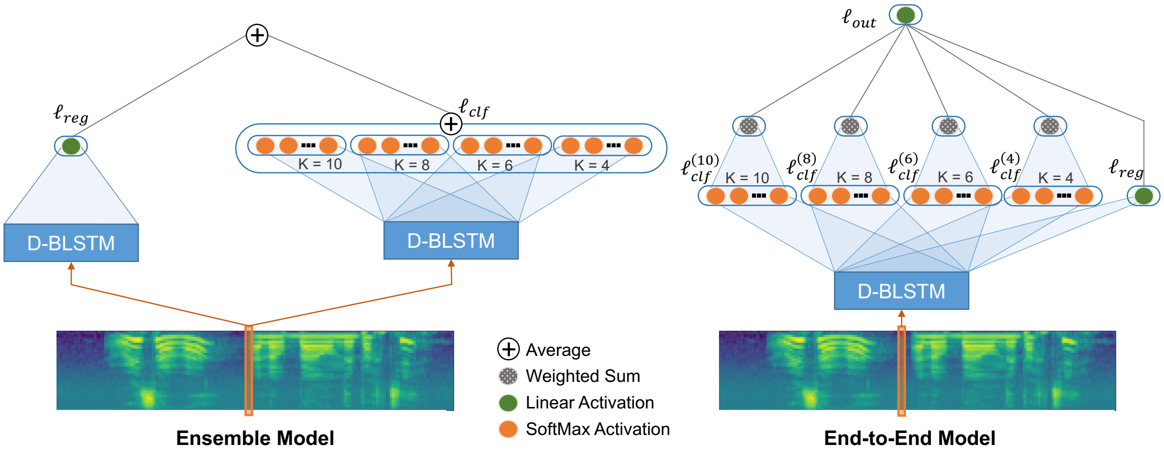


Figure 2.1: Our proposed joint modeling approaches using deep BLSTM ('D-BLSTM') architecture for ensemble (left) and end-to-end (right) models. The ensemble combines the classification and regression predictions at the decision level, whereas the end-to-end model combines the classification (ℓ_{clf}), regression (ℓ_{reg}), and final combination of the two tasks (ℓ_{out}) at the representation level (using the total loss function, ℓ_{total}, in Eq. 2.13). The figure is best shown in color.

2.4 Proposed Methods

In this section, we first introduce our individual baselines that use the state-of-the-art deep BLSTM architecture for classification (*clf*) and regression (*reg*) tasks (Section 2.4.1). We then describe our proposed joint modeling methods: ensemble and end-to-end (Section 2.4.2). For the end-to-end model, we introduce a new loss that allows us to investigate the contribution of both *clf* and *reg* tasks. Finally, we explore the special case of the end-to-end model when the contributions of *clf* and *reg* are ignored. To this end, we design a BLSTM with fully connected (FC) layer and compare its performance with the end-to-end model, and explore if this new model can be used in our joint modeling framework (Section 2.4.3).

2.4.1 Individual classification (*clf*) and regression (*reg*) models

We build strong individual modeling baselines using deep BLSTM for the *clf* and *reg* models similar to Le et al. [58]. We introduce a new cost function, called *norm cost*, that can provides smoother predictions for discretized emotion labels in the loss computation.

For *clf*, we use deep BLSTM network with four k-means models as in Le et al. [58] (details in Section 2.2.1). Each k-means model is considered as a target task [58]. For this model, Le et al. [58] introduced a cost-sensitive cross entropy loss (CCE). We use similar loss function with the new cost function C_{norm}, and the loss function is defined as follows:

$$\ell_{clf} = \sum_{t=1}^{4} \frac{1}{F} \sum_{f=1}^{F} C_{norm}(y_{tf}, \hat{y}_{tf}) \sum_{l=1}^{L_t} y_{tf}^{(l)} . \log \hat{y}_{tf}^{(l)} \tag{2.10}$$

where F is the number of frames and L_t is the number of labels in the task t. $y_{tf}^{(l)}$ and $\hat{y}_{tf}^{(l)}$ are the one-hot encoded ground truth and the predicted probability respectively at label index l for frame f and target function t.

Our proposed norm cost function C_{norm} is defined as follows:

$$C_{norm}(y_{tf}, \hat{y}_{tf}) = 1 + \left\| \sum_{l=1}^{L_t} K_{L_t}^{(l)} (y_{tf}^{(l)} - \hat{y}_{tf}^{(l)}) \right\|_2 \tag{2.11}$$

where $K_{L_t}^{(l)}$ is the centroid of the label l for the k-mean model that has L_t labels. For example is K_{L_t} can be $[-0.31, -0.12, 0.05, 0.22]$ for $L_t = 4$, $\hat{y}_{tf}$ can be $[0.1, 0.6, 0.2, .1]$, and y_{tf} can be $[0, 0, 1, 0]$. The cost function C_{norm} takes into consideration the spatial relation between the labels, as it helps to have more stable training [58] where its value is 1 for normal cross entropy calculation. The new norm cost function C_{norm} takes the l_2 norm distance between the actual and the predicted centroids. This allows us to take into account all the predicted probabilities for each time step to get the final predicted value. However, Le et al. [58] used argmax-based cost function C that calculates the final predicted value based on the highest probability for each task, which may ignore the full label distribution. For example, if we use the norm cost, the two different softmax predictions of $[0, 0.3, 0.6, 0]$ and $[0, 0, 0.6, 0.3]$ will result in different final predicted values, $0.3 \times -0.12 + 0.6 \times 0.05 = -0.006$ and $0.6 \times 0.05 + 0.3 \times 0.22 = 0.096$, respectively. However, if we use the argmax-based cost C, both of the predictions will have the same final predicted value of $1 \times 0.05 = 0.05$, and the subtle difference between the two predictions will be ignored. Hence, the norm cost C_{norm} is more sensitive to inherent label distribution than the argmax-based cost C.

For *reg*, we use a deep BLSTM network trained with CCC loss function, $CCC_{loss} = 1 - CCC$, where CCC is defined in Eq. 2.9. Previous studies have shown that using CCC as the loss function enhance the continuous emotion predictions compared to RMSE loss [107, 85, 58]. The CCC loss has the advantage over RMSE that it takes into consideration the overall shape of the time series.

2.4.2 Joint Modeling: ensemble and end-to-end

As shown in Fig. 1, we explore two different joint modeling of *clf* and *reg* models: ensemble and end-to-end. These approaches differ from previous methods that formulated continuous emotion prediction as either an individual classification or a regression task. We hypothesize that joint modeling of the two tasks will find the optimal trade-offs between the easiness and precision of the training and improve the overall prediction power of the model. We use the deep BLSTM model for both classification and regression tasks.

The ensemble model combines two BLSTM models. The first model is trained for the classification task, where it has four target tasks for each k-means model as shown in Fig 2.1. To get the predicted centroid value for each task, we take the highest probability in the softmax layer outputs. Using the example in Section 2.4.1, where $K_{L_t} = [-0.31, -0.12, 0.05, 0.22]$ for $L_t = 4$, and $\hat{y}_{tf} = [0.1, 0.6, 0.2, .1]$, the highest probability in $\hat{y}_{tf}$ is 0.6 for the second centroid; hence the predicted centroid for this case will be -0.12. The final predicted value will be the average of the four predicted centroids form each task. The second model is trained for regression task as mentioned in Section 2.4.1. The final predictions are generated by averaging both the classification and regression models' predictions. For the ensemble model, we use CCE loss ℓ_{clf} defined in Eq. 2.10 for classification, and the CCC loss defined in Section 2.4.1 for regression.

The end-to-end model is a single deep BLSTM model that is trained end-to-end with these two target tasks as shown in Fig 2.1. In addition to the 4 classification nodes, we add a regression node with linear activation that shares the same BLSTM output with the classification task. We apply the weighted sum operation to get the predicted value for each classification task, where we multiply each probability by its corresponding centroid value for each task. Using the previous example for $L_t = 4$, $K_{L_t} = [-0.31, -0.12, 0.05, 0.22]$, $\hat{y}_{tf} = [0.1, 0.6, 0.2, .1]$, the predicted value for this case will be $-0.31 \times 0.1 - 0.12 \times 0.6 + 0.05 \times 0.2 + 0.22 \times 0.1 = -0.071$. We then add a final node to combine the predicted values for the five nodes (4 classification nodes, and one regression node) with learnable weights.

We train the end-to-end model that minimizes the total loss ℓ_{total}, which combines the three different losses. We use the same loss function ℓ_{clf} (Eq. 2.10) for classification, $RMSE_{loss}$ defined below for regression (ℓ_{reg}), and the CCC loss for the final node (ℓ_{out}) defined in Sec 2.4.1. The loss functions of the regression and final nodes are chosen empirically. The $RMSE_{loss}$ and total losses are defined as follows:

$$RMSE_{loss} = \sqrt{\frac{1}{F}\sum_{f=1}^{F}(l_f - \hat{l}_f)^2} \tag{2.12}$$

$$\ell_{total} = \alpha_1\ell_{clf} + \alpha_2\ell_{reg} + \alpha_3\ell_{out} \tag{2.13}$$

$$\alpha_3 = 2 - \frac{1}{2}(\alpha_1 + \alpha_2) \tag{2.14}$$

In Eq. 2.13, α_1 and α_2 are the hyper-parameters that take values between 0 and 1, and they control the contribution of ℓ_{clf} and ℓ_{reg} losses in the total loss ℓ_{total}. Eq. 2.14 for α_3 is designed so that we can enforce a loss value for the final node to be always equal to or greater than α_1 and α_2 and take a value between 1 and 2. For example, when $\alpha_1 = \alpha_2 = 1$, this means both tasks have the same weight of contribution in the final loss and α_3 will be 1 as well. Similarly, $\alpha_1 = \alpha_2 = 0$ means that ℓ_{clf} and ℓ_{reg} losses are completely ignored and the model will try to come up with its own representation for these tasks. α_3 will be 2 in this case.

Table 2.1: CCC differences between previous state-of-the-art work, baselines (*clf* and *reg*), and our proposed two joint models for speech-based experiments. Dev: development set, Test: test set.

Model	**Arousal**		**Valence**	
	dev	test	dev	test
Valstar et al. [111]	0.796	0.648	0.455	0.375
Brady et al. [9]	0.846	-	0.450	-
Le et al. [58] (CLS-Raw)	0.858	0.682	0.563	0.448
Le et al. [58] (CLS-Decoded)	0.859	0.680	0.596	0.460
Khorram et al. [49]	0.867	0.684	0.592	0.502
clf [baseline]	0.860	0.686	0.558	0.527
reg [baseline]	0.863	0.686	0.595	0.544
clf+*reg* (ensemble)	**0.868**	0.693	0.601	**0.555**
clf+*reg* (end-to-end)	**0.868**	**0.697**	**0.623**	0.530

2.4.3 BLSTM-FC model

We further explore the special case of end-to-end model when the *clf* and *reg* tasks are ignored and the representation of this layer is learned in an unsupervised manner. The end-to-end model with $\alpha_1 = \alpha_2 = 0$ is equivalent to a model with a special type of hidden layer that has softmax and linear activated nodes. Since softmax activation restricts the hidden layer outputs to have a probability distribution that sums up to one, this can increase the model complexity. To address this complexity issue, we replace the mid-layer with a fully connected layer (FC) that has 16 units. We call this model BLSTM-FC. The FC and the final node are calculated as follows:

$$FC_\tau = tanh(W_{FC}h_\tau + b_{FC}) \tag{2.15}$$

$$\hat{l}_\tau = W_{out}FC_\tau + b_{out} \tag{2.16}$$

where FC_τ is the FC layer output at time τ, $tanh$ is tanh activation function, and $\hat{l}_\tau$ is predicted value for the time frame τ. We use the CCC loss function that is defined in Sec 2.4.1.

2.5 Experimental Settings

For all of our experiments, we perform two-stage training over the BLSTM model, similar to [58]. We train the first stage for 20 epochs using Adam optimizer with a base learning rate of 0.002, and batch size of one utterance (7500 frames). For the 2nd stage, we start with the best CCC value we get from the first stage based on the development set. After each epoch, if the CCC value on the development set decreases for more than 0.01, we half the learning rate. If a better CCC value is found than the previous best CCC, the learning rate is rested to be 0.002. This continues until we reach 40 epochs or the learning rate becomes below 0.00001.

The number of BLSTM layers are cross-validated using the development CCC ($\{5, 7\}$ for arousal, $\{3, 5\}$ for valence) with size of 160 hidden units (80 for forward and 80 for backward paths) for speech-based experiments (Sections 2.6.1–2.6.4). For multimodal experiments (Section 2.6.5), we increase cross-validate the number of BLSTM layers over $\{5, 7, 9\}$ and the number of hidden units over $\{160, 320\}$ (80 for each path, and 160 for each path, respectively) to increase the model complexity for the increased feature dimensionality. The hyper-parameters α_1 and α_2 (0, 0.25, 0.50 and 1) in Eq. 2.13 are cross-validated with development CCC. We implement all of our experiments using Tensorflow [1][1].

We empirically found that initializing the final layer weights with Xavier initialization [23], and the biases with zero initialization are crucial to have stable training. To reduce the randomness of training, we run every experiment 3 times and use the average of 3 models' predictions as in [58].

Table 2.2: CCC differences between the end-to-end (with multiple α_1 and α_2) vs. BLSTM-FC models for speech-based experiments. Dev: development set, Test: test set.

Model			Arousal		Valence	
	α_1	α_2	dev	test	dev	test
clf+reg (end-to-end)	1	1	**0.868**	**0.697**	0.583	-
	0	0	0.854	-	0.608	-
	0.50	1	0.840	-	0.557	-
	0.25	1	0.864	-	0.591	-
	1	0.50	0.867	-	0.549	-
	1	0.25	0.858	-	**0.623**	0.530
reg (BLSTM-FC)			0.867	0.692	0.613	0.516
clf+reg (BLSTM-FC) (ensemble)			**0.869**	0.694	0.604	**0.538**

2.6 Results and Discussion

In this section, we first explore our proposed methods with audio features for the comparison with the state-of-the-art discretization work [58] (Sections 2.6.1–2.6.4) and then carry out multimodal experiments in Section 2.6.5. We compare our baselines *clf* and *reg* discussed in Section 2.4.1 against the previous state-of-the-art models (Section 2.6.1). In Section 2.6.2, we compare our proposed joint modeling against the strong baselines and previous work. In Section 2.6.3, we compare the BLSTM-FC and end-to-end model when the classification and regression losses are ignored ($\alpha_1 = \alpha_2 = 0$). This allows us to investigate whether it is beneficial to replace the discrete and continuous representations layer with a FC layer (BLSTM-FC). In Section 2.6.4, we investigate the classifier convergence for arousal and valence with *clf* and *clf+reg* (end-to-end) models to compare the convergence rate performance in the first stage of training. In Section 2.6.5, we compare the joint modeling and the baseline models in multimodal (audio-visual) environment.

[1] Our source code will be publicly available online.

2.6.1 Individual Modeling Methods

Table 2.1 shows the CCC onthe development (dev) and test (test) sets when using our proposed methods, baseline models, and previous state-of-the-art speech emotion recognition work. *clf* is the classification model presented in Section 2.4.1, *reg* is the regression model presented in Section 2.4.1, *clf*+*reg* (ensemble) is the ensemble model presented in Section 2.4.2, *clf*+*reg* (end-to-end) is the BLSTM model trained with classification and regression tasks end-to-end as presented in Section 2.4.2, and BLSTM-FC is the BLSTM model with FC layer before the final node as presented in Section 2.4.3.

As shown in Table 2.1, our proposed baseline models (*clf* and *reg*) achieve competitive results compared to previous state-of-the-art results. Both *clf* and *reg* outperform AVEC 2016 baseline model [111] by a large margin. For arousal, *clf* achieves 0.86 and 0.686 on the development and test sets respectively, and *reg* achieves 0.863 and 0.868. The Valstar et al. [111] baseline model, achieved 0.796 on the development set and 0.648 on the test set. For valence, *clf* achieves 0.558 on the development set and 0.527 on the test set, and *reg* gets 0.595 and 0.544, in comparison to 0.455 and 0.375 from Valstar et al. [111]. *clf* and *reg* also outperform the AVEC 2016 challenge winner, Brady et al. [9]. Brady et al. [9] achieved 0.846 on arousal and 0.450 on valence for the development set.

We also found that using *norm cost* improves the classification task results on arousal (0.860 for the development set and 0.686 for the test set) compared to the original loss function computation introduced by Le et al. [58]. Le et al. [58] reported 0.858 CCC value on the development set and 0.682 on the test set. For valence, *clf* with *norm cost* also outperforms the original classification model introduced by Le et al. [58] with a great margin (0.079) on the test set, where *clf* with *norm cost* gets 0.527 CCC value compared to 0.448 from Le et al. [58].

2.6.2 Joint Modeling Methods

Joint modeling (ensemble and end-to-end) improves the overall CCC for both arousal and valence compared to our proposed baselines (*clf* and *reg*) and previous state-of-the-art models.

We found that the end-to-end model has the best performance on arousal prediction where it gets 0.868 for the development set and 0.697 for the test set. These results are higher than *clf* (0.860 and 0.686 on the development and test set respectively) and *reg* (0.863 and 0.686) baselines.

Our proposed joint modeling outperforms Valstar et al. [111] with a large margin (0.796 and 0.648 on on the development set test sets respectively) and Brady et al. [9] (0.846 on the development set).

For valence prediction, ensemble model has better performance in terms of the test set CCC value, where it achieves 0.601 on the development set and 0.555 on the test set compared to 0.623 and 0.530 for the end-to-end model. These results are higher than *clf* (0.558 and 0.527 on the development and test set respectively) and *reg* (0.595 and 0.544) baseline models. Compared to previous state-of-the-art results, Valstar et al. [111] achieved 0.455 and 0.375 CCC value and Brady et al. [9] achieved 0.450 on the development set.

We also achieve the best performance compared to the best model proposed by Le et al [58] (0.859 on the development set and 0.680 on the test set for arousal and 0.596 and 0.460 for valence) where they used BLSTM for classification with additional emotion decoding step. We found that both of our proposed joint models (ensemble and end-to-end) get new state-of-the-art results on arousal (0.868) on the development set where we only use the audio features, compared to Brady et al. [9], where they received 0.862 when they used audio, video, and physiological data. Our proposed models have additional advantage where we get the final predictions with a single model without performing any additional post-processing step on the predicted values.

2.6.3 End-to-End vs. BLSTM-FC

In this section, we first compare BLSTM-FC model with the special case in the end-to-end model where both classification (ℓ_{clf}) and regression (ℓ_{reg}) tasks contributions in the final loss are ignored ($\alpha_1 = \alpha_2 = 0$). Second, we investigate the different combination of α_1 and α_2 that control the contribution of the classification and regression task for the end-to-end model. Third, we explore the ensemble model when BLSTM-FC is used as the regression model instead of *reg* model presented in Section 2.4.1.

First of all, the comparison between the end-to-end model with $\alpha_1 = \alpha_2 = 0$ and BLSTM-FC model will demonstrate whether to keep the model architecture the same or if it is more beneficial to replace the layer before the final node with FC layer as in BLSTM-FC model. We found that it is better to use an FC layer instead of the softmax and linear activations. BLSTM-FC gets 0.867 for arousal, and 0.613 for valence on the development set compared to 0.854, and 0.608 for arousal and valence respectively for *clf*+*reg* (end-to-end).

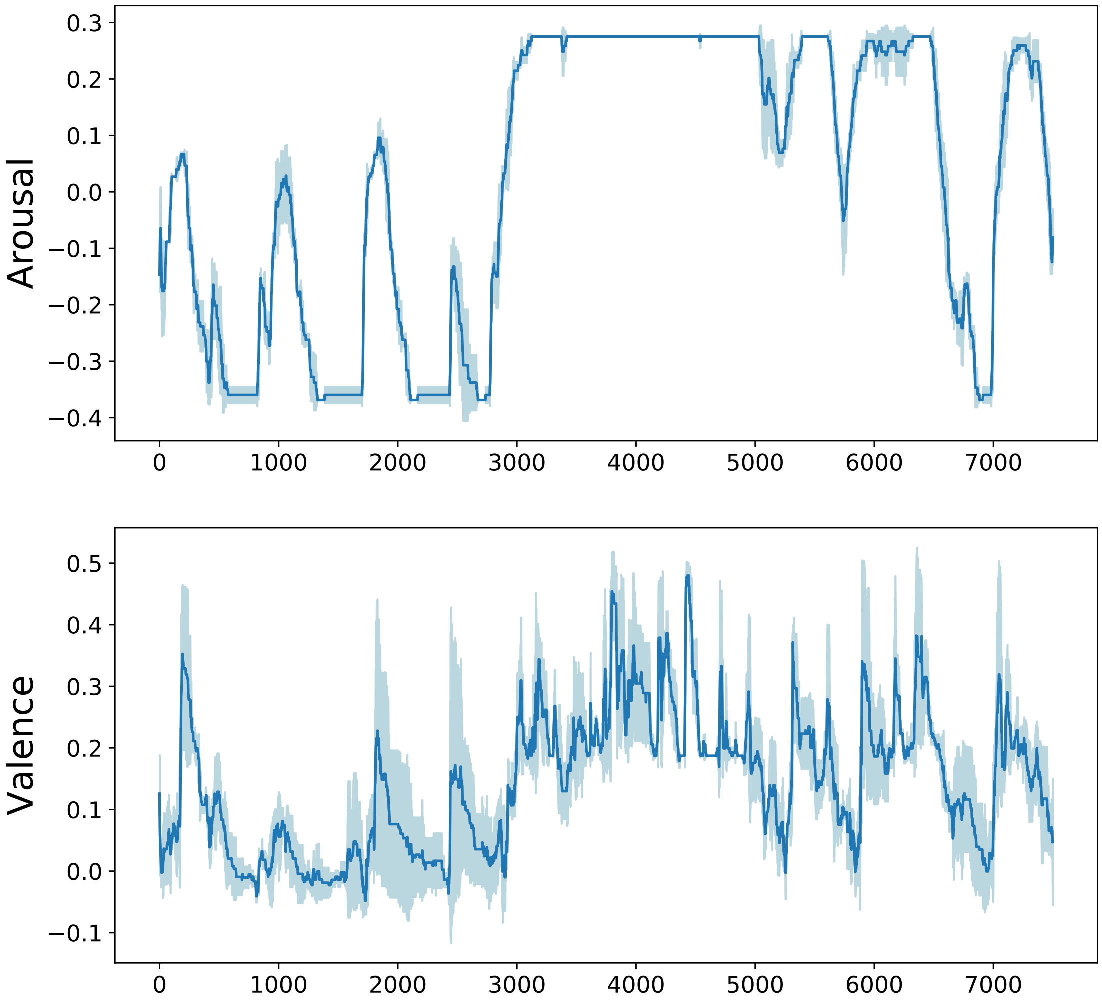


Figure 2.2: The 95% confidence interval for both arousal (top) and valence (bottom) on a given speaker from the development set while using *clf* model.

By cross validating $\alpha 1$ and $\alpha 2$ on the development set (Table 2.2), we found that $\alpha 1 = \alpha 2 = 1$ works the best for arousal (0.868 on the development set and 0.697 on the test set), and $\alpha 1 = 1$, $\alpha 2 = 0.25$ for valence (0.623 and 0.530). For this specific choice of $\alpha 1$ and $\alpha 2$ = 1, *clf+reg* (end-to-end) performs slightly better than BLSTM-FC (0.867 on the development set and 0.692 on the test set for arousal, 0.613 and 0.516 for valence). For other combinations of $\alpha 1$ and $\alpha 2$, there is no consistent trend found. For instance, when α_1 is greater than α_2 and more weight is given to the classification loss than regression, there is no consistent performance. In the case where $\alpha_1 = 1$ and $\alpha_2 = 0.5$ we get higher CCC on arousal (0.867) compared to 0.840 when $\alpha_1 = 0.5$ and $\alpha_2 = 0.1$. However, for $\alpha_1 = 1$ and $\alpha_2 = 0.25$ we get lower CCC (0.858) compared to 0.864 when $\alpha_1 = 0.25$ and $\alpha_2 = 0.1$ for arousal.

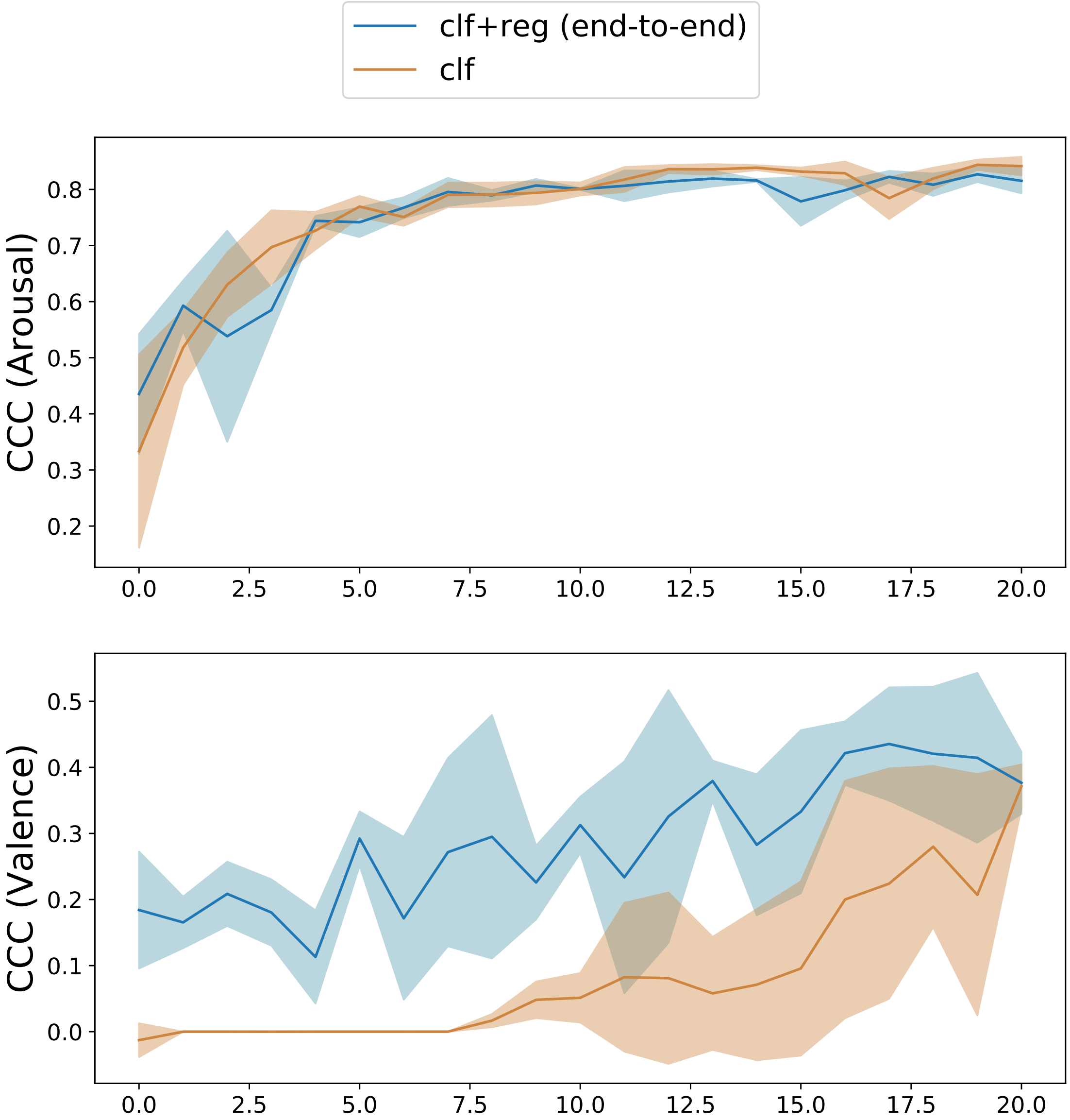


Figure 2.3: Learning curves for the convergence rate comparison between the *clf* and *clf+reg* (end-to-end) models, for arousal (top) and valence (bottom) prediction (best shown in color).

Based on BLSTM-FC development set results, we found that it is better to perform the regression task for both arousal and valence with the additional FC layer before the final node. BLSTM-FC achieves CCC value of 0.867 compared to 0.863 from the *reg* model for arousal, and 0.613 compared to 0.595 for valence. By replacing *reg* with BLSTM-FC in *clf+reg* (ensemble) model, CCC improves from 0.868 to 0.869 for arousal, and from 0.601 to 0.604 for valence achieving the heights CCC for arousal.

2.6.4 *clf* and *clf+reg* (end-to-end) Convergence

Figure 2.2 shows a visualization of 95% confidence interval of the *clf* model. It demonstrates that the 95% confidence interval for valence has a large margin over 3 different runs. This is because valence highly depends on video features, and it is not easy to predict using only audio features [9]. For arousal, *clf* is more stable over the 3 runs, especially for the regions where there is no transition in the emotion state.

To investigate the large margin in confidence interval for valence, we further explore the learning curve of *clf* and *clf+reg* (end-to-end) models (Fig. 2.3). The learning curve analysis can provide insight into the convergence rate for *clf* and *clf+reg* models. The learning curves demonstrate that *clf+reg* (end-to-end) performs more efficiently in terms of valence convergence compared to *clf* model. We found that for valence, the learning curve for *clf+reg* (end-to-end) model has an accelerated convergence speed. For *clf+reg* (end-to-end), the model starts with CCC around 0.2 for the development set and increase over the first 20 epochs. However, for *clf* model, the training starts with CCC around zero and does not increase until it reaches to the 7th epoch. For arousal on the other hand, both *clf* and *clf+reg* (end-to-end) models to have similar learning curves for the first 20 epochs.

2.6.5 Joint Modeling in Multimodal Environment

Table 2.3: CCC differences between baselines and proposed joint model (ensemble) using multimodal (audio-visual) features. Dev: development set, Test: test set.

Model	**Arousal**		**Valence**	
	dev	test	dev	test
clf [baseline]	0.864	0.690	0.698	0.622
reg [baseline]	0.863	0.699	0.681	0.583
clf+reg (ensemble)	**0.869**	0.699	**0.705**	0.617

Table 2.3 demonstrates how our best joint modeling method performs in multimodal (audio-visual) experiments. For the development set, the highest CCC is achieved when a joint model (ensemble) is used for both arousal and valence prediction. For arousal, *clf* and *reg* achieve 0.864 and 0.863, whereas the *clf+reg* (ensemble) achieves 0.869. For valence, *clf* and *reg* achieve 0.698 and 0.681, whereas the *clf+reg* (ensemble) achieves 0.705. These results are higher than the audio-based performance in Table 2.1, demonstrating the importance of using multimodal features in

predicting emotion. The improvement compared to unimodal systems shows a larger margin for valence, where valence CCC improves from 0.555 to 0.705 (27.02% improvement). This finding is consistent with previous work that demonstrated that visual features contribute more on valence prediction. Both arousal and valence results achieve higher than the RECOLA multimodal baseline [111]. These results demonstrate that the joint representation of discrete and continuous emotion helps the overall prediction compared to individual classification or regression models.

On the other hand, for test set, the ensemble model achieves higher CCC than *clf* for arousal (0.699 vs. 0.690, respectively), however the CCC remains the same compared to *reg*. Also, the valence prediction gets the highest CCC for *clf* (0.622), and the second highest for *clf*+*reg* (ensemble, 0.617), and finally *reg* (0.583). We assume that the different performance trend shown between the test and development sets may indicate the difficulty of valence prediction in the RECOLA dataset. Indeed, previous multimodal systems have shown a relatively better performance for arousal than valence [111, 9].

2.7 Conclusions

In this paper, we propose joint modeling methods that combine discrete (classification, *clf*) and continuous (regression, *reg*) emotion prediction. Our results indicate that joint modeling helps the training stage converge faster for the valence dimension, where the end-to-end model reaches CCC value of 0.3 starting from the 5th epoch, and keep increasing to reach CCC value of 0.4 at 16th epoch. However, for *clf* model it reaches its best CCC value (0.3) at 20th on the first stage of training. The results demonstrate that our proposed joint modeling approaches, ensemble and end-to-end, can predict continuous emotion labels more accurately than previous baseline approaches, especially for valence prediction. By further investigating the learning curves, we found that joint modeling has faster convergence in comparison to individual models. This work provides insight into the new joint representation of continuous and discrete emotion.

CHAPTER 3

Voice Conversion Using Speech-to-Speech Neuro-Style Transfer

3.1 Introduction

Recently, deep neural networks have been widely used for speech synthesis using different techniques such as text-to-speech (TTS) [78, 97, 81, 55, 120] and speech-to-speech based approaches [8, 43, 42]. Despite the significant improvement in the synthesized audio quality introduced by [78], TTS based approaches tend to miss the emotional characteristics in the speech sample for a given speaker. We argue that using a similar approach as in speech-to-speech systems can overcome such a problem and improve the synthesized speech quality.

Speech style transfer is the technique for synthesizing one speech sample of a target speaker from one of a different source speaker by keeping the linguistic information and the style of the original speaker. In this work, we introduce a speech-to-speech neural network system that is able to transfer the speech style across different speakers. Our approach consists of two primary steps. Firstly, given a mel-spectrogram speech utterance input, we train VAE-GAN model to reconstruct the input sample using L1-loss for the target speaker and GAN loss for other different speakers. To further refine the model performance, we introduce the latent space loss on the VAE encoder features embedding as well as cycle consistency loss [138]. The training is performed end-to-end in an unsupervised manner without any alignments between the input samples. Secondly, we train WaveNet-based vocoder on the VAE-GAN mel-spectrogram outputs to generate the synthesized speech in the time domain. Our method is inspired by a recent image-to-image style transfer model [62] applied to mel-spectrograms. Our method uses a single encoder for all input speakers to make it more feasible to generalize to multiple target speakers. Moreover, we introduce the latent loss to further constrain the encoded features in eliminating input speaker identity. This allows us to generate natural human-like synthesized speech with a unique style for each speaker.

We use three different objective metrics to evaluate our model, namely: (1) ASVSpoof [105] for measuring the difficulty of distinguishing between the synthesized and real human samples, (2) content verification for evaluating integrity in transferring the linguistic information between the source and the target speakers, and (3) speaker encoding [43] for validating the speaker identity in the synthesized speech samples. Experimental evaluations on the Flickr8k audio corpus [86] show the effectiveness of our method in generating human-like speech samples while capturing the linguistics and speech style of the input speaker.

3.2 Related Work

3.2.1 Speech Synthesis

We focus discussion on neural network based speech synthesis methods that are relevant to our current work. Hasegawa-Johnson *et al.* [33] proposed a sequence-to-sequence model to generate spoken description from the input image in the image2speech problem. They used both Flickr8k [86] and SPEECH-COCO [61] corpora to show the intelligibility of their model in generating relevant words and sequence them in a meaningful sentence. Jia *et al.* [43] proposed a neural network model to tackle TTS problem to generate synthesized speech for a given speaker. Their model contains three independent components; Speaker encoder, Synthesizer, and a neural vocoder based on Tacotron 2 [97]. They showed that their model is capable of synthesize speech for unseen speakers based on the features embedding coming from the speaker encoder model. Biadsy *et al.* [8] introduced a speech-to-speech model named Parrotron where it is trained end-to-end. They used their model for speech normalization where they map input spectrogram of different speakers to an output spectrogram of a single target speaker. Their model is trained to transfer the linguistic content to the target speaker while ignoring non-linguistic content. Our proposed model defers from Biadsy *et al.* [8] in that we preserve both the linguistic content and speech style of the input speaker for the transferring to the target speaker.

3.2.2 Neural Style Transfer

A notable amount of work has been introduced to tackle the style transfer problem mostly in the image domain. Liu *et al.* [62] proposed the UNIT model for image-to-image translation from one domain to another in an unsupervised manner, which is the major inspiration of the current work. This model consists of one encoder, generator, and a discriminator for each input/target domain [53]. In the audio domain, Mor *et al.* [74] presented a multi-domain WaveNet autoencoder to translate an input music record to different musical instruments and styles. Their model consists of one encoder and different target decoders each for target instruments. They used a domain con-

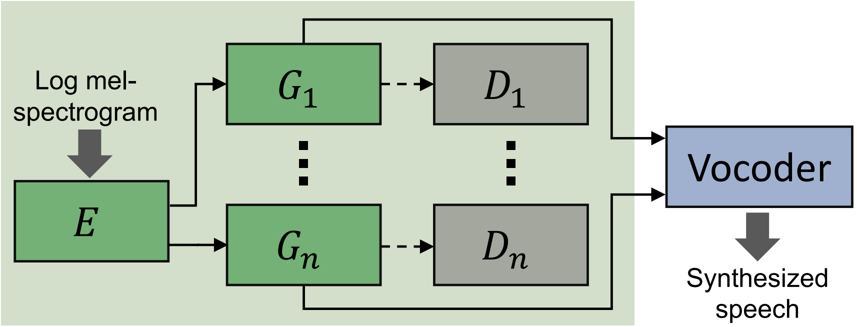


Figure 3.1: The overall pipeline of our proposed model. The VAE-GAN network is trained independently from the vocoder model.

fusion network to constrain the encoder not to memorize the input signal and produce a semantic encoding instead. In our work, we tackle the style transfer problem in the frequency domain using mel-spectrogram input/output instead of the time domain with waveforms. This helps us to have a more stable and faster training procedure while having simpler model architecture compared to [74] where they used a WaveNet model for both their encoder and decoders subnetworks.

3.3 Methods

In this section, we describe in detail the model architecture and the training procedure of our method. Fig 3.1 shows the overall pipeline of our method. The input to the system is a speech signal of one speaker, which is converted to the target speaker's voice while keeping the content and style of the original speaker. The input speech signal is first converted to the mel-spectrogram representation (details in Sec 3.4.1). We then employ a neural style transfer model similar to [62], treating the input mel-spectrogram as a gray-scale image, to create an output mel-spectrogram with the style of the target speaker. The generated mel-spectrogram is then fed to the vocoder to reconstruct the speech signal in the time domain. We use the WaveNet vocoder [112] based on the open-source implementation [128].

3.3.1 Framework

The core components of the neural style transfer model are a pair of convolutional neural networks corresponding to the encoder and generator (decoder), as shown in Fig.3.1. The encoder preserves the linguistic information in the input speech while removes identity-related informa-

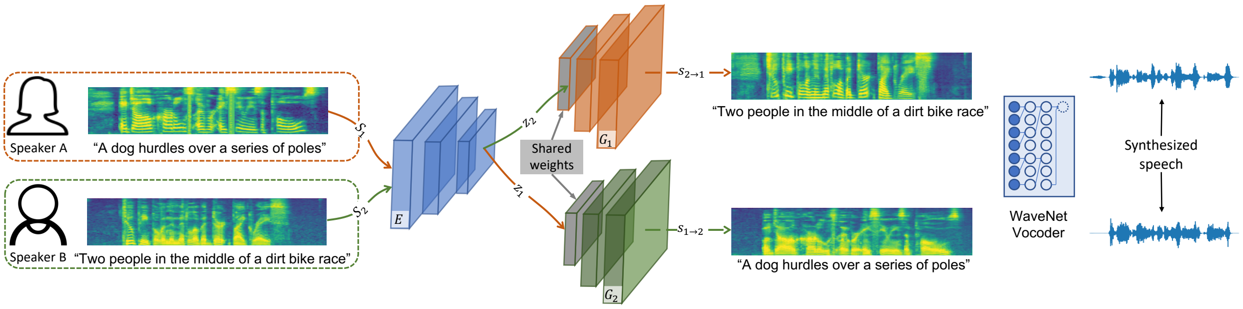

Speaker A
"A dog hurdles over a series of poles"
S_1
Speaker B
"Two people in the middle of a dirt bike race"
S_2
E
z_2
z_1
Shared weights
G_1
G_2
$s_{2\to1}$
$s_{1\to2}$
"Two people in the middle of a dirt bike race"
"A dog hurdles over a series of poles"
WaveNet Vocoder
Synthesized speech

Figure 3.2: The overall procedure of voice conversion with our method for style transfer between two different speakers during evaluation. The Encoder E generates features embedding for a given input speaker which then based on one of the Generators (G_1 or G_2) to produce a mel-spectrogram output for the target speaker. We use WaveNet-based vocoder to generate the speech in the time domain.

tion. The generator combines the style and the content of the input speech signal to create the mel-spectrogram of a new speech signal. To ensure the encoder capture identity-independent attributes such as speech volume and tempo, there is one single encoder regardless of the identities of the input speakers. We assume that using a shared encoder for the different speakers will imply a shared-latent space z that contains each sample content while removing the original speaker identity.

The architecture of the encoder E has three main parts. The first part contains the initial convolutional layer with 7×7 kernel size and no stride. The second part has two down-sampling convolutional layers with 4×4 kernel size and stride of 2. Each convolutional layer is followed by a batch normalization [40] and a LeakyReLU non-linear activation function. The third part consists of three residual blocks [63] as a final feature extractor.

Each of the different generators (G_1 to G_n where n is the number of target speakers) consists of the same parts as the encoder E in reverse order with two exceptions. First, instead of convolutional layers, we use transposed convolutional layer for up-sampling. Second, because of the shared latent space assumption coming from the single encoder E, all of the generators share the first residual block as a pre-processing step of the latent code for each of the generators [62].

To transfer the style from one speaker to another, we exchange the latent space for both decoders as shown in Fig 3.2. More specifically, for speaker S_i there is a latent code $z_i = E(S_i)$ where i is the i-th speaker. To get the same sentence spoken by different speakers, we feed the latent code z_i to the corresponding speaker Generator. Where $G_i(z_i)$ will reconstruct the same input $\tilde{S}_i$, and $G_j(z_i)$ will generate $S_{i\to j}$ that has the same content i being said by speaker j (i.e. style transfer between speakers i and j).

3.3.2 Training

The encoder and generator are trained in tandem using un-corresponded sets of speech signals of multiple subjects in an unsupervised manner. To facilitate the subsequent description, we will use the following notations:

- i and j are speaker indices where $i, j \in [1, n]$ and $i \neq j$
- S_i is a data point for speaker i drawn from distribution P_{S_i}
- $S_{i\to j}$ represents the translated speech from speaker i to speaker j where $S_{i\to j} = G_j(E(S_i))$
- $q(z_i|S_i)$ is probabilistic encoder produces distribution z_i given speaker sample S_i
- $p_{G_i}(S_i|z_i)$ is probabilistic generator for speaker i that produces distribution S_i given latent code z_i

The overall training loss of the neural style-transfer model for mel-spectrogram is defined as follows:

$$L = \lambda_1 L_{\text{VAE}} + \lambda_2 L_{\text{GAN}} + \lambda_1 L_{\text{CC}} + \lambda_3 L_{\text{latent}} \tag{3.1}$$

We explain each term in Eq.(3.1) in the following. The **VAE loss (L_{VAE})** is defined as:

$$L_{\text{VAE}} = \lambda_4 \sum_i D_{\text{KL}}(q(z_i|S_i)||p(z)) - \sum_i \mathbb{E}_{z_i\sim q(z_i|S_i)}[\log p_{G_i}(S_i|z_i)] \tag{3.2}$$

where the first term is the KL divergence (KLD) of the approximated posterior and the prior of the latent space and the second term is calculated through the Monte Carlo method, which can be understood in terms of the reconstruction of the input from the posterior distribution and the likelihood. For the KL-divergence we use prior distribution $p(z)$ as a zero mean Gaussian $\mathcal{N}(z|0, I)$ [53].

For each speaker there is GAN subnetwork that use the VAE subnetwork for the generation step followed by D_k as a discriminator where k is the index of a given speaker and $k \in [1, n]$.

For example, in speaker 1 we have GAN$_1$ that consists of G_1 and D_1. Positive samples for GAN$_1$ are sampled from S_1, while negative samples are G_1 outputs for input speaker i where $i \in [2, n]$. Then, the **GAN Loss (L_{GAN})** aims to penalize the VAE network for the translated samples between speakers i and j ($i \neq j$)

$$L_{\text{GAN}} = \sum_{i} \mathbb{E}_{S_i \sim P_{S_i}}[\log D_i(S_i)] + \sum_{i,j} \mathbb{E}_{S_{j\to i} \sim p_{G_i}(S_{j\to i}|z_j)}[\log(1 - D_i(S_{j\to i})] \tag{3.3}$$

The **Cycle Consistency (CC) Loss (L_{CC})** helps to enforce the speaker independent shared-latent space assumption by having a cycle-reconstruction stream [138]

$$L_{\text{CC}} = \lambda_4 \sum_{i,j} D_{\text{KL}}(q(z_j|S_{i\to j})||p(z)) - \sum_{i,j} \mathbb{E}_{z_j \sim q(z_j|S_{i\to j})}[\log p_{G_i}(S_i|z_j)] \tag{3.4}$$

Similar to VAE loss (L_{VAE}), we use KL divergence and negative log-likelihood for L_{CC} computation. The KL divergence penalizes the network on the latent codes of both the original and translated samples from the prior distribution. While the negative likelihood term ensures the reconstruction of the original S_i from the translated one $S_{i\to j}$.

The **Latent Loss (L_{latent})** is the L1-distance between the codes' centroids of all speakers i and j

$$L_{\text{latent}} = |C_i - C_j| \tag{3.5}$$

where C_i and C_j are the centroids of speakers i and j distributions respectively defined as follows:

$$C_i = \frac{1}{|P_{S_i}|} \sum_{S_i \in P_{S_i}} E(S_i). \tag{3.6}$$

We implemented our neural speech style transfer model using PyTorch framework, and train it on a Titan Xp GPU for approximately 8 hours. We use the Adam optimizer with $1e-4$ learning rate with a batch size of 4 samples for each speaker. The training algorithm is run for 100 epochs. For the regularization parameters in the objective functions, we use $\lambda_1 = 100$, $\lambda_2 = 10$, $\lambda_3 = 10$, and $\lambda_4 = 1e-3$. We choose these values of the regularization parameters to give more weight to the reconstruction loss in L_{VAE} compared to other loss terms.

Finally, the WaveNet vocoder [112] is also trained separately using the mel-spectrograms from both generators G_1 and G_2, where the target ground truth is the original waveform for each sample.

3.4 Experiments

3.4.1 Dataset and Feature Extraction

In our work, we use the Flickr8k Audio Caption Corpus [86], which contains $40,000$ spoken captions generated from $8,000$ photographs from `Flickr.com`. All utterances have a sampling rate of 16 kHz and are 1.9 seconds in length. To increase the diversity in the training/testing sets, we pick two speakers with opposite genders and have the most number of utterances. The total number of utterances for both training and testing are $4,668$ (60% male and 40% female). We strictly divide the dataset to training and testing sets with an approximate ratio of 70% and 30%, respectively. We use the training set to train both the VAE-GAN network and the WaveNet vocoder [112], while the testing set is used to evaluate the model performance.

We compute the log mel-spectrogram with 0.05 seconds window length and quarter-window overlap; this produces n windows and 128 frequency bands where n depends on each utterance length. For each training step, we randomly crop 128 consecutive frames from the log mel-spectrogram of each utterance to generate a 128×128 input sample.

We train the mel-spectrogram based neural style transfer model using the procedure in Sec 3.3.2. We then perform voice conversion with a style transfer experiment between the two selected speakers as described in Section 3.3.1.

3.4.2 Evaluation

To date, there has not been universally agreed objective metrics for the quality of the synthesized utterance. To this end, we use three objective methods to evaluate the naturalness of synthesized voices using our model. For fairness, we train each the evaluation methods on its original dataset it proposed with to eliminate any possible bias in our results. For testing, we report the final results on the held out test set using both the original and synthesized samples. For the synthesized samples, we only pick the ones with style transfer between two different speakers ($S_{i \to j}$). In the following, we provide more details about each evaluation method and the train set for each of them.

ASVspoof 2019 Baseline. The AVSspoof 2019 Challenge [105] provides a Gaussian Mixture Model (GMM)-based model as their main classifier with linear frequency cepstral coefficients [90] (LFCC) features. For training, we use the original ASVspoof 2019 dataset to train the GMM model with. We use the Equal Error Rate (EER) to evaluate the classifier performance in the test split. While the ASVspoof baseline does not measure the quality of the synthesized speech, we use it as a quantitative evaluation to measure the difficulty of distinguishing between real human voices and synthesized ones.

Content Verification Metric. To ensure that the synthesized speeches contain the same linguistic content as the original speech, we use the *word error rate* (WER) between the original transcript and the predicted one from the synthesized speeches to measure the intelligibility. We use the `SpeechRecognition` [134] open-source library to get the transcript of each sample.

Speaker Encoding Metric [43]. We use the `Speaker Encoder` model from [43] to verify if the speaker identity is preserved in the synthesized speech. This model uses a `long-short term memory` (LSTM) based RNN model for speaker encoding. The input to the model is the mel-spectrogram frames translated to a 265-dim vector for each speech sample. In training this model, a generalized end-to-end speaker verification loss [116] is minimized, where the samples with the same speaker preserve high cosine similarity, while the samples from different speakers are far apart in the embedding space. We use their pre-trained model without any fine-tuning and test whether the original and synthesized samples from the same speaker are in the same cluster. To classify each sample for one of the two speakers, we use the centroids of original samples for each of the two speakers using Eq 3.6, and then get the probability for each class using the following equation:

$$p(y = k|x) = \frac{\exp -d(f_\theta(x), c_k)}{\sum_{i=1}^{2} \exp -d(f_\theta(x), c_i)} \tag{3.7}$$

where k is the class number and $k \in 1, 2$, x is the input sample, $f_\theta x$ is the speaker encoder model, c_k is the centroid of speaker k, and d is the euclidean distance function. We use EER to evaluate the model performance.

Table 3.1: Equal Error Rate (EER) [%] using ASVSpoof and Speaker encoding metrics, and Word Error Rate (WER) [%] using Content Verification metric

Method	Data	EER/WER
AVSspoof 2019 [105]	Evaluation set in [105]	9.57
	Flickr8k [86] test split	38.89
Content Verification	Original samples	2.01
	Synthesised samples	10.36
Speaker Encoding [43]	Flickr8k [86] test split	0.001

3.4.3 Results

Table 3.1 summarizes the performance of our model with regards to the three evaluation methods in Sec 3.4.2. To test the difficulty of distinguishing between real and fake samples, we use the ASVspoof baseline [105]. From the held-out test set, we construct a balanced number of real and synthesized samples. The real samples come from the original speech of each speaker, where the synthesized samples are the ones with style transfer between two different speakers. The ASVspoof 2019 baseline method has a 38.89% EER on the Flickr8k test set while its performance on the original ASVspoof 2019 dataset is 9.57 EER [105]. This indicates it is more difficult to differentiate between real and synthesized samples from our method than those from the original ASVspoof baseline dataset.

For the second evaluation method, we use WER to see if the model preserves the original linguistic content in the synthesized samples. We first compute the WER between the original and the predicted transcripts from the speech samples using the open-source `SpeechRecognition` Library [134]. As shown in Table 3.1, we get 2.01% WER on the original samples. We use this value as an upper bound for the content verification method's performance. Computing the WER on the synthesized samples we achieve relatively close WER to the upper bound with 10.36% WER. This indicates the intelligence of the model to preserve the original content in the synthesized samples.

As shown in Fig 3.3, both the original and synthesized samples of the same speaker occupy the same cluster while there is a distinct separation between the two speakers' clusters. We compute EER on the predicted probabilities using Eq 3.7 where we achieve 0.001%EER (Table 3.1). These results demonstrate the efficacy of our proposed model to preserve the speaker's identity on the synthesized samples.

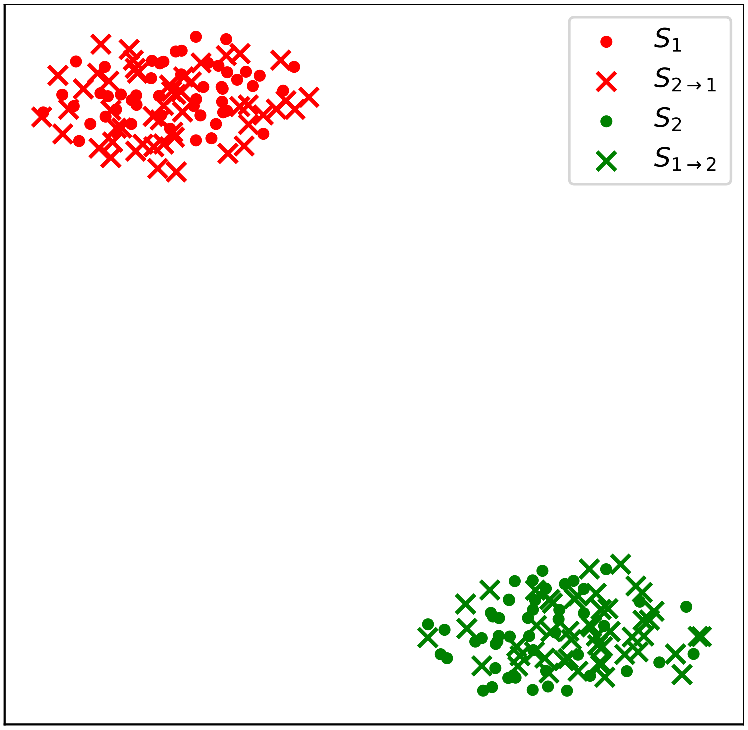


Figure 3.3: tSNE [113] visualization of the features embedding for each speaker on the original (S_1, S_2) and synthesized samples ($S_{2\rightarrow1}, S_{1\rightarrow2}$) using speaker encoding evaluation method [43].

CHAPTER 4

VocBench: A Neural Vocoder Benchmark for Speech Synthesis

4.1 Introduction

Throughout the years, speech synthesis techniques have gone through different phases of improvements, from knowledge-based approaches [103, 29, 7] to data-based ones [106, 119, 60]. To date, there are two types of speech synthesis algorithms, *text to speech*, which converts input text to audio signals, and *voice conversion*, which transforms an input audio to different identities or styles. Regardless of this difference, most of the recent speech synthesis approaches [97, 45, 4] rely on *neural vocoders* to generate the final waveform for more natural-sounding speech synthesis.

In this context, a vocoder is designed to synthesize waveform from the lower feature dimension, such as Mel-spectrograms. For many years, the state-of-the-art (SOTA) methods used DSP-based approaches [28] for vocoder development. While the advantage of fast speech generation time, the quality of the synthesized waveform is largely limited due to the assumptions under the heuristics. In recent years, more sophisticated vocoders have been developed based on the use of deep neural networks for more enhanced quality for the generated speech. These methods include (1) *autoregressive* approaches [112, 44], (2) *Generative Adversarial Networks (GANs)* approaches [56, 130, 129], and (3) *diffusion based* approaches [15, 54]. Due to the different variables in the evaluation process, datasets selection, hardware configuration, and evaluation metrics used, how best to compare and evaluate these different approaches remains an open challenge.

In this work, we present the VocBench framework, a comprehensive benchmark for vocoder quality and speed evaluations. More specifically, we build VocBench to train and test neural vocoders in a shared environment with public datasets. We construct three datasets, including one single-speaker and two multi-speaker scenarios, and then train six vocoders covering three different categories: *autoregressive*, *GAN*, and *diffusion* based approaches. All vocoders are trained and evaluated following the same pipeline. We design two main experiments. First, we test the efficacy of each vocoder in synthesizing the waveform from lower-dimensional features such as Mel-Spectrogram. Second, we test the generalizability of each vocoder in synthesizing speech for speakers who are not included in the training set. Figure 4.1 provides an overview of the proposed

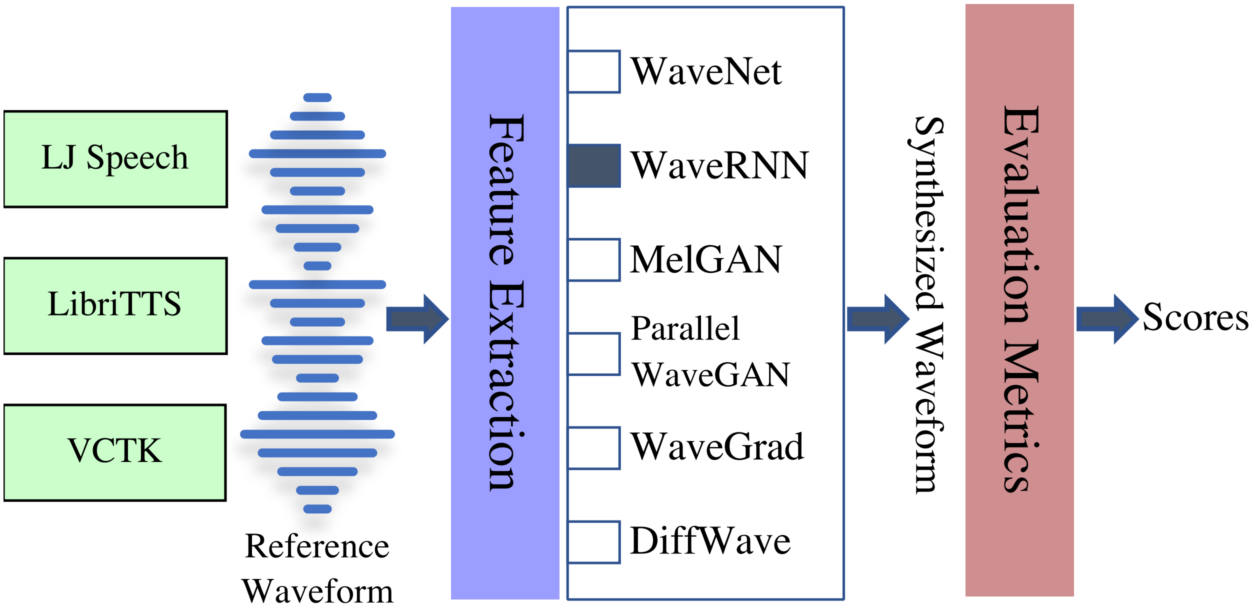


Figure 4.1: An overview of the proposed VocBench framework.

framework.

Recently, various studies have been conducted for neural vocoder evaluation. Govalkar *et al.* [26] conducted a study with six autoregressive-based vocoders and two additional phase reconstructions vocoders. Airaksinen *et al.* [2] adopted classical methods for vocoder design in their study. Both of these works used MUSHRA [92] as their main evaluation metric to compare the performance of each vocoder. In this study, we extend the vocoder implementations to include both GAN and diffusion-based models. Additionally, we carry on the evaluation using both subjective and objective metrics. We use the Mean Opinion Score (MOS) test as a subjective evaluation. To evaluate each of the different vocoders objectively, we used the following four different evaluation metrics: Structural Similarity Index Measure (SSIM) [122], Fréchet Audio Distance (FAD) [50], Log-mel Spectrogram Mean Squared Error (LS-MSE), and Peak Signal-to-Noise Ratio (PSNR). More details about the experiment setup and evaluation metrics are presented in $\mathcal{S}$ 4.3.

4.2 Neural Vocoders

We next describe the three main categories of the neural vocoders used in our study: the autoregressive models ($\mathcal{S}$ 4.2.1), GAN based models ($\mathcal{S}$ 4.2.2), and diffusion models ($\mathcal{S}$ 4.2.3).

4.2.1 Autoregressive Models

The key feature of the autoregressive models is that they are designed as probabilistic models to predict the probability of each waveform sample based on the previous samples. This allows generating a natural, high-quality speech signal. However, due to the sample-by-sample generation process, the overall synthesis speed is slow compared to other methods. In the following, we will consider two main autoregressive models: WaveNet and WaveRNN.

The **WaveNet** [112] model works on the waveform level to achieve long-range temporal dependency through the depth of the model. It combines a stack of causal filters and dilated convolutions to help their receptive fields grow exponentially with the depth. We use the open-source implementation from [128] with different configurations of input types and loss functions. More details are provided in $\mathcal{S}$ 4.3. The autoregressive **WaveRNN** [44] architecture utilizes a recurrent neural network (RNN) for sequential modeling of the target waveform. A single layer RNN with a dual softmax layer is used.

4.2.2 GAN Based Models

GAN-based vocoders have shown remarkable performance often exceeding autoregressive models in the speed and quality of the synthesized speech. The main idea of GANs [24] use a *generator* to model the waveform signal in the time domain and a *discriminator* to assess the quality of the generated speech. We consider two representative models, MelGAN and Parallel WaveGAN among the different variants of GAN-based vocoders.

MelGAN [56] takes the standard GAN architecture for fast waveform generation. A fully convolutional model is used for high-quality Mel-Spectrogram inversion. With fewer parameters compared to the autoregressive model, MelGAN achieves higher real-time factor on both GPU and CPU without the need of hardware-specific optimization.

The **Parallel WaveGAN** [129] architecture is distillation-free, fast, and requires only small memory footprint for waveform synthesis. Parallel WaveGAN jointly optimizes the waveform-domain adversarial loss and multi-resolution short-time Fourier transform (STFT) loss.

4.2.3 Diffusion Based Models

Diffusion probabilistic models are generative models entailing two main processes: *diffusion* and *reverse* [37]. The diffusion process is defined as a Markov chain that gradually adds Gaussian noise to the original signal until it gets destroyed. The reverse process, on the other hand, is a denoising process that progressively removes the added Gaussian noise and restores the original signal. We included two diffusion-based vocoders in our study: WaveGrad and DiffWave.

The **WaveGrad** [15] model architecture is built on prior works from score matching [115] and diffusion probabilistic models [37]. The WaveGrad model takes a white Gaussian noise as input, and condition on the Mel-Spectrogram to iteratively refine the signal via a gradient-based sampler.

DiffWave [54] is a versatile diffusion probabilistic model for waveform synthesis that works well under both conditional and unconditional scenarios. Using a white Gaussian noise as input, DiffWave performs a Markov chain process with a constant number of steps to gradually generate a structured waveform [98, 27, 37]. The model is trained to optimize a choice of variational bound on the data likelihood.

Table 4.1: Evaluation results for the four objective metrics (SSIM, LS-MSE, PSNR, and FAD) and the 5-scale MOS with 95% confidence intervals evaluated on the three datasets: LJ Speech, LibriTTS, and VCTK. We welcome researchers to submit or update their results at our GitHub repository https://github.com/facebookresearch/vocoder-benchmark for comparisons.

Metric	Corpus	WaveNet	WaveRNN	MelGAN	Parallel WaveGAN	WaveGrad	DiffWave	Griffin-Lim	Ground Truth
SSIM	LJ Speech	0.66	0.62	0.89	0.84	0.76	0.82	0.90	-
	LibriTTS	0.056	0.53	0.91	0.86	0.71	0.74	0.89	-
	VCTK	0.46	0.43	0.88	0.79	0.59	0.64	0.86	-
LS-MSE	LJ Speech	0.006	0.010	0.001	0.002	0.006	0.006	0.001	-
	LibriTTS	0.008	0.008	0.001	0.001	0.005	0.006	0.001	-
	VCTK	0.009	0.010	0.001	0.002	0.007	0.007	0.001	-
PSNR	LJ Speech	23.20	20.36	28.53	26.70	22.57	22.51	28.77	-
	LibriTTS	21.54	21.17	29.98	28.62	22.94	22.18	29.03	-
	VCTK	21.36	20.40	30.40	28.17	21.54	21.22	28.77	-
FAD	LJ Speech	1.05	3.43	1.51	**0.92**	3.12	3.62	2.69	0.31
	LibriTTS	1.55	2.60	2.95	**1.41**	3.10	3.74	4.27	1.23
	VCTK	**0.99**	3.59	1.76	1.22	4.10	5.59	3.92	0.61
MOS	LJ Speech	3.68±0.037	3.96±0.089	3.73±0.075	3.99±0.059	3.85±0.068	**4.07±0.060**	3.68±0.082	4.10±0.059
	LibriTTS	3.75±0.107	3.74±0.099	3.50±0.086	**3.82±0.069**	3.48±0.083	3.80±0.073	3.36±0.092	4.03±0.065
	VCTK	**3.95±0.032**	3.94±0.089	3.75±0.074	3.87±0.068	3.77±0.074	3.86±0.069	3.66±0.079	3.98±0.064

4.3 Dataset and Experiments

4.3.1 Dataset and Feature Extraction

We use three datasets in this study: LJ Speech for the *single-speaker* scenario as well as LibriTTS and VCTK for the *multi-speaker* scenarios. For all of the three different datasets, the train, validation, and test splits are fixed across the different vocoders that are used in our study.

The **LJ Speech** dataset [41] consists of $13,100$ short audio clips of a single speaker reading passages from 7 non-fiction books. A transcription is provided for each clip. The length of each clip varies from 1 to 10 seconds, and the total length is approximately 24 hours. We reserve the first 20 clips for testing, and the following 10 clips for validation. The rest of the clips are used for training.

The **LibriTTS** dataset [133] is a multi-speaker English corpus of approximately 585 hours of reading English speech at a 24kHz sampling rate. It is derived from the original materials (MP3 audio files from LibriVox and text files from Project Gutenberg) of the LibriSpeech corpus. We use *train-clean*-100 and *train-clean*-360 subsets for training with about 1150 speakers and 25 minutes of recordings on average per speaker. For validation and test splits, we use the *dev-clean* and *test-clean* subsets respectively.

The **VCTK** corpus [114] includes speech data uttered by 110 English speakers with various accents. Each speaker reads out about 400 sentences selected from a newspaper. We randomly select 85% of the samples for training data, 10% for validation, and 5% for testing.

Log-spectrogram computations. The speech signals in the three datasets are re-sampled to 24 kHz. We extract the 80-dimensional Mel-Spectrogram features using 40 ms Hanning window, 12.5 ms frameshift, 1024-point FFT, and 0 Hz & 12 kHz lower & upper-frequency cutoffs. We then perform $\log$ dynamic range compression on the resulting Mel-Spectrogram features followed by a min-max normalization.

4.3.2 Training Setup

For training each of the vocoders in our study, we conduct a hyperparameter search and report the best model configuration on the three different datasets described in $\mathcal{S}$ 4.3.1. Our framework is implemented on the PyTorch library, and training is performed on a Tesla V100 GPU. For reproducibility, we use the Amazon Web Services (AWS) to compute the evaluation metrics. Specifically, for CPU computations, we use `c5.4xlarge` AWS instance with 16 vCPU of 3.6GHz Intel Xeon Processors. For GPU computations, we use `p3.2xlarge` AWS instance with 8 vCPU of 2.3GHz Intel Xeon Processors and one NVIDIA Tesla V100 GPU.

For each of the vocoders, we start from the original configuration provided in the respective open-source implementation. However, for WaveNet, there are different configurations that vary in terms of input types and loss functions. For the input, we can use either raw waveform or pre-processed waveform using μ-law compression. For the loss function, there are two different options: Mixture of Logistics (MoL-loss) or a single Gaussian distribution (*normal-loss*). We run different versions of the WaveNet model using each configuration and report the one with the best performance. We found that on LJ Speech and VCTK, it is better to use μ-law compression on the input waveform; and on LibriTTS, raw waveform input yields the best results. For the loss function, using *normal-loss* helps to increase the overall performance.

4.3.3 Evaluation

Our aim is to evaluate multiple vocoders along different axes numerically and qualitatively. The choice of metrics is crucial for evaluation, and we consider the following metrics:

- **Mean Opinion Score (MOS)** is a subjective numerical measure of the human-judged overall quality after listening to a sample. We conducted the MOS study on each of the vocoder models with three different datasets. Each MOS test consists of 400 participants asked to rate the quality of each sample between 1-5 (1:bad - 5:excellent). We report the MOS for each vocoder as well as the ground truth over 20 samples from the test set.

- **Structural Similarity Index Measure (SSIM)** [122] is a quantitative metric that measures the similarity between two given images in the original study. We perform SSIM in the frequency domain to compare the synthetic spectrogram with the real-world sample.

- **Fréchet Audio Distance (FAD)** [50] measures the quality and diversity of the generated samples. FAD score is the distance between two multivariate Gaussian distributions estimated on the sets of embeddings, *i.e.* the background and evaluation embeddings. To generate these feature embeddings, FAD use a VGG model [36] trained on a large YouTube dataset as the audio classifier.

- **Log-mel Spectrogram Mean Squared Error (LS-MSE)** is computed between the ground truth spectrogram sample and a generated one. We use the computation in $\mathcal{S}$ 4.3.1 to obtain the Log-mel Spectrogram for the synthesized speech samples. The LS-MSE can be interpreted as a measure of how close the low-dimensional representation of the spectrogram is when compared to the ground truth spectrogram.

- **Peak Signal-to-Noise Ratio (PSNR)** is the ratio of the power of a peak signal, which is the magnitude of the best-case output of a signal to the power of the noise at the peak measured in dB. We apply PSNR computation in the frequency domain, where the peak signal of the output is 1 and the distorting noise is represented by LS-MSE.

4.3.4 Results and Discussion

Table 4.1 shows the results of the five objective and subjective evaluation metrics described in $\mathcal{S}$ 4.3.3. Each of the metrics are computed using 20 audio samples from each dataset. For MOS, we report the mean value as well as the 95% confidence intervals. We use the Griffin-Lim vocoder [28] as a baseline to compare with each of the other vocoders in our study.

FAD and MOS metrics show close correlation especially for GAN-based vocoders. Both metrics have the same best-performing models in each dataset except LJ Speech. MOS reports that Diffwave is the best performing vocoder for LJ Speech with a MOS score of 4.07 ± 0.06, while Parallel WaveGAN achieves the best FAD score of 0.92 (which is the second-best in terms of MOS 3.99 ± 0.059). For LibriTTS dataset, Parallel WaveGAN has the best performance for both FAD (1.41) and MOS (3.82 ± 0.69). As for VCTK dataset, WaveNet achieves lower FAD (0.99) and higher MOS (3.95 ± 0.032) scores compared to other vocoders.

Observe that when using FAD and MOS metrics, each of the different models achieves their best performance on LJ Speech dataset, while having lower performance on VCTK and LibriTTS, respectively. This is due to the fact that LJ Speech is a single-speaker dataset which makes it easier to train and evaluate on. On the other hand, VCTK and LibriTTS are multi-speaker datasets. When LibriTTS is used for speaker generalizability, the scenario is more challenging as suggested from our experimental results.

Table 4.2 shows the model complexity of each vocoder and how that affects the voice synthesis computation time. We compare the following aspects of the neural vocoder models: the model parameter size, the number of Floating Point Operations per Second (FLOPS) of a speech sample, total training iterations, and Real-Time Factor (RTF).

The autoregressive models, namely WaveNet and WaveRNN, have a consistent number of parameters (3.79 and 4.35 Million parameters respectively) and FLOPS (89.65 and 94.98 GFLOPS respectively) when compared to other models. We exclude the RTF computation for the autoregressive model as they are significantly slower compared to other vocoders in our study. For real-time applications, custom kernels are used for autoregressive models such as LPCNET [109].

For GAN-based vocoders, we report the number of parameters and FLOPS for the generator. MelGAN has fewer number of FLOPS (3.01 GLOPS) compared with the Parallel WaveGAN (31.26 GLOPS). This difference is also reflected in the RTF values, where MelGAN has RTF 0.001 RTF for GPU and 0.029 for CPU. On the other hand, Parallel WaveGAN achieves 0.002 RTF on GPU and 0.576 on CPU.

In Diffusion-based vocoders, WaveGrad has a relatively higher number of parameters (15.81 Million parameters) compared to DiffWave, while both models maintain the same order of magnitude for the number of FLOPS (33.75 and 31.70 GFLOPS respectively). We report the computation of a single step of the inference for both the number of model parameters and FLOPS. In our experiments, during inference we use 50 steps noise scheduler for WaveGrad and 6 steps for DiffWave, following the original implementation. This explains the higher RTF obtained for both vocoders in comparison to GAN-based, where WaveGrad has 0.381 RTF on GPU and 9.858 on CPU, respectively. DiffWave reports 0.070 and 4.452 RTF on GPU and CPU respectively.

Table 4.2: Space and time complexity for vocoders under evaluation in terms of: (1) the number of parameters, (2) computation FLOPS, and (3) their corresponding RTF using on GPU and CPU setup. #Param for GANs (MelGAN and Parallel WaveGAN) is only for the generator and for a single step of inference for the diffusion models (WaveGrad and DiffWave).

Model	**#Param (M)**	**GFLOPS**	**RTF**	
			GPU	CPU
WaveNet	3.79	89.65	-	-
WaveRNN	4.35	94.98	-	-
MelGAN	3.05*	3.01	0.001	0.029
Parallel WaveGAN	1.34*	31.26	0.002	0.576
WaveGrad	15.81*	33.75	0.381	9.858
DiffWave	2.62*	31.70	0.070	4.452

CHAPTER 5

Detecting AI-Synthesized Speech Using Bispectral Analysis

5.1 Introduction

Recent advances in AI-synthesized content-generation are leading to the creation of highly realistic audio clips [81, 30], images [47, 46], and videos [76, 51, 104, 101, 11]. While there are many interesting and artistic applications for this type of synthesized content, these same techniques can also be weaponized. Nowadays, advances in deep learning have led to the development of synthesis tools for creating forgery videos and audios that can be very harmful. For example, it is not difficult for a hacker to create a forgery video of a world leader threatening another nation leading to an international crisis. People with bad intentions can create a fake video of a presidential candidate saying something inappropriate which, if released 24 hours before an election, could lead to interference with a democratic election. One more example, a fake video of a CEO privately claiming that the company's profits are down can turn into a part of global stock manipulation.

As these synthesis tools become more powerful and readily available, there is a growing need to develop forensic techniques to detect the resulting synthesized content. We describe a technique for distinguishing human speech from synthesized speech that leverages higher-order spectral correlations revealed by bispectral analysis. We show that these correlations are not present in a wide variety of recorded human speech, but are present in speech synthesized with several state of the art AI systems. We also show that these correlations are likely the result of fundamental properties of the synthesis process, which would be difficult to eliminate as a counter measure.

In the general area of audio forensics, there are a number of techniques for detecting various forms of audio spoofing [132]. These techniques, however, do not explicitly address the detection of synthesized speech. Previous work [19] showed that certain forms of audio tampering can introduce the same type of higher-order artifacts that we exploit here. This previous work, however, did not address the issue of synthesized content.

In comparing different features and techniques for synthetic-speech detection, the authors in [91] found that features based on high-frequency spectral magnitudes and phases are most effective for distinguishing human from synthesized speech. These features are based on first-order Fourier coefficients or their second-order power spectrum correlations. In contrast to these first- and second-order spectral features – which might be easy to adjust to match human speech – we explore higher-order polyspectral features which are both discriminating and should prove to be more difficult to adjust by the synthesizer.

5.2 Methods

We begin by describing the data set of human and synthesized content that we recorded and created. We then describe the polyspectral analysis tools that underlie our technique followed by a qualitative assessment of the differences in the bispectral properties of human and synthesized content. We conclude this section with a description of a simple classifier that characterizes these differences for the purposes of automatically distinguishing between human and synthesized speech.

5.2.1 Data set

We collected a data set consisting of $1,845$ human and synthesized speech recordings. The human speech are obtained from nine people (five male and four female). These recordings were extracted from various high-quality podcasts. Each recording averaged 10.5 seconds in length.

The same texts spoken by the human subjects (transcribed from the recordings) were used to synthesize audio samples using various automatic text-to-speech synthesis methods including Amazon Polly, Apple text-to-speech, Baidu DeepVoice, and Google WaveNet[1]. We also include samples generated using the Lyrebird.ai API, which, unlike other synthesis methods, generates personalized speech styles (because of limited access to this API, the texts spoken were not matched to the human and other synthesized speech). In synthesizing these recordings, a range of speaker profiles was selected to increase the diversity of the synthesized voices.

[1] Sources: Amazon Polly `aws.amazon.com/polly/`, Apple text-to-speech API `developer.apple.com/documentation/appkit/nsspeechsynthesizer`, Baidu DeepVoice `r9y9.github.io/deepvoice3_pytorch/`, and Google WaveNet `r9y9.github.io/wavenet_vocoder/`.

5.2.2 Bispectral Analysis

In this section, we describe the basic statistical tools used to analyze audio recordings. The bispectrum of a signal represents higher-order correlations in the Fourier domain.

An audio signal $y(k)$ is first decomposed according to the Fourier transform:

$$Y(\omega) = \sum_{k=-\infty}^{\infty} y(k) e^{-ik\omega}, \tag{5.1}$$

with $\omega \in [-\pi, \pi]$. It is common practice to use the power spectrum of the signal $P(\omega)$ to detect the presence of second-order correlations, which is defined as:

$$P(\omega) = Y(\omega)Y^*(\omega), \tag{5.2}$$

where $*$ denotes complex conjugate. The power spectrum is, however, blind to higher-order correlations, which are of primary interest to us. These third-order correlations can be detected by turning to higher-order spectral analysis [69]. The bispectrum, for example, is used to detect the presence of third-order correlations:

$$B(\omega_1, \omega_2) = Y(\omega_1)Y(\omega_2)Y^*(\omega_1 + \omega_2). \tag{5.3}$$

Unlike the power spectrum, the bispectral response reveals correlations between the triple of harmonics $[\omega_1, \omega_1, \omega_1 + \omega_1]$, $[\omega_2, \omega_2, \omega_2 + \omega_2]$, $[\omega_1, \omega_2, \omega_1 + \omega_2]$, and $[\omega_1, -\omega_2, \omega_1 - \omega_2]$. Note that, unlike the power spectrum, the bispectrum in Equation (5.3) is a complex-valued quantity. From an interpretive stance it will be convenient to express the complex bispectrum with respect to its magnitude:

$$|B(\omega_1, \omega_2)| = |Y(\omega_1)| \cdot |Y(\omega_2)| \cdot |Y(\omega_1 + \omega_2)|, \tag{5.4}$$

and phase:

$$\angle B(\omega_1, \omega_2) = \angle Y(\omega_1) + \angle Y(\omega_2) - \angle Y(\omega_1 + \omega_2). \tag{5.5}$$

Also from an interpretive stance it is helpful to work with the normalized bispectrum [18], the bicoherence:

$$B_c(\omega_1, \omega_2) = \frac{Y(\omega_1)Y(\omega_2)Y^*(\omega_1 + \omega_2)}{\sqrt{|Y(\omega_1)Y(\omega_2)|^2 |Y(\omega_1 + \omega_2)|^2}}. \tag{5.6}$$

This normalized bispectrum yields magnitudes in the range $[0, 1]$.

In the absence of noise, the bicoherence can be estimated from a single realization as in Equation (5.6). However in the presence of noise some form of averaging is required to ensure stable estimates. A common form of averaging is to divide the signal into multiple segments. For example the signal $y(n)$ with $n \in [1, N]$ can be divided into K segments of length $M = N/K$, or K overlapping segments with $M > N/K$. The bicoherence is then estimated from the average of each segment's bicoherence spectrum:

$$\hat{B}_c(\omega_1, \omega_2) = \frac{\frac{1}{K}\sum_k Y_k(\omega_1)Y_k(\omega_2)Y_k^*(\omega_1+\omega_2)}{\sqrt{\frac{1}{K}\sum_k |Y_k(\omega_1)Y_k(\omega_2)|^2 \frac{1}{K}\sum_k |Y_k(\omega_1+\omega_2)|^2}}. \tag{5.7}$$

Throughout, we compute the bicoherence with a segment length of $N = 64$ with an overlap of 32 samples.

5.2.3 Bispectral Artifacts

Shown in Figure 5.1 is the bicoherent magnitude and phase for three different human speakers. Shown in the second to the sixth rows are the bicoherent magnitude and phase for five different synthesized voices, as described in Section 5.2.1. Each bicoherent magnitude and phase panel are displayed on the same intensity scale. At first glance, there are some glaring differences in the bicoherent magnitude (with the exception of Apple) between the human and synthesized speech. There are also strong differences in the bicoherent phases across all synthesized speech.

Because most synthesis methods use deep neural networks, we hypothesize that these bicoherence differences are due to the underlying speech-synthesis network architecture and, in particular, that long-range temporal connections give rise to the unusual spectral correlations. To determine if this might be the case, we created three "clipped" WaveNet network architectures in which the network connectivity was effectively reduced. This was done by first noticing that WaveNet employs 3-tap filters in its convolutional layers. We, therefore, truncate the full WaveNet models in which the left-most value of the convolution filter in one of three layers was fixed at a value of zero[2]. With a total of 24 convolutional layers we performed this manipulation at level 24 (closest to the output level), 12, or 1 (closest to the input level). The effective network clipping was more pronounced for the manipulations at the levels closest to the input level, as this clipping propagates through the entire network.

[2] A more direct approach is to use simply use a 2-tap filter. This, however, would require retraining the entire model and so we adopted the simpler approach of zeroing out one of the filter values.

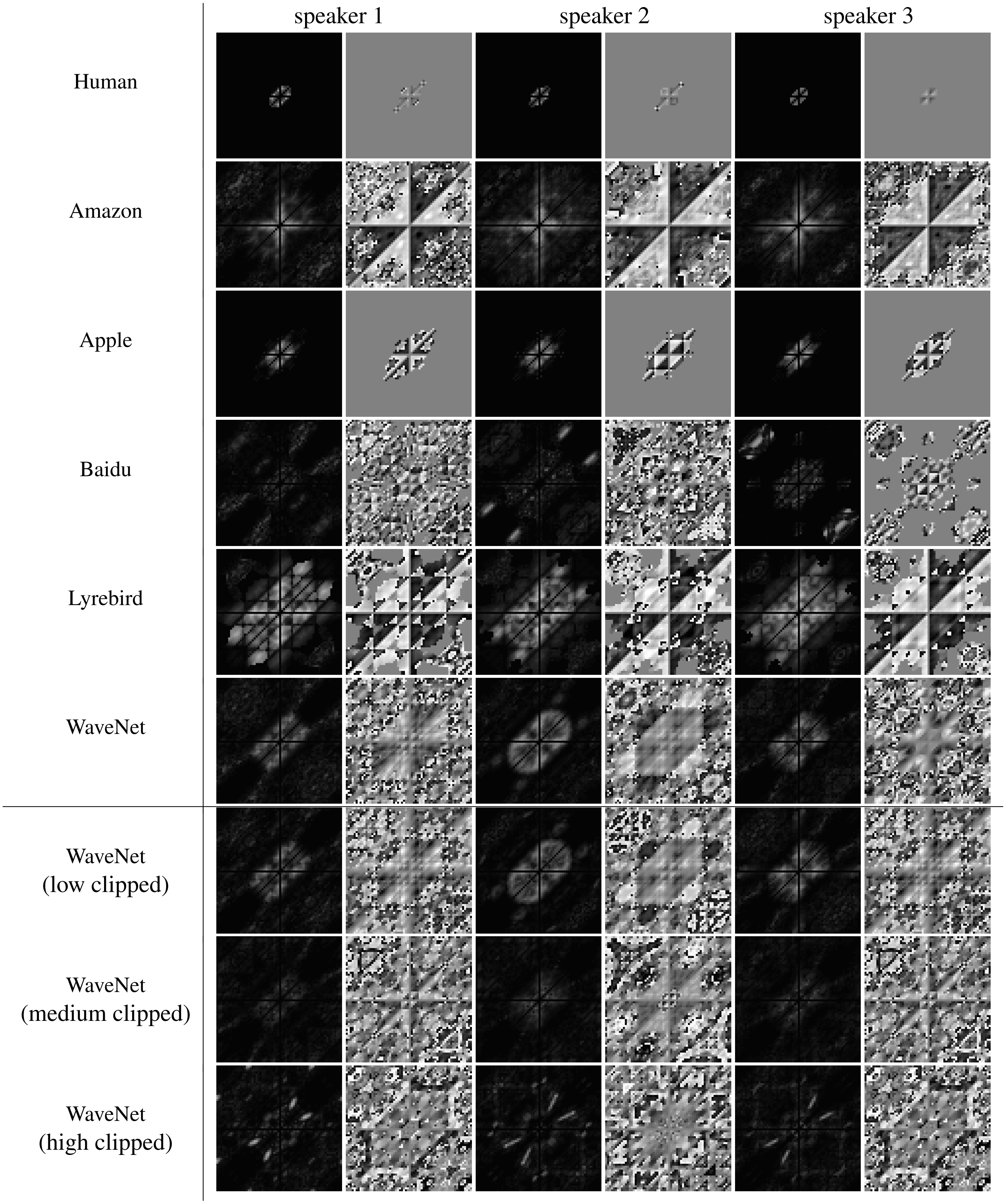


Figure 5.1: Bicoherent magnitude and phase for human speakers and five different synthesized voices. Shown in the lower three rows are the results for three different clipped versions of the WaveNet architecture. The magnitude plots are displayed on an intensity scale of $[0, 1]$ and the phase plots are displayed on a scale of $[-\pi, \pi]$.

Shown in the last three rows of Figure 5.3 are the resulting bicoherence magnitudes and phases for three recordings synthesized with these three networks with increasing amounts of “clipping”. As can be clearly seen, the bicoherence magnitude reduces with an increasing reduction in network connectivity, and begins to appear more like the human speakers in the first row of Figure 5.3. At the same time, there is little impact on the bicoherence phase, most likely because our network manipulation did not remove all of the long-range connections. Although this does not prove that the network architecture is solely responsible for the increased bicoherence properties, it provides preliminary evidence to suggest that this is the case. We note that the artifacts from Apple are more subdued than others. This may be related to the fact that Apple’s quality of speech is significantly less realistic than Google and Amazon, possibly because the underlying technique is not based on the same type of network architecture that we believe is introducing the polyspectral correlations. Regardless of precisely why these correlations are introduced, we next show that the bicoherence differences can be used to automatically distinguish between human and synthesized speeches.

5.2.4 Bispectral Classification

The bicohernece, Equation (5.7), is computed for each human and synthesized speech, from which the bicoherence magnitude and phase are computed. These two-dimensional quantities are normalized such that the magnitude and phase for each frequency ω_1 are normalized into the range $[0, 1]$ by subtracting the minimum value and dividing by the resulting maximum value.

The normalized magnitude and phase are each characterized using the first four statistical moments. Let the random variable M and P denote the underlying distribution for the bicoherence magnitude and phase. The first four statistical moments are given by:

- mean, $\mu_X = E_X[X]$
- variance, $\sigma_X = E_X[(X - \mu_X)^2]$
- skewness, $\gamma_X = E_X\left[\left(\frac{X-\mu_X}{\sigma_X}\right)^3\right]$
- kurtosis, $\omega = E_X\left[\left(\frac{X-\mu_X}{\sigma_X}\right)^4\right]$

where $E_X[\cdot]$ is the expected-value operator with regards to random variable X. From the magnitude

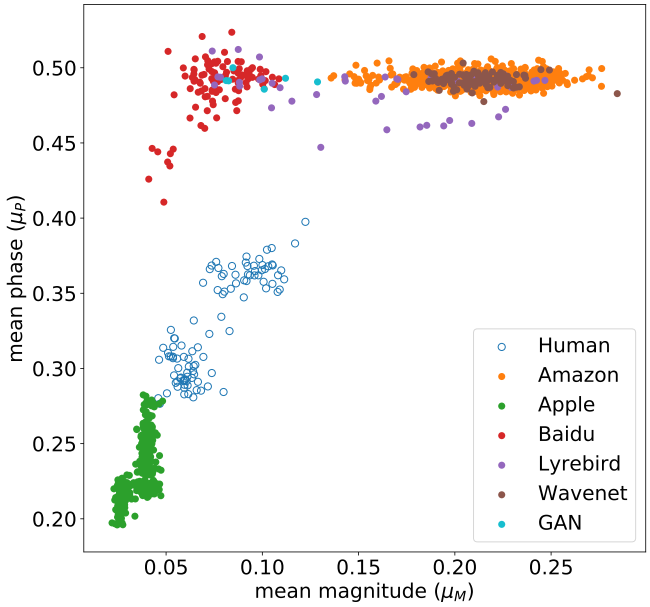


Figure 5.2: A 2-D slice of the full 8-D statistical characterization of the bicoherence magnitude and phase. The open blue circles correspond to human speech and the remaining filled colored circles correspond to synthesized speech. Even in this reduced dimensional space, the human speech is clearly distinct from the synthesized speech.

$X = M$ and phase $X = P$, these four moments are estimated by replacing the expected-value operator with an average. With this statistical characterization, each recording is reduced to an 8-D feature vector.

Shown in Figure 5.2 is a scatter plot of the mean bicoherence magnitude versus the mean bicoherence phase for the human speech and each type of synthesized speech. This figure illustrates some interesting aspects of the bicoherence statistics of the human and synthesized recordings. Even in this reduced-dimensional space that does not account for variance, skewness, or kurtosis, each type of signal is well clustered and (with the exception of Amazon and WaveNet) distinct from the other types. This suggests that it will be relatively straight-forward to distinguish between these different recordings.

Also shown in Figure 5.2 are six speech samples synthesized with a more recent generative adversary network (GAN) based model [65][3]. Although the GAN-based model has a different synthesis mechanism, the synthesized contents still exhibit distinct bispectral statistics.

The scatter plot in Figure 5.2 suggests two possible approaches to building a classifier. A

[3] There is no code publicly available and the six samples were downloaded from fangfm.github.io/crosslingualvc.html.

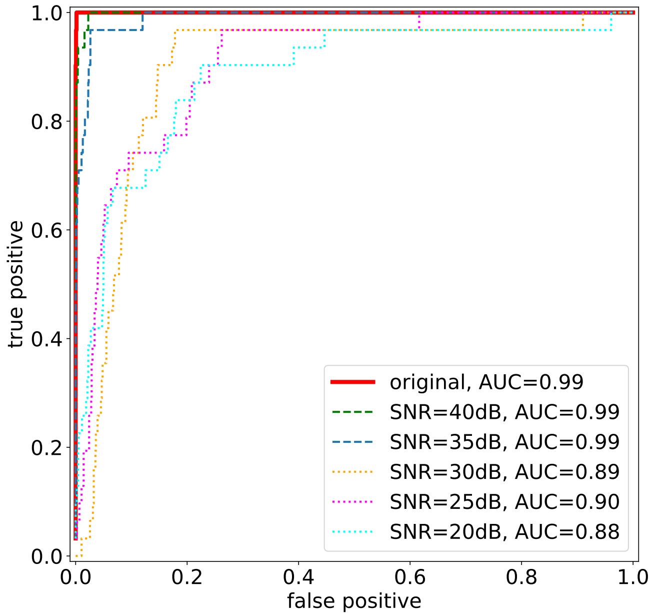
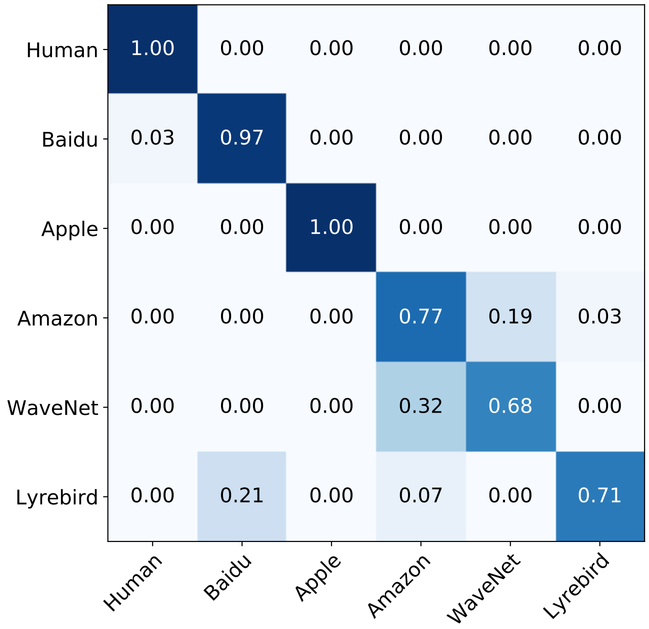


Figure 5.3: ROC curve for binary classification of human versus synthetic speech (solid red line). The dashed and dotted lines correspond to the accuracies for these same recordings with varying amounts of additive noise. See also Figure 5.4.

one-class non-linear support vector machine (SVM) or a collection of linear classifiers. We, primarily for simplicity, choose the latter. In particular, we train a linear classifier to distinguish each category of recording – human, Amazon, Apple, Baidu, Google, and Lyrebird – from all other recordings. Following this strategy, five separate logistic regression classifiers are trained to distinguish each synthesized audio from all other categories. For example, the first classifier is trained to distinguish Amazon recordings from Apple, Baidu, Google, Lyrebird, and human recordings. Our full data set consists of 100 human recordings, and 800 Amazon (8 speaker profiles), 400 Apple (4 speaker profiles), 100 Baidu (1 speaker profile), 400 Google (4 speaker profiles), and 45 Lyrebird recordings (5 recordings for each of 9 speaker profiles). Because of the across class imbalance, the training data set consisted of 70% of these samples with a maximum of 90 samples per category, with the remaining data used for testing.

The logistic regression classifier is implemented using `scikit-learn`[4]. At testing, a speech sample is classified by each classifier (Amazon, Apple, Baidu, Google, and Lyrebird). If the maximum classification score across all five classifiers is above a specified threshold, then the recording is classified as synthesized, otherwise it is classified as human.

[4] scikit-learn.org



	Human	Baidu	Apple	Amazon	WaveNet	Lyrebird
Human	1.00	0.00	0.00	0.00	0.00	0.00
Baidu	0.03	0.97	0.00	0.00	0.00	0.00
Apple	0.00	0.00	1.00	0.00	0.00	0.00
Amazon	0.00	0.00	0.00	0.77	0.19	0.03
WaveNet	0.00	0.00	0.00	0.32	0.68	0.00
Lyrebird	0.00	0.21	0.00	0.07	0.00	0.71

Figure 5.4: Confusion matrix for classifying a recording as human or as synthesized by one of five techniques. See also Figure 5.3.

5.3 Results

We test the performance of distinguishing human speech from synthesized speech based on the 8-D summary bicoherence statistics. Shown in Figure 5.3 are the receiver operator characteristic (ROC) curves for this binary classification. The solid curve with an area under the curve (AUC) of 0.99 corresponds to the original quality recordings. The remaining dashed/dotted colored curves correspond to the recordings that were laundered with varying amounts of additive noise (with a signal-to-noise ratio (SNR) between 20 and 40 dB) followed by re-compression at a quality of 128 kilobits per second (kbit/s). At high SNR, the AUC remains above 0.98, and the AUC decreases with increasing amounts of additive noise.

When the original recordings are recompressed at a lower quality of 64 kbit/s, the overall AUC remains high at 0.99 suggesting that the bispectral statistics are robust to recompression.

Shown in Figure 5.4 is the confusion matrix for the multi-class classification showing that the differences in bicoherence statistics are sufficient not only to distinguish human from synthesized speeches but also, with a reasonable degree of accuracy, to distinguish between different types of synthesized speech.

CHAPTER 6

A Study of Neural Vocoder Artifacts for Audio Forensic Analysis

6.1 Introduction

Recent years have seen a proliferation of AI-synthesized media, more commonly known as "deepfakes" [72]. Deepfake images, audios, and videos have reached a level of quality that challenges human ability to distinguish them from real media. While AI-synthesized imagery receives much research and media attention, techniques for generating synthetic audio are also emerging with unprecedented quality and generation efficiency. Recent incidents in which deepfake audios were used by scammers for financial gains [20] highlight the need to develop detection methods of AI-synthesized audios. While these techniques are capable of producing high-quality deepfakes, we hypothesize that unique artifacts of the generation process are present in deepfake outputs. Most of these artifacts are not audible, which increases the difficulty of differentiating between real and synthesized speech samples.

In this work, we develop a new method to detect synthetic audio by identifying the neural vocoders used in the generation process. The neural vocoder is a core component of most deepfake audio synthesis algorithms. It is a specially designed deep neural network that synthesizes audio waveforms from temporal-frequency representations, e.g., mel spectrograms. Recent years have seen active development of vocoders, improving training efficiency and synthesis quality. The generated waveforms from a neural vocoder contain artifacts that often are not audible. But the existence of such artifacts raises reasonable suspicion about the origin of the audio, as such traces are highly unlikely to be present in real media. Although one cannot claim with certainty that an audio sample was AI-generated using vocoder detection alone, it may provide important indirect evidence of audio tampering.[1]

In this work, we develop the first method to solve the vocoder identification problem. Our solution is based on the RawNet2 model [102], which works with audio waveforms directly and can therefore detect subtle vocoder artifacts that may be distorted or eliminated by additional pro-

[1] This form of indirect but definite evidence is analogous to identifying image manipulation by detecting double JPEG compression [82].

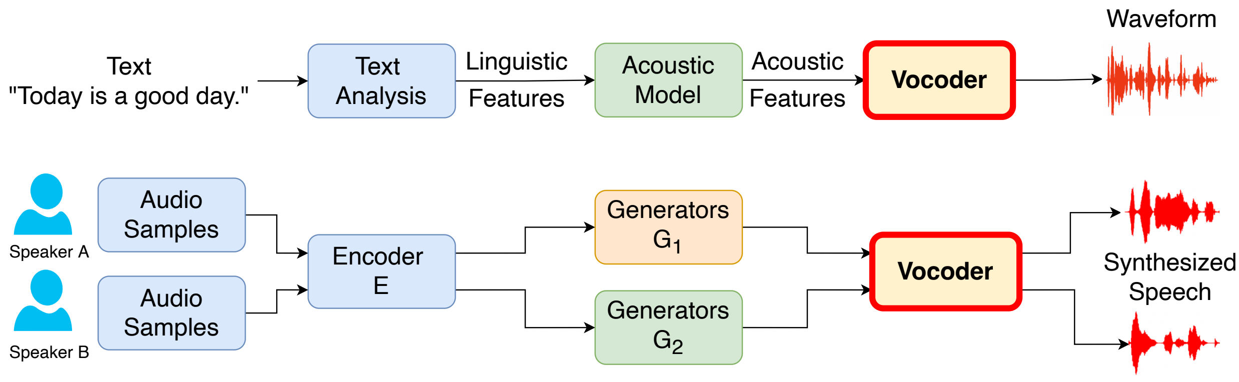


Figure 6.1: Overall pipeline for deepfake audio synthesis, including (top) text-to-speech and (bottom) voice conversion. Note that the vocoder is the common component in both types of methods.

cessing, such as mel spectrogram calculation. As vocoder identification is a previously unexplored problem, there are no pre-existing datasets that can be used to evaluate performance of an identification algorithm. Therefore we also make available a large-scale dataset, referred to as *LibriVoc*, which includes a total of 118.08 hours of synthesized samples derived from human audio samples from the LibriTTS dataset [133]. To isolate the effect of each neural vocoder on the generation process, the samples in our dataset are created by taking human voice audio waveforms as input and converting them into mel spectrogram representations, then generating waveform samples from those representations using six state-of-the-art neural vocoders. Because these "self-vocoding" samples are sourced from the same original audio signals, we can use them to highlight the artifacts of different vocoders, which can then be identified by a trained detector. The final detection algorithm is based on the identification of these artifacts, formulated as a binary or multi-class classification problem. Our experiments show that the RawNet2 model trained on LibriVoc achieves an overall vocoder identification Equal Error Rate (EER) of 1.61% when considering all types of vocoders. Our experiments also reveal that there is an increased difficulty in identifying individual vocoders.

The main contributions of our work are as follows:

- We propose a new approach to detect AI-synthesized audios based on exposing subtle artifacts left by neural vocoders.
- We propose a baseline algorithm for the unexplored problem of vocoder identification using the RawNet2 model.

- We provide a large-scale dataset that includes samples generated from six different vocoder algorithms. To the best of our knowledge, this is the only dataset generated for the purpose of vocoder identification. We will make this dataset available to the research community.

6.2 Related Works

6.2.1 Neural Network-Based Audio Synthesis

There are two general categories of audio synthesis techniques, as shown in Figure 6.1.

Text-to-speech (TTS) models convert an input text to audio with the target voice. Notable examples of TTS systems include WaveNet [112], Tacotron [118], Tacotron 2 [121], ClariNet [80], and FastSpeech 2s [87]. TTS systems consist of three essential components: a text analysis module, an acoustic model, and a vocoder. The text analysis module transforms a text sequence into linguistic features. The acoustic model then generates acoustic features from the linguistic features. Finally, the vocoder synthesizes a waveform from the acoustic features.

Voice conversion (VC) represents the second category of audio synthesis. Neural network-based VC models take advantage of the strong capability of neural networks to generate accurate mappings between the spectral features of one speaker's voice to that of another speaker. Variational Auto-Encoder (VAE) or Generative Adversarial Network (GAN) models are often used to capture the spectral features of the voice of a speaker, then perform style transfer over a mel spectrogram, and finally use a vocoder to reconstruct and output an audio waveform. Notable examples of neural VC methods include [5, 13, 73].

6.2.2 Neural Vocoders

Common to most existing neural network-based TTS and VC systems is the neural vocoder, which is a specially designed deep neural network that synthesizes audio waveforms from temporal-frequency representations. Existing vocoders can be divided into three main categories: autoregressive models, GAN-based models, and diffusion models.

Autoregressive models are probabilistic models that predict the distribution of each waveform sample based on all previous samples. However, since this process undergoes a linear sample-by-sample generation, the output speed of the model is slower than that of other methods.

WaveNet [112] is the first autoregressive neural vocoder. WaveRNN [44] is another autoregressive vocoder that uses a single-layer recurrent neural network for audio generation, which is designed to efficiently predict 16-bit raw audio samples.

GAN-based models employ a generator to model the waveform signal in the time domain and a discriminator to estimate the quality of the generated speech. The two most commonly used GAN-based neural vocoders are Mel-GAN [56] and Parallel WaveGAN [129]. GAN-based vocoders have demonstrated extraordinary performance in recent works. They have been shown to outperform autoregressive and diffusion models in both generation speed and generation quality.

Diffusion models are probabilistic generative models, which run *diffusion* and *reverse* as two main processes [37]. The diffusion process is characterized by a Markov chain, which gradually adds Gaussian noise to an original signal until that noise is eliminated. The reverse process is a de-noising stage that steadily removes the added Gaussian noise and converts a sample back to the original signal. There are two diffusion-based vocoders commonly represented in audio research: WaveGrad [15] and DiffWave [54].

6.2.3 Synthetic audio detection

We next survey prior works aiming at detecting synthesized audios. The ASVspoof Challenge 2021 focused on detecting synthetic audios and introduced four primary baseline algorithms: the Gaussian mixture models CQCC-GMM [105] and LFCC-GMM [105], a light convolutional neural network model LFCC-LCNN [105], and RawNet2 [102]. RawNet2 is the most recently introduced baseline, which yielded the second-best system results as reported from the complete ASVspoof 2019 logical access evaluation condition. The bi-spectral analysis algorithm [6] represents another of the first detection methods for deepfake audios. It utilizes spectral correlations estimated using bi-spectral analysis tools to differentiate human from synthesized speech. More recently, the DeepSonar [117] leverages network responses of audio signals as the feature to detect synthetic audios. For works comparing different neural vocoders, the work of [25] compares a few neural vocoders for speech reconstruction on a small set of input audio signals. However, to the best of our knowledge, there exists no large-scale dataset for the task of vocoder identification and synthetic audio detection. The lack of such a dataset is a critical bottleneck for developing vocoder-based audio deepfake detection methods.

Table 6.1: The number of hours of audio synthesized by each neural vocoder.

Model	train-clean-100	train-clean-360	dev-clean	test-clean
WaveNet (A01)	4.28	15.49	0.75	0.76
WaveRNN (A02)	4.33	14.92	0.67	0.72
MelGAN (G01)	4.36	15.26	0.71	0.76
Parallel WaveGAN (G02)	4.37	15.54	0.68	0.75
WaveGrad (D01)	4.19	15.81	0.76	0.74
DiffWave (D02)	4.16	15.37	0.62	0.66
Total	25.69	92.39	4.19	4.39

6.3 The LibriVoc Dataset

We create LibriVoc as a new open-source, large-scale dataset for the study of neural vocoder artifact detection. LibriVoc is derived from the LibriTTS speech corpus [133]. The LibriTTS corpus [133] itself is derived from the Librispeech dataset [79], wherein each sample is extracted from LibriVox audiobooks.[1] LibriTTS contains 585 hours of recorded speech samples from $2,456$ speakers. An average of 25 minutes of speech samples is available for each speaker. Due to the organization structure and the alignment between each speech sample and the corresponding text, the LibriTTS corpus has been widely used in text-to-speech research [52, 110, 14].

We use the state-of-the-art neural vocoders to generate synthesized speech samples in the LibriVoc dataset. Specifically, we employed six neural vocoders representing each vocoder category as detailed in Section 6.2.2: namely, WaveNet and WaveRNN from the autoregressive vocoders, Mel-GAN and Parallel WaveGAN from the GAN-based vocoders, and WaveGrad and DiffWave from the diffusion-based vocoders. Throughout the rest of this paper, we will respectively refer to the two **autoregressive models** WaveNet and WaveRNN as A01 and A02, the two **GAN-based models** Mel-GAN and Parallel WaveGAN as G01 and G02, and the two **diffusion models** WaveGrad and DiffWave as D01 and D02. Original samples will be referred to as OGA; see Table 6.1 and Figures 6.2 and 6.3. WaveNet (A01) is the most commonly used neural vocoder among the six studied in this work. However, we found Parallel WaveGAN (G02) to be the best performing neural vocoder in terms of audio quality and processing speed.

Each vocoder is trained to synthesize waveform samples from a given mel spectrogram ex-

[1] https://librivox.org/

tracted from an original sample; this process is referred to as "self-vocoding." By providing each vocoder with the same mel spectrogram, we ensure that any unique artifacts present in the synthesized samples are attributable to the specific vocoder used to reconstruct the audio signal. We withhold a set of real samples to use as a validation set in the training process. By doing so, we also ensure that input samples will always be new to the vocoder, regardless of the training split. We fix the *sampling rate* across all audio samples to 24 kHz. We rely on AlBadawy *et al.* [3] for their open-source implementation of spectrogram computation, vocoder training, and synthesized sample generation.

To ensure that our classifier does not overfit to speaker identity during the training process, we design the LibriVoc dataset as follows:

- Samples corresponding to 25% of the speakers contain only real (original) samples.
- Samples corresponding to 25% of the speakers contain only synthesized samples.
- For each speaker in the remaining 50%, we allocate half of the samples from that speaker to be real and the other half to be synthesized.

The choice of each speaker and their samples is performed at random. This configuration is applied to each dataset split, namely the training set, development set, and test set. Half of the samples obtained from this configuration are reserved as real samples, and the other half are kept as synthesized samples. Specifically, we have 126.41 hours of real samples and 118.08 hours of synthesized, self-vocoded samples in the training set. Table 6.1 shows a breakdown of the number of hours of synthesized samples allocated to each neural vocoder in our experiments.

6.4 Vocoder Detection

Our vocoder detection method is based on the recent RawNet2 model [102]. RawNet2 is an end-to-end model that was originally designed for the automatic speaker verification anti-spoofing task. It ranks among the best-performing baselines in the ASVspoof challenge [127]. The main reason for choosing RawNet2 as our main classifier is that RawNet2 was designed to work directly on raw waveforms. This helps by reducing any possible information loss associated with neural vocoder artifacts, as compared to working with pre-processed features *e.g.*, mel spectrograms or linear frequency cepstral coefficients (LFCCs).

The RawNet2 model consists of three main components: fixed sink filters, a residual network, and a gated recurrent unit (GRU). We use the same model architecture that was originally proposed in [102]. We retrain the model on the LibriVoc training set and modify the number of predicted labels based on the experiments we conducted.

Let x_t be a waveform in the time domain. We assume that the self-vocoding process performed on x_t yields generation artifacts $\gamma_{t|v}$ where v is the neural vocoder employed in this process. The training samples will be $y_t = x_t + \gamma_{t|v}$ for synthesized waveforms and $y_t = x_t$ for the original waveforms. The RawNet2 classifier is trained on y_t waveforms to indirectly look for the existence of the neural vocoder artifacts $\gamma_{t|v}$.

Data augmentation. To strengthen the generalizability of the classifier, we conduct a series of offline augmentation procedures on each input speech segment. Note that these augmentations are performed *after* self-vocoding for the synthetic samples. Specifically, we perform two augmentation steps. First, we resample the input speech to an intermediate sampling rate, and then resample back to the original sampling rate (24 kHz). We hypothesize that this helps our work generalize across any possible sampling rate. The choices for our intermediate sampling rates are 8kHz, 16kHz, 22.05kHz, 32kHz, and 44.1kHz. In our second augmentation step, we add background noise across one of three SNR values (8, 10, and 20). For simplicity, the noise was drawn from a single pre-recorded sample of crowd noise. The probabilities of choosing between the original, re-sampled, or noisy speech segments are 40%, 40% and 20% respectively.

6.5 Evaluation Results

To evaluate the ability of our classifier in detecting neural vocoder artifacts, we designed the following three experiments with different configurations.

Synthesized audio detection. In this experiment, we trained the RawNet2 classifier to predict whether a given sample y_t is synthesized using a vocoder. To solve this problem, the classifier must look for the existence of neural vocoder artifacts $\gamma_{t|v}$. As this is a binary classification problem, the type of neural vocoder v employed in the generation process is not used to penalize the discriminator in the loss function. The RawNet2 classifer achieved a 1.61% EER on the test-clean subset, and a 3.50% EER after applying the data augmentation procedures introduced in the previous section. Even though the artifacts of the neural vocoders are not audible, these results

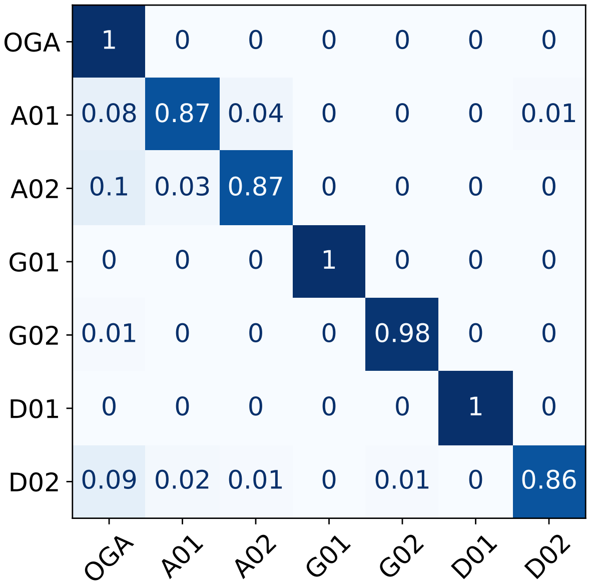
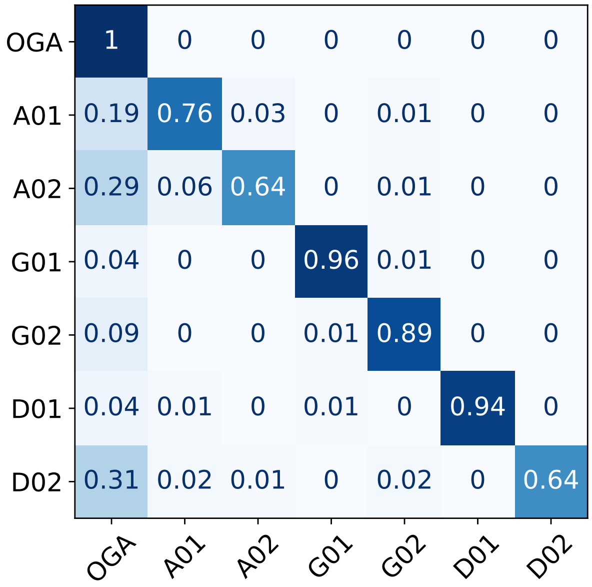


Figure 6.2: Confusion matrix for the classification of individual neural vocoders and the original samples on the test-clean subset.

demonstrate that a classifier can detect such artifacts with high accuracy.

Classification of neural vocoder artifacts. This experiment aimed to verify whether artifacts $\gamma_{t|v}$ generated by each neural vocoder v were unique to that vocoder. We altered the classification layer of RawNet2 to predict the seven different labels, wherein the first label is reserved for original samples and the other six for each of the neural vocoders we selected for our study. Figures 6.2 and 6.3 show the confusion matrices evaluated on the test-clean data subset from our second experiment with and without augmentation, respectively. The experiment yielded an EER of 3.15% when using augmentation and a 2.69% EER without augmentation. These results confirm two observations. First, the RawNet2 classifier can robustly detect vocoder artifacts even despite additive noise. Second, each neural vocoder v does produce unique artifacts $\gamma_{t|v}$, akin to a signature or vocoder fingerprint. As different vocoders produce different artifacts, we hypothesize that the inclusion of each neural vocoder at training time may be crucial for a classifier to reliably detect artifacts in the test set.

Leave N out cross-validation. To test the aforementioned hypothesis, we design the following experiment. We trained the RawNet2 as a binary classifier in a leave N out setting, where N represents the number of excluded vocoders (ranging from 0 to 5 excluded). For reliable results, we tested all possible combinations of which neural vocoder to be included in N. This resulted in



	OGA	A01	A02	G01	G02	D01	D02
OGA	1	0	0	0	0	0	0
A01	0.19	0.76	0.03	0	0.01	0	0
A02	0.29	0.06	0.64	0	0.01	0	0
G01	0.04	0	0	0.96	0.01	0	0
G02	0.09	0	0	0.01	0.89	0	0
D01	0.04	0.01	0	0.01	0	0.94	0
D02	0.31	0.02	0.01	0	0.02	0	0.64

Figure 6.3: Confusion matrix for the classification of individual neural vocoders and the original samples on the augmented test-clean subset.

63 possible combinations to use for all N values. Figure 6.4 shows the EER value on the y-axis versus the number of vocoders included in the training set on the x-axis. The error bar reports the mean and and standard deviation of all possible combinations for the same N value on both the augmented and non-augmented test-clean subset. As shown in Figure 6.4, both the mean and standard deviation of EER decrease as more vocoders are added to the training set. We also observed that the effect of augmentation on overall performance is more noticeable on experiments with low EER values, as compared to those with higher ones. These results confirm our hypothesis that using fewer vocoders in the training set reduces the efficacy of the RawNet2 classifier, when detecting artifacts from unseen vocoders.

6.6 Conclusion

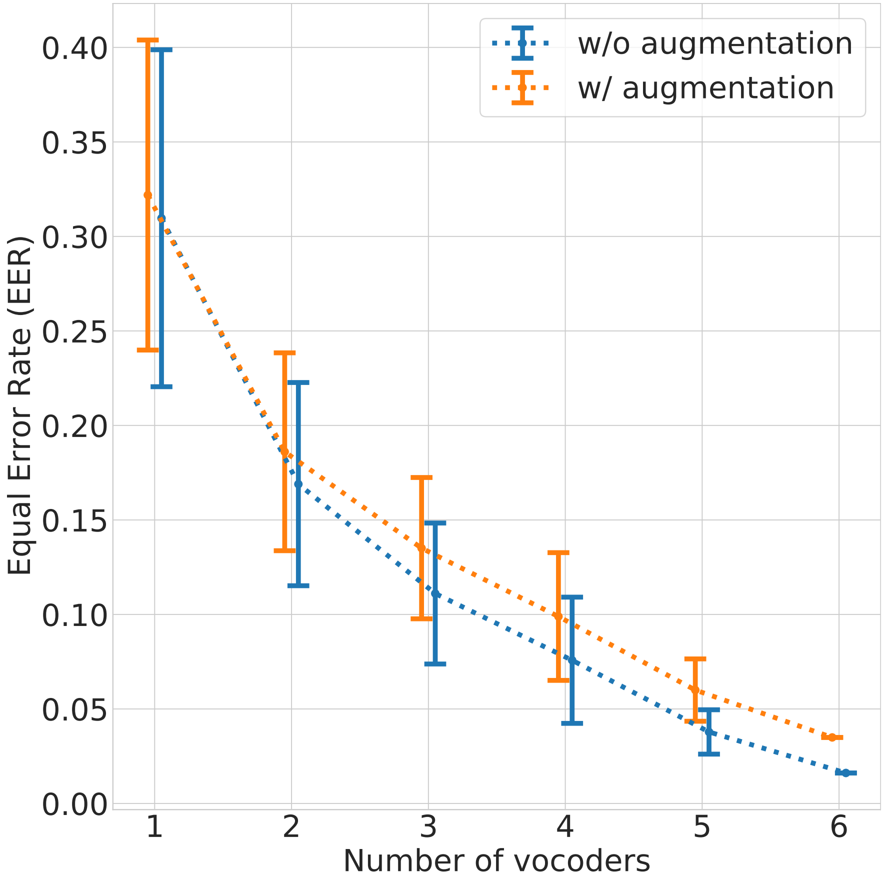


Figure 6.4: Mean Equal Error Rate (EER) for detecting vocoded speech on the test-clean subset, for different models trained with different numbers of vocoders. X-axis represent the number of vocoders used in the training set. Error bars represent the standard error for the different combinations of the selected vocoders. Blue and orange curves show the EER on the non-augmented and augmented test-clean subsets, respectively.

CHAPTER 7

Conclusion

Through this dissertation, we have explored the use of deep learning methods on speech signals. In the emotion recognition problem, we proposed joint modeling methods that combine discrete (classification, *clf*) and continuous (regression, *reg*) emotion prediction. Recent studies have found improvement in continuous emotion prediction performance when noisy continuous labels are quantized and these quantized, discrete labels are used for training classification models. However, although effective, the previous quantization approach may introduce a quantization error when converting the continuous labels to discrete ones. To overcome this challenge, we introduced two joint modeling methods: ensemble and end-to-end, using the state-of-the-art deep BLSTM architectures. The ensemble method combines the outputs from the *clf* and *reg* models at the prediction level. The end-to-end model optimizes both *clf*, *reg*, and the combination tasks at the same time, end-to-end.

To evaluate our method, we compared our proposed joint modeling with the state-of-the-art individual modeling baseline. The results show that the ensemble and end-to-end joint modeling methods outperform the individual models, where the end-to-end model has CCC value of 0.868 on the development set and 0.697 on the test set for arousal and the ensemble model 0.601 and 0.555 for valence. These results are higher than the best CCC of the baseline model, which achieved 0.863 on the development set and 0.686 on the test set for arousal and 0.595 and 0.544 for valence. Our joint modeling has new state-of-the-art results on both the development and test set (0.868 and 0.697 for arousal, 0.601 and 0.555 for valence) in comparison to previous state-of-the-art results by khorram et al. [49], where they received 0.867 and 0.684 for the development and test sets respectively on arousal, and 0.592 and 0.502 on valence.

Moreover, we introduced a novel approach for the voice conversion problem based on a neural style transfer model of the mel-spectrograms. Our method takes advantage of the recent developments in neural network models for image style transfer. Experimental results show that our method can faithfully transfer styles across different speakers while preserving the content of the original speech. In future work, we will further explore the possible modification of our proposed model to generalize to broad samples with noise in the background as well as cross-linguistic speech style transfer.

For a reliable evaluation of different neural vocoders' performance, We presented VocBench, a framework for a general-purpose benchmark of neural vocoders on the speech synthesis task. VocBench provides the speech community with a standard and comprehensive approach for neural vocoders evaluation. Our study includes results of both the objective and subjective differences for the vocoders. We have open-sourced our toolkit for training and evaluating neural vocoders on GitHub. We welcome the community to contribute and share their implementations and evaluations against SOTA vocoders.

We investigated the possible different approaches to detect AI-synthesized speech signals. More specifically, we have developed a forensic technique that can distinguish humans from synthesized speech. This technique is based on the observation that current speech-synthesis algorithms introduce specific and unusual higher-order bispectral correlations that are not typically found in human speech. We have provided preliminary evidence that these correlations are the result of the long-range correlations introduced by the underlying network architectures used to synthesize speech. This bodes well for us in the forensic community as it appears that these network architectures are also what is giving rise to more realistic-sounding speech (despite the unusual bispectral correlations). More work, however, remains to be done to more precisely understand the specific source of the unusual bispectral correlations.

As with any forensic technique, thought must be given to counter-measures that our adversary might adopt. While it would be straightforward to match first-order spectral correlations between human and synthesized speech, the higher-order spectral correlations are not so easily matched. In particular, we know of no closed-form solution for inverting the bispectrum or bicoherence. It remains to be seen if other techniques like generative adversarial networks can synthesize audio while matching the bispectral artifacts that currently can be used to distinguish human from synthesized speech.

Finally, We developed a model for vocoder identification based on the RawNet2 model. We also provided a large-scale dataset named LibriVoc, with synthetic audios of human voice samples created with a diverse set of neural vocoders. Experiments on this dataset show that our method can achieve an overall vocoder identification EER of 1.61%. There is still room for improvement in this work. We will consider a few extensions as future work. First, we would like to augment the LibriVoc dataset to include more diverse real audio signals and environments. Second, there are more neural vocoders developed in recent years, and it is important to continue augmenting the model zoo to keep pace with the latest development. Third, we will further explore more tailored solutions to the vocoder identification problem. Last, identification of vocoders is only indirect evidence of voice synthesis. It is our interest to further develop effective methods that can directly differentiate real and synthetic audios by combining cues from vocoders and other signal features.

BIBLIOGRAPHY

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al., *Tensorflow: A system for large-scale machine learning.*, OSDI, vol. 16, 2016, pp. 265–283.

[2] Manu Airaksinen, Lauri Juvela, Bajibabu Bollepalli, et al., *A comparison between straight, glottal, and sinusoidal vocoding in statistical parametric speech synthesis*, IEEE Trans Audio Speech Lang Process (2018).

[3] Ehab A AlBadawy, Andrew Gibiansky, Qing He, Jilong Wu, Ming-Ching Chang, and Siwei Lyu, *VocBench: A neural vocoder benchmark for speech synthesis*, arXiv preprint arXiv:2112.03099 (2021).

[4] Ehab A. AlBadawy and Siwei Lyu, *Voice conversion using speech-to-speech neuro-style transfer*, Proc. Interspeech, 2020, pp. 4726–4730.

[5] Ehab A. AlBadawy and Siwei Lyu, *Voice conversion using speech-to-speech neuro-style transfer*, Proc. Interspeech 2017 (2020).

[6] Ehab A AlBadawy, Siwei Lyu, and Hany Farid, *Detecting AI-synthesized speech using bispectral analysis.*, CVPR Workshops, 2019, pp. 104–109.

[7] Gopala Krishna Anumanchipalli, Ying-Chang Cheng, Joseph Fernandez, et al., *KLATT-STAT: Knowledge-based parametric speech synthesis*, Seventh ISCA Workshop, 2010.

[8] Fadi Biadsy, Ron J Weiss, Pedro J Moreno, Dimitri Kanvesky, and Ye Jia, *Parrotron: An end-to-end speech-to-speech conversion model and its applications to hearing-impaired speech and speech separation*, Proc. Interspeech 2019 (2019), 4115–4119.

[9] Kevin Brady, Youngjune Gwon, Pooya Khorrami, Elizabeth Godoy, William Campbell, Charlie Dagli, and Thomas S Huang, *Multi-modal audio, video and physiological sensor learning for continuous emotion prediction*, Proceedings of the 6th International Workshop on Audio/Visual Emotion Challenge, ACM, 2016, pp. 97–104.

[10] Carlos Busso, Murtaza Bulut, Chi-Chun Lee, Abe Kazemzadeh, Emily Mower, Samuel Kim, Jeannette N Chang, Sungbok Lee, and Shrikanth S Narayanan, *Iemocap: Interactive emotional dyadic motion capture database*, Language resources and evaluation **42** (2008), no. 4, 335.

[11] Caroline Chan, Shiry Ginosar, Tinghui Zhou, and Alexei A Efros, *Everybody dance now*, arXiv preprint arXiv:1808.07371 (2018).

[12] Linlin Chao, Jianhua Tao, Minghao Yang, Ya Li, and Zhengqi Wen, *Long short term memory recurrent neural network based multimodal dimensional emotion recognition*, Proceedings of the 5th International Workshop on Audio/Visual Emotion Challenge, ACM, 2015, pp. 65–72.

[13] Ling-Hui Chen, Zhen-Hua Ling, Li-Juan Liu, and Li-Rong Dai, *Voice conversion using deep neural networks with layer-wise generative training*, IEEE/ACM Transactions on Audio, Speech, and Language Processing **22** (2014), no. 12, 1859–1872.

[14] Mingjian Chen, Xu Tan, Yi Ren, Jin Xu, Hao Sun, Sheng Zhao, Tao Qin, and Tie-Yan Liu, *Multispeech: Multi-speaker text to speech with transformer*, arXiv preprint arXiv:2006.04664 (2020).

[15] Nanxin Chen, Yu Zhang, Heiga Zen, et al., *WaveGrad: Estimating gradients for waveform generation*, ICLR, 2020.

[16] Ira Cohen, Ashutosh Garg, Thomas S Huang, et al., *Emotion recognition from facial expressions using multilevel hmm*, Neural information processing systems, vol. 2, Citeseer, 2000.

[17] Li Deng and John C Platt, *Ensemble deep learning for speech recognition*, Fifteenth Annual Conference of the International Speech Communication Association, 2014.

[18] J.W.A. Fackrell and Stephen McLaughlin, *Detecting nonlinearities in speech sounds using the bicoherence*, Proceedings of the Institute of Acoustics **18** (1996), no. 9, 123–130.

[19] Hany Farid, *Detecting digital forgeries using bispectral analysis*, Tech. Report AI Memo 1657, MIT, June 1999.

[20] Forbes, *A Voice Deepfake Was Used To Scam A CEO Out Of $243,000*, https://www.cnn.com/2020/02/20/tech/fake-faces-deepfake/index.html, 11 2019.

[21] Benoît Frénay and Michel Verleysen, *Classification in the presence of label noise: a survey*, IEEE transactions on neural networks and learning systems **25** (2014), no. 5, 845–869.

[22] Sayan Ghosh, Eugene Laksana, Louis-Philippe Morency, and Stefan Scherer, *Representation learning for speech emotion recognition.*, INTERSPEECH, 2016, pp. 3603–3607.

[23] Xavier Glorot and Yoshua Bengio, *Understanding the difficulty of training deep feedforward neural networks*, Proceedings of the thirteenth international conference on artificial intelligence and statistics, 2010, pp. 249–256.

[24] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, et al., *Generative adversarial nets*, NeurIPS **27** (2014).

[25] Prachi Govalkar, Johannes Fischer, Frank Zalkow, and Christian Dittmar, *A Comparison of Recent Neural Vocoders for Speech Signal Reconstruction*, Proc. 10th ISCA Workshop on Speech Synthesis (SSW 10), 2019, pp. 7–12.

[26] Prachi Govalkar, Johannes Fischer, Frank Zalkow, et al., *A comparison of recent neural vocoders for speech signal reconstruction*, ISCA, 2019.

[27] Anirudh Goyal, Nan Rosemary Ke, Surya Ganguli, and Yoshua Bengio, *Variational walkback: Learning a transition operator as a stochastic recurrent net*, NeurIPS (2017).

[28] Daniel Griffin and Jae Lim, *Signal estimation from modified short-time Fourier transform*, IEEE Transactions on acoustics, speech, and signal processing (1984).

[29] Florin Grigoras, Horia-Nicolai Teodorescu, Lakhmi C Jain, and Vasile Apopei, *Fuzzy and knowledge-based control for speech synthesis*, ECC, 1999.

[30] Yu Gu and Yongguo Kang, *Multi-task WaveNet: A multi-task generative model for statistical parametric speech synthesis without fundamental frequency conditions*, Interspeech (Hyderabad, India), 2018.

[31] Hatice Gunes and Björn Schuller, *Categorical and dimensional affect analysis in continuous input: Current trends and future directions*, Image and Vision Computing **31** (2013), no. 2, 120–136.

[32] Jing Han, Zixing Zhang, Maximilian Schmitt, Maja Pantic, and Björn Schuller, *From hard to soft: Towards more human-like emotion recognition by modelling the perception uncertainty*, Proceedings of the 2017 ACM on Multimedia Conference, ACM, 2017, pp. 890–897.

[33] Mark Hasegawa-Johnson, Alan Black, Lucas Ondel, Odette Scharenborg, and Francesco Ciannella, *Image2speech: Automatically generating audio descriptions of images*, Proceedings of ICNLSSP, Casablanca, Morocco (2017).

[34] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, *Deep residual learning for image recognition*, Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[35] Lang He, Dongmei Jiang, Le Yang, Ercheng Pei, Peng Wu, and Hichem Sahli, *Multimodal affective dimension prediction using deep bidirectional long short-term memory recurrent neural networks*, Proceedings of the 5th International Workshop on Audio/Visual Emotion Challenge, ACM, 2015, pp. 73–80.

[36] Shawn Hershey, Sourish Chaudhuri, Daniel PW Ellis, et al., *CNN architectures for large-scale audio classification*, ICASSP, IEEE, 2017, pp. 131–135.

[37] Jonathan Ho, Ajay Jain, and Pieter Abbeel, *Denoising diffusion probabilistic models*, NeurIPS (2020).

[38] Sepp Hochreiter and Jürgen Schmidhuber, *Long short-term memory*, Neural computation **9** (1997), no. 8, 1735–1780.

[39] Zhengwei Huang, Ming Dong, Qirong Mao, and Yongzhao Zhan, *Speech emotion recognition using cnn*, Proceedings of the 22nd ACM international conference on Multimedia, ACM, 2014, pp. 801–804.

[40] Sergey Ioffe and Christian Szegedy, *Batch normalization: Accelerating deep network training by reducing internal covariate shift*, International Conference on Machine Learning, 2015, pp. 448–456.

[41] Keith Ito, *The lj speech dataset*, https://keithito.com/LJ-Speech-Dataset/, 2017.

[42] Ye Jia, Ron J Weiss, Fadi Biadsy, Wolfgang Macherey, Melvin Johnson, Zhifeng Chen, and Yonghui Wu, *Direct speech-to-speech translation with a sequence-to-sequence model*, arXiv preprint arXiv:1904.06037 (2019).

[43] Ye Jia, Yu Zhang, Ron Weiss, Quan Wang, Jonathan Shen, Fei Ren, Patrick Nguyen, Ruoming Pang, Ignacio Lopez Moreno, Yonghui Wu, et al., *Transfer learning from speaker verification to multispeaker text-to-speech synthesis*, Advances in neural information processing systems, 2018, pp. 4480–4490.

[44] Nal Kalchbrenner, Erich Elsen, Karen Simonyan, et al., *Efficient neural audio synthesis*, ICML, 2018.

[45] Hirokazu Kameoka, Takuhiro Kaneko, Kou Tanaka, and Nobukatsu Hojo, *StarGAN-VC: Non-parallel many-to-many voice conversion using star generative adversarial networks*, SLT Workshop, IEEE, 2018.

[46] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen, *Progressive growing of GANs for improved quality, stability, and variation*, arXiv preprint arXiv:1710.10196 (2017).

[47] Tero Karras, Samuli Laine, and Timo Aila, *A style-based generator architecture for generative adversarial networks*, arXiv preprint arXiv:1812.04948 (2018).

[48] Heysem Kaya, Furkan Gürpınar, and Albert Ali Salah, *Video-based emotion recognition in the wild using deep transfer learning and score fusion*, Image and Vision Computing **65** (2017), 66–75.

[49] Soheil Khorram, Zakaria Aldeneh, Dimitrios Dimitriadis, Melvin McInnis, and Emily Mower Provost, *Capturing long-term temporal dependencies with convolutional networks for continuous emotion recognition*, arXiv preprint arXiv:1708.07050 (2017).

[50] Kevin Kilgour, Mauricio Zuluaga, Dominik Roblek, and Matthew Sharifi, *Fréchet audio distance: A metric for evaluating music enhancement algorithms*, arXiv (2018).

[51] Hyeongwoo Kim, Pablo Garrido, Ayush Tewari, Weipeng Xu, Justus Thies, Matthias Nießner, Patrick Pérez, Christian Richardt, Michael Zollhöfer, and Christian Theobalt, *Deep Video Portraits*, ACM Transactions on Graphics (2018), no. 4, 163.

[52] Jaehyeon Kim, Sungwon Kim, Jungil Kong, and Sungroh Yoon, *Glow-TTS: A generative flow for text-to-speech via monotonic alignment search*, Advances in Neural Information Processing Systems **33** (2020), 8067–8077.

[53] Diederik P Kingma and Max Welling, *Auto-encoding variational bayes*, arXiv preprint arXiv:1312.6114 (2013).

[54] Zhifeng Kong, Wei Ping, Jiaji Huang, et al., *DiffWave: A versatile diffusion model for audio synthesis*, ICLR, 2020.

[55] Zvi Kons, Slava Shechtman, Alex Sorin, Carmel Rabinovitz, and Ron Hoory, *High quality, lightweight and adaptable tts using lpcnet*, Proc. Interspeech 2019 (2019), 176–180.

[56] Kundan Kumar, Rithesh Kumar, Thibault de Boissiere, et al., *MelGAN: Generative adversarial networks for conditional waveform synthesis*, arXiv (2019).

[57] I Lawrence and Kuei Lin, *A concordance correlation coefficient to evaluate reproducibility*, Biometrics (1989), 255–268.

[58] Duc Le, Zakaria Aldeneh, and Emily Mower Provost, *Discretized continuous speech emotion recognition with multi-task deep recurrent neural network*, Interspeech, 2017 (to apear) (2017).

[59] Gil Levi and Tal Hassner, *Emotion recognition in the wild via convolutional neural networks and mapped binary patterns*, Proceedings of the 2015 ACM on international conference on multimodal interaction, ACM, 2015, pp. 503–510.

[60] Naihan Li, Shujie Liu, Yanqing Liu, et al., *Neural speech synthesis with transformer network*, AAAI, 2019.

[61] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick, *Microsoft coco: Common objects in context*, European conference on computer vision, Springer, 2014, pp. 740–755.

[62] Ming-Yu Liu, Thomas Breuel, and Jan Kautz, *Unsupervised image-to-image translation networks*, Advances in neural information processing systems, 2017, pp. 700–708.

[63] Ming-Yu Liu and Oncel Tuzel, *Coupled generative adversarial networks*, Advances in neural information processing systems, 2016, pp. 469–477.

[64] Paula Lopez-Otero, Laura Docio-Fernandez, and Carmen Garcia-Mateo, *ivectors for continuous emotion recognition*, Training **45** (2014), 50.

[65] Jaime Lorenzo-Trueba, Fuming Fang, Xin Wang, Isao Echizen, Junichi Yamagishi, and Tomi Kinnunen, *Can we steal your vocal identity from the internet?: Initial investigation of cloning Obama's voice using GAN, WaveNet and low-quality found data*, The Speaker and Language Recognition Workshop (Odyssey) (Les Sables d'Olonne, France), 2018.

[66] Xingchen Ma, Hongyu Yang, Qiang Chen, Di Huang, and Yunhong Wang, *Depaudionet: An efficient deep model for audio based depression classification*, Proceedings of the 6th International Workshop on Audio/Visual Emotion Challenge, ACM, 2016, pp. 35–42.

[67] Qirong Mao, Ming Dong, Zhengwei Huang, and Yongzhao Zhan, *Learning salient features for speech emotion recognition using convolutional neural networks*, IEEE Transactions on Multimedia **16** (2014), no. 8, 2203–2213.

[68] Gary McKeown, Michel Valstar, Roddy Cowie, Maja Pantic, and Marc Schroder, *The semaine database: Annotated multimodal records of emotionally colored conversations between a person and a limited agent*, IEEE Transactions on Affective Computing **3** (2012), no. 1, 5–17.

[69] Jerry M. Mendel, *Tutorial on higher order statistics (spectra) in signal processing and system theory: theoretical results and some applications*, Proceedings of the IEEE **79** (1996), 278–305.

[70] Hongying Meng and Nadia Bianchi-Berthouze, *Affective state level recognition in naturalistic facial and vocal expressions*, IEEE Transactions on Cybernetics **44** (2014), no. 3, 315–328.

[71] Angeliki Metallinou, Martin Wöllmer, Athanasios Katsamanis, Florian Eyben, Björn Schuller, and Shrikanth Narayanan, *Context-sensitive learning for enhanced audiovisual emotion classification*, Affective Computing, IEEE Transactions on **3** (2012), no. 2, 184–198.

[72] Yisroel Mirsky and Wenke Lee, *The creation and detection of deepfake: A survey*, ACM Computing Surveys **54** (2021), no. 1.

[73] Seyed Hamidreza Mohammadi and Alexander Kain, *Voice conversion using deep neural networks with speaker-independent pre-training*, 2014 IEEE Spoken Language Technology Workshop (SLT), IEEE, 2014, pp. 19–23.

[74] Noam Mor, Lior Wolf, Adam Polyak, and Yaniv Taigman, *A universal music translation network*, International Conference on Learning Representations, 2018.

[75] Donn Morrison, Ruili Wang, and Liyanage C De Silva, *Ensemble methods for spoken emotion recognition in call-centres*, Speech communication **49** (2007), no. 2, 98–112.

[76] Koki Nagano, Jaewoo Seo, Jun Xing, Lingyu Wei, Zimo Li, Shunsuke Saito, Aviral Agarwal, Jens Fursund, Hao Li, Richard Roberts, et al., *paGAN: real-time avatars using dynamic textures*, SIGGRAPH Asia 2018 Technical Papers, ACM, 2018, p. 258.

[77] Tin Lay Nwe, Say Wei Foo, and Liyanage C De Silva, *Speech emotion recognition using hidden markov models*, Speech communication **41** (2003), no. 4, 603–623.

[78] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu, *Wavenet: A generative model for raw audio*, arXiv preprint arXiv:1609.03499 (2016).

[79] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, *LibriSpeech: an ASR corpus based on public domain audio books*, 2015 IEEE international conference on acoustics, speech and signal processing (ICASSP), IEEE, 2015, pp. 5206–5210.

[80] Wei Ping, Kainan Peng, and Jitong Chen, *Clarinet: Parallel wave generation in end-to-end text-to-speech*, arXiv preprint arXiv:1807.07281 (2018).

[81] Wei Ping, Kainan Peng, Andrew Gibiansky, Sercan O Arik, Ajay Kannan, Sharan Narang, Jonathan Raiman, and John Miller, *Deep voice 3: 2000-speaker neural text-to-speech*, arXiv preprint arXiv:1710.07654 (2017).

[82] Alin C. Popescu and Hany Farid, *Statistical tools for digital forensics*, In 6th International Workshop on Information Hiding, Springer-Verlag, Berlin-Heidelberg, 2004, pp. 128–147.

[83] Jonathan Posner, James A Russell, and Bradley S Peterson, *The circumplex model of affect: An integrative approach to affective neuroscience, cognitive development, and psychopathology*, Development and psychopathology **17** (2005), no. 3, 715–734.

[84] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al., *The kaldi speech recognition toolkit*, IEEE 2011 workshop on automatic speech recognition and understanding, no. EPFL-CONF-192584, IEEE Signal Processing Society, 2011.

[85] Filip Povolny, Pavel Matˇejka, Michal Hradis, Anna Popková, Lubomír Otrusina, Pavel Smrz, Ian Wood, Cecile Robin, and Lori Lamel, *Multimodal emotion recognition for avec 2016 challenge*, Proceedings of the 6th International Workshop on Audio/Visual Emotion Challenge, ACM, 2016, pp. 75–82.

[86] Cyrus Rashtchian, Peter Young, Micah Hodosh, and Julia Hockenmaier, *Collecting image annotations using amazon's mechanical turk*, Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk, Association for Computational Linguistics, 2010, pp. 139–147.

[87] Yi Ren, Chenxu Hu, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu, *FastSpeech 2: Fast and high-quality end-to-end text to speech*, arXiv preprint arXiv:2006.04558 (2020).

[88] Fabien Ringeval, Björn Schuller, Michel Valstar, Shashank Jaiswal, Erik Marchi, Denis Lalanne, Roddy Cowie, and Maja Pantic, *Av^{+}ec 2015: The first affect recognition challenge bridging across audio, video, and physiological data*, Proceedings of the 5th International Workshop on Audio/Visual Emotion Challenge, ACM, 2015, pp. 3–8.

[89] Fabien Ringeval, Andreas Sonderegger, Juergen Sauer, and Denis Lalanne, *Introducing the recola multimodal corpus of remote collaborative and affective interactions*, Automatic Face and Gesture Recognition (FG), 2013 10th IEEE International Conference and Workshops on, IEEE, 2013, pp. 1–8.

[90] Md Sahidullah, Tomi Kinnunen, and Cemal Hanilçi, *A comparison of features for synthetic speech detection*, (2015).

[91] Md Sahidullah, Tomo Kinnunen, and Cemal Hanilci, *A comparison of features for synthetic speech detection*, Interspeech (Dresden, Germany), 2015.

[92] Michael Schoeffler, Fabian-Robert Stöter, Bernd Edler, et al., *Towards the next generation of web-based experiments: A case study assessing basic audio quality following the ITU-R recommendation BS. 1534 (MUSHRA)*, 1st Web Audio Conference, 2015, pp. 1–6.

[93] Marc Schroder, Elisabetta Bevacqua, Roddy Cowie, Florian Eyben, Hatice Gunes, Dirk Heylen, Mark Ter Maat, Gary McKeown, Sathish Pammi, Maja Pantic, et al., *Building autonomous sensitive artificial listeners*, IEEE Transactions on Affective Computing **3** (2012), no. 2, 165–183.

[94] Marc Schröder, Sathish Pammi, Hatice Gunes, Maja Pantic, Michel F Valstar, Roddy Cowie, Gary McKeown, Dirk Heylen, Mark Ter Maat, Florian Eyben, et al., *Come and have an emotional workout with sensitive artificial listeners!*, Automatic Face & Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on, IEEE, 2011, pp. 646–646.

[95] Björn Schuller, Stephan Reiter, Ronald Muller, Marc Al-Hames, Manfred Lang, and Gerhard Rigoll, *Speaker independent speech emotion recognition by ensemble classification*, Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on, IEEE, 2005, pp. 864–867.

[96] Björn Schuller, Gerhard Rigoll, and Manfred Lang, *Hidden markov model-based speech emotion recognition*, Multimedia and Expo, 2003. ICME'03. Proceedings. 2003 International Conference on, vol. 1, IEEE, 2003, pp. I–401.

[97] Jonathan Shen, Ruoming Pang, Ron J Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, Rj Skerrv-Ryan, et al., *Natural tts synthesis by conditioning wavenet on mel spectrogram predictions*, 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2018, pp. 4779–4783.

[98] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli, *Deep unsupervised learning using nonequilibrium thermodynamics*, ICML, 2015.

[99] Mohammad Soleymani, Sander Koelstra, Ioannis Patras, and Thierry Pun, *Continuous emotion detection in response to music videos*, Automatic Face & Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on, IEEE, 2011, pp. 803–808.

[100] Yang Sun, Louis Ten Bosch, and Lou Boves, *Hybrid hmm/blstm-rnn for robust speech recognition*, International Conference on Text, Speech and Dialogue, Springer, 2010, pp. 400–407.

[101] Supasorn Suwajanakorn, Steven M Seitz, and Ira Kemelmacher-Shlizerman, *Synthesizing Obama: learning lip sync from audio*, ACM Transactions on Graphics **36** (2017), no. 4, 95.

[102] Hemlata Tak, Jose Patino, Massimiliano Todisco, Andreas Nautsch, Nicholas Evans, and Anthony Larcher, *End-to-end anti-spoofing with RawNet2*, ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2021, pp. 6369–6373.

[103] Mark Tatham, *An integrated knowledge base for speech synthesis and automatic speech recognition*, Journal of Phonetics **13** (1985), no. 2, 175–188.

[104] Justus Thies, Michael Zollhöfer, Christian Theobalt, Marc Stamminger, and Matthias Nießner, *Headon: Real-time reenactment of human portrait videos*, ACM Transactions on Graphics **36** (2018), no. 4, 95.

[105] Massimiliano Todisco, Xin Wang, Ville Vestman, Md Sahidullah, Héctor Delgado, Andreas Nautsch, Junichi Yamagishi, Nicholas Evans, Tomi Kinnunen, and Kong Aik Lee, *Asvspoof 2019: Future horizons in spoofed and fake audio detection*, arXiv preprint arXiv:1904.05441 (2019).

[106] Keiichi Tokuda, Takayoshi Yoshimura, Takashi Masuko, et al., *Speech parameter generation algorithms for HMM-based speech synthesis*, ICASSP, IEEE, 2000, pp. 1315–1318.

[107] George Trigeorgis, Fabien Ringeval, Raymond Brueckner, Erik Marchi, Mihalis A Nicolaou, Björn Schuller, and Stefanos Zafeiriou, *Adieu features? end-to-end speech emotion recognition using a deep convolutional recurrent network*, Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on, IEEE, 2016, pp. 5200–5204.

[108] Panagiotis Tzirakis, George Trigeorgis, Mihalis A Nicolaou, Björn W Schuller, and Stefanos Zafeiriou, *End-to-end multimodal emotion recognition using deep neural networks*, IEEE Journal of Selected Topics in Signal Processing **11** (2017), no. 8, 1301–1309.

[109] Jean-Marc Valin and Jan Skoglund, *LPCNet: Improving neural speech synthesis through linear prediction*, ICASSP, IEEE, 2019, pp. 5891–5895.

[110] Rafael Valle, Kevin Shih, Ryan Prenger, and Bryan Catanzaro, *Flowtron: an autoregressive flow-based generative network for text-to-speech synthesis*, arXiv preprint arXiv:2005.05957 (2020).

[111] Michel Valstar, Jonathan Gratch, Björn Schuller, Fabien Ringeval, Denis Lalanne, Mercedes Torres Torres, Stefan Scherer, Giota Stratou, Roddy Cowie, and Maja Pantic, *Avec 2016: Depression, mood, and emotion recognition workshop and challenge*, Proceedings of the 6th International Workshop on Audio/Visual Emotion Challenge, ACM, 2016, pp. 3–10.

[112] Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu, *Wavenet: A generative model for raw audio*, 9th ISCA Speech Synthesis Workshop, 2016, pp. 125–125.

[113] Laurens Van Der Maaten, *Barnes-hut-sne*, arXiv preprint arXiv:1301.3342 (2013).

[114] C. Veaux, J. Yamagishi, and Kirsten MacDonald, *CSTR VCTK corpus: English multi-speaker corpus for CSTR voice cloning toolkit*, https://datashare.ed.ac.uk/handle/10283/2651, 2017.

[115] Pascal Vincent, *A connection between score matching and denoising autoencoders*, Neural Comput (2011).

[116] Li Wan, Quan Wang, Alan Papir, and Ignacio Lopez Moreno, *Generalized end-to-end loss for speaker verification*, 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2018, pp. 4879–4883.

[117] Run Wang, Felix Juefei-Xu, Yihao Huang, Qing Guo, Xiaofei Xie, Lei Ma, and Yang Liu, *Deepsonar: Towards effective and robust detection of ai-synthesized fake voices*, Proceedings of the 28th ACM International Conference on Multimedia, 2020, pp. 1207–1216.

[118] Yuxuan Wang, R. J. Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J. Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, Quoc V. Le, Yannis Agiomyrgiannakis, Rob Clark, and Rif A. Saurous, *Tacotron: A fully end-to-end text-to-speech synthesis model*, CoRR **abs/1703.10135** (2017).

[119] Yuxuan Wang, RJ Skerry-Ryan, Daisy Stanton, et al., *Tacotron: Towards end-to-end speech synthesis*, arXiv (2017).

[120] Yuxuan Wang, RJ Skerry-Ryan, Ying Xiao, Daisy Stanton, Joel Shor, Eric Battenberg, Rob Clark, and Rif A Saurous, *Uncovering latent style factors for expressive speech synthesis*, ML4Audio Workshop, NIPS (2017b).

[121] Zhi Wang, Yinhua Liu, and Liang Shan, *CE-Tacotron2: End-to-end emotional speech synthesis*, 2021 60th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE), 2021, pp. 48–52.

[122] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli, *Image quality assessment: from error visibility to structural similarity*, IEEE Trans. Image Process (2004).

[123] Martin Wöllmer, Florian Eyben, Stephan Reiter, Björn Schuller, Cate Cox, Ellen Douglas-Cowie, and Roddy Cowie, *Abandoning emotion classes-towards continuous emotion recognition with modelling of long-range dependencies*, Proc. 9th Interspeech 2008 incorp. 12th Australasian Int. Conf. on Speech Science and Technology SST 2008, Brisbane, Australia, 2008, pp. 597–600.

[124] Martin Wöllmer, Angeliki Metallinou, Nassos Katsamanis, Björn Schuller, and Shrikanth Narayanan, *Analyzing the memory of blstm neural networks for enhanced emotion classification in dyadic spoken interactions*, Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on, IEEE, 2012, pp. 4157–4160.

[125] Martin Wollmer, Björn Schuller, Florian Eyben, and Gerhard Rigoll, *Combining long short-term memory and dynamic bayesian networks for incremental emotion-sensitive artificial listening*, IEEE Journal of Selected Topics in Signal Processing **4** (2010), no. 5, 867–881.

[126] Xinzhou Xu, Jun Deng, Maryna Gavryukova, Zixing Zhang, Li Zhao, and Björn Schuller, *Multiscale kernel locally penalised discriminant analysis exemplified by emotion recogni-*

tion in speech, Proceedings of the 18th ACM International Conference on Multimodal Interaction, ACM, 2016, pp. 233–237.

[127] Junichi Yamagishi, Xin Wang, Massimiliano Todisco, Md Sahidullah, Jose Patino, Andreas Nautsch, Xuechen Liu, Kong Aik Lee, Tomi Kinnunen, Nicholas Evans, et al., *ASVspoof 2021: accelerating progress in spoofed and deepfake speech detection*, arXiv preprint arXiv:2109.00537 (2021).

[128] Ryuichi Yamamoto, Martin Andrews, Michael Petrochuk, Wang Hy, cbrom, Olga Vishnepolski, Matt Cooper, Kuan Chen, and Aleksas Pielikis, *r9y9/wavenet_vocoder: v0.1.1 release*, https://github.com/r9y9/wavenet_vocoder, October 2018.

[129] Ryuichi Yamamoto, Eunwoo Song, and Jae-Min Kim, *Parallel WaveGAN: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram*, ICASSP, 2020.

[130] Geng Yang, Shan Yang, Kai Liu, et al., *Multi-band melgan: Faster waveform generation for high-quality text-to-speech*, SLT workshop, 2021, pp. 492–498.

[131] Yi-Hsuan Yang and Homer H Chen, *Prediction of the distribution of perceived music emotions using discrete samples*, IEEE Transactions on Audio, Speech, and Language Processing **19** (2011), no. 7, 2184–2196.

[132] Mohammed Zakariah, Muhammad Khurram Khan, and Hafiz Malik, *Digital multimedia audio forensics: past, present and future*, Multimedia Tools and Applications **77** (2018), no. 1, 1009–1040.

[133] Heiga Zen, Viet Dang, Rob Clark, et al., *Libritts: A corpus derived from librispeech for text-to-speech*, arXiv preprint arXiv:1904.02882 (2019).

[134] Anthony Zhang, *Speech recognition (version 3.8) [software]*, https://github.com/Uberi/speech_recognition#readme, 2017.

[135] Biqiao Zhang, Georg Essl, and Emily Mower Provost, *Predicting the distribution of emotion perception: capturing inter-rater variability*, Proceedings of the 19th ACM International Conference on Multimodal Interaction, ACM, 2017, pp. 51–59.

[136] Biqiao Zhang, Emily Mower Provost, Robert Swedberg, and Georg Essl, *Predicting emotion perception across domains: A study of singing and speaking.*, AAAI, 2015, pp. 1328–1335.

[137] Shiliang Zhang, Qi Tian, Shuqiang Jiang, Qingming Huang, and Wen Gao, *Affective mtv analysis based on arousal and valence features*, Multimedia and Expo, 2008 IEEE International Conference on, IEEE, 2008, pp. 1369–1372.

[138] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros, *Unpaired image-to-image translation using cycle-consistent adversarial networks*, Proceedings of the IEEE international conference on computer vision, 2017, pp. 2223–2232.

ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 - 1346 USA