

INVESTIGATION OF DEEP NEURAL MODELS FOR SUPERVISED EMOTIONAL SPEECH SYNTHESIS WITH LIMITED DATA

A Dissertation

by

Huda Mohammed Mohammed Barakat

Submitted to the
Graduate School of Sciences and Engineering
in Partial Fulfillment of the Requirements for
the Degree of

Doctor of Philosophy

in the
Department of Computer Science

Özyeğin University
June 2024

Copyright © 2024 by Huda Mohammed Mohammed Barakat

INVESTIGATION OF DEEP NEURAL MODELS FOR SUPERVISED EMOTIONAL SPEECH SYNTHESIS WITH LIMITED DATA

Approved by:

Associate Professor Cenk Demiroğlu,
Advisor
Department of Electrical and
Electronics Engineering
Özyeğin University

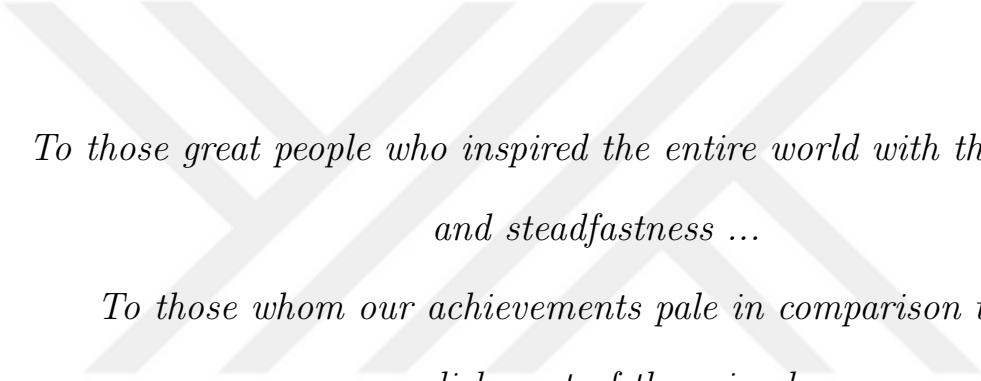
Professor Ümit Güz
Department of Electrical and
Electronics Engineering
Isik University

Assistant Professor Mustafa Furkan
Kıraç
Department of Computer Science
Özyeğin University

Professor Bahadır Kürşat Güntürk
Department of Electrical and
Electronics Engineering
Medipol University

Date Approved: May 16, 2024

Professor Hasan Sözer
Department of Computer Science
Özyeğin University



*To those great people who inspired the entire world with their courage
and steadfastness ...*

*To those whom our achievements pale in comparison to their
accomplishment of the miracles ...*

To those who taught us the real meaning of pride and dignity ...

*To the great people who remain steadfast on the holy land of Palestine
despite everything.*

ABSTRACT

Human speech serves as a richly expressive medium, conveying a multitude of information beyond mere words, including the speaker’s identity, emotions, and different speaking styles. Today’s Text-to-Speech (TTS) models, powered by deep learning, have advanced to the point of producing speech that is almost human-like in its naturalness and intelligibility. Nevertheless, there remains considerable ground to cover in the field of expressive speech synthesis when compared to the expressiveness of human speech. Supervised emotional TTS models (ETTS), capable of generating speech across various emotional states, primarily depend on training data annotated with multiple emotions. The creation of such datasets poses significant challenges, resulting in limited availability and typically smaller sizes. In this thesis, we delve into the investigation of various deep neural models aimed at improving the emotion expressivity of supervised ETTS approaches. The models explored in our research include voice conversion models, emotion classifiers, and adversarial training discriminators. We propose two extended ETTS models: the first combines an emotional voice converter into the ETTS model, while the second incorporates two ensembles, each comprising an emotion classifier and a discriminator, into the ETTS architecture. Within the first model, we examine two distinct voice conversion models: MaskCycleGAN and Seq2Seq voice converters. In the second model, two types of emotion expressivity-related features are employed as input to the ensembles: pitch features and emotion embeddings generated by a variational autoencoder. We conducted several evaluation tests to assess the feasibility of the applied components and the overall performance of each proposed model. Our experiments revealed significant improvements by the proposed models over the baseline model.

ÖZETÇE

İnsan konuşması, konuşmacının kimliği, duyguları ve farklı konuşma stilleri gibi yalnızca kelimelerin ötesinde birçok bilgiyi ileten, zengin bir şekilde ifade edici bir ortam olarak hizmet eder. Günümüzün Derin Öğrenme ile güçlendirilmiş Metin-Konuşma (TTS) modelleri, doğallık ve anlaşılabilirlik açısından neredeyse insan benzeri konuşmalar üretecek kadar ilerlemiştir. Ancak, insan konuşmasının ifadesiyle kıyaslandığında ifade edici konuşma sentezi alanında hâlâ kat edilecek önemli bir yol bulunmaktadır. Çeşitli duygusal durumlarda konuşma üretebilen denetimli duygusal TTS modelleri (ETTS), öncelikle birden fazla duygu ile anotasyonlanmış eğitim verilerine dayanır. Bu tür veri setlerinin oluşturulması önemli zorluklar teşkil eder, bu da sınırlı kullanılabilirlik ve genellikle daha küçük boyutlarla sonuçlanır. Bu tezde, denetimli ETTS yaklaşımlarının duygu ifadesini geliştirmeyi amaçlayan çeşitli derin sinir modellerinin araştırılmasına dalyoruz. Araştırmamızda incelenen modeller arasında ses dönüştürme modelleri, duygu sınıflandırıcılar ve adversarial eğitim ayırt edicileri yer almaktadır. İki genişletilmiş ETTS modeli öneriyoruz: ilki, ETTS modeline duygusal bir ses dönüştürücü entegre ederken, ikincisi, her biri bir duygu sınıflandırıcı ve bir ayırt edici içeren iki ensemble'ı ETTS mimarisine dahil eder. İlk modelde, iki farklı ses dönüştürme modelini inceliyoruz: MaskCycleGAN ve Seq2Seq ses dönüştürücüler. İkinci modelde, ensemble'lara giriş olarak iki tür duygu ifadesiyle ilgili özellik kullanılmaktadır: perde özellikleri ve bir varyasyonel otokodlayıcı tarafından oluşturulan duygu gömmeleri. Uygulanan bileşenlerin uygulanabilirliğini ve her bir önerilen modelin genel performansını değerlendirmek için birkaç değerlendirme testi gerçekleştirdik. Deneylerimiz, önerilen modellerin baz model üzerinde önemli iyileştirmeler sağladığını ortaya koydu.

ACKNOWLEDGEMENTS

First and foremost, I express my gratitude to Allah Almighty, whose support has been indispensable in completing this thesis, and whom I hope will accept it from me. I also extend sincere thanks to my Ph.D. supervisor, Assoc. Prof. Cenk Demiroglu, for his continuous support, advice, and collaboration throughout this long journey. Appreciation is also due to all members of the thesis committee for their valuable guidance and feedback at different stages of this dissertation. I am grateful to Dr. Oytun Turk for his insightful discussions and comments, particularly regarding my publications. I must acknowledge the contributions of both current and former colleagues from the speech lab, with a special mention for the outstanding colleague Eray Eren, whose guidance and knowledge were instrumental to me. Additionally, I thank all my friends at Özyeğin University, with special thanks to my friend Sara Hamdan, for her great companionship during half of this journey before she left the university. Last but not least, I deeply thank all my family members, especially my beloved mother and late father, whose spirit continues to inspire me. To my dear sisters, brothers, nieces, and nephews, your unwavering support has been invaluable throughout this challenging yet rewarding journey.

TABLE OF CONTENTS

DEDICATION	iii
ABSTRACT	iv
ÖZETÇE	v
ACKNOWLEDGEMENTS	vi
LIST OF TABLES	x
LIST OF FIGURES	xi
I INTRODUCTION	1
II BACKGROUND AND MOTIVATION	6
2.1 Supervised Approaches	7
2.1.1 Labels as Input Features	8
2.1.2 Labels as Separate Layers/Models	9
2.1.3 Labels for Emotion Predictors/Classifiers	10
2.2 Unsupervised Approaches	12
2.2.1 Direct Reference Encoding	12
2.2.2 Latent Features via Variational Auto-Encoders	15
2.2.3 Global Style Tokens	18
2.2.4 Approaches Based on In-Context Learning	20
2.2.5 Other Approaches	22
2.3 Main Challenges of ETTS Models	22
2.3.1 Irrelevant Information Leakage	23
2.3.2 Inference without Reference Audio	30
2.3.3 Prosody Controllability	36
2.3.4 Speech Synthesis for Unseen Speakers and Unseen Styles	41
2.3.5 Lack of Labelled Expressive Datasets	42
2.3.6 Inefficient Performance Evaluation Metrics	44

2.4	Thesis Motivation	45
III	VOICE CONVERSION FOR EMOTIONAL SPEECH SYNTHESIS	47
3.1	CycleGAN-Based Voice Conversion Model	48
3.2	Seq2Seq-Based Voice Conversion Model	50
IV	ENSEMBLES OF DISCRIMINATORS AND EMOTION CLASSIFIERS FOR EMOTIONAL SPEECH SYNTHESIS	54
4.1	ETTS Model	55
4.2	Pitch Extractor	57
4.3	Variational Autoencoder	57
4.4	Emotion Classifiers	58
4.5	Discriminators	59
V	RESULTS AND DISCUSSION	62
5.1	Experimental Setup	62
5.1.1	FastSpeech 2	62
5.1.2	Variational Auto-Encoders	63
5.1.3	Baseline Model	64
5.1.4	Dataset	65
5.1.5	General Settings	65
5.2	Voice Conversion for Emotional Speech Synthesis	66
5.2.1	MaskCycleGAN Voice Conversion with FastSpeech2	66
5.2.2	Seq2Seq Voice Conversion with FastSpeech2	70
5.3	Ensembles of Discriminators and Emotion Classifiers for Emotional Speech Synthesis	72
5.3.1	Model Implementation	72
5.3.2	Model Objective Evaluation	76
5.3.3	Model Subjective Evaluation	79
5.3.4	Ablation Study	82
VI	CONCLUSIONS	87

REFERENCES	90
VITA	104



LIST OF TABLES

1	Classification accuracy for the two proposed emotion classifiers at the initial training phase.	74
2	Classification accuracy with wav2vec-based SER model for baseline model and the proposed model.	77
3	Emotions' ratios from the preference test results of the proposed model. Ratios are calculated with regard to the total number of preferred samples by listeners from the proposed model.	81
4	Comparison of proposed model performance via wav2vec-based SER model when excluding each of the components in the proposed integrated ensembles to the ETTS model.	83
5	Comparison of proposed model performance via wav2vec-based SER model when including only one of the components in the proposed integrated ensembles to the ETTS model.	84

LIST OF FIGURES

1	Statistical parametric speech synthesis model structure.	1
2	Structure of TTS models based on deep learning. The autoregressive models follows the upper track with sequential attention mechanism, non-autoregressive models follow the lower track with parallel attention unit utilizing alignments from an external aligner or a pretrained autoregressive model.	2
3	General proposed classification schema of the deep learning based models for expressive speech synthesis.	7
4	Labels represented as multiple separated layers, the shared layers are trained with data from all emotions, the emotion specific layers are trained with emotion related data only.	11
5	Baseline model(1) for unsupervised approaches: Direct reference encoding integrated to an encoder-decoder TTS model.	13
6	Baseline model(2) for unsupervised approaches: Variational autoencoder (VAE) integrated to encoder-decoder TTS model. VAE learns latent space variables mean $\mu(x)$ and variance $\sigma^2(x)$ and samples latent prosody vector(z) from the learned prosody space.	16
7	Baseline model(3) for unsupervised approaches: Global style tokens (GSTs) integrated to encoder-decoder TTS model. Style token layer learns a single weight for each token and generates the style embedding from summation of all the tokens multiplied by their corresponding weights.	18
8	Utilizing in-context learning for training speech synthesis models, adapted from [1].	22
9	Venn diagram of two random variables X and Y where $P(X)$ and $P(Y)$ represent their entropies, $P(X Y)$ is the conditional entropy of X given Y and $P(Y X)$ is the conditional entropy of Y given X, $H(X,Y)$ is the joint entropy of X and Y and $MI(X,Y)$ is their mutual information.	28
10	The general framework of wav2vec technique and its utilization as a feature extractor for generating speech representations as input to the TTS model.	29
11	First proposed model for enhanced emotional speech synthesis. The model combines an emotional voice conversion model to ETTS model where output of ETTS is used as input to the voice converter.	47

12	The two-direction mapping applied by CycleGAN training process utilizing non-parallel data. Reprinted from [2]	49
13	MaskCycleGAN-VC pipeline with filling in frames training (FIF) for the forward-inverse mapping. Reprinted from [3]	50
14	The 2-stage training of Seq2Seq voice conversion model. Reprinted from [4]	52
15	Proposed model of expressive TTS model with ensembles of discriminators and emotion classifiers.	55
16	Modified variational autoencoder structure replacing the decoder with emotion classifier to learn emotion representative latent vectors. . . .	58
17	Applying adversarial training on TTS model which represent the generator in this process. The first discriminator receives pitch sequence as input. The second discriminator receive emotion embedding from VAE.	60
18	Results of emotion expression and speech quality preference test for the two proposed baseline models.	67
19	Results of emotion expression and speech quality (XAB) preference test for each of the proposed baseline models with and without MaskCycleGAN voice converter.	69
20	Results of emotion expression and speech quality (AB) preference test for the two proposed baseline models with Seq2Seq voice converter. .	71
21	Results of emotion expression and speech quality preference test for second baseline model (FS2-VAE-EMBD) with MaskCycleGAN-VC and Seq2Seq-VC voice converters.	72
22	Pitch couture extracted by the trained pitch extractor. (a) and (b) represent pitch extracted when the extractor is trained on original data only while (c) and (d) show pitch extracted when the extractor is trained on both original and synthetic data. (a) and (c) represent pitch extracted from synthetic mel-spectograms while (b) and (d) show pitch extracted from original mel-spectograms.	73
23	K-mean clusters of emotion embeddings from encoder of VAE with where embedding dimension is reduced by t-SNE algorithm.	75
24	Confusion matrix for the Wav2Vec-based emotion classifier results on test samples from the proposed model.	78

25	Listeners' preferences for the emotion expression and speech quality between the baseline model and the proposed model. 95% confidence intervals are shown. Test involves utterances from all speakers and all emotions in the applied dataset.	80
26	Listeners' preferences for the emotion expression between the baseline model and the proposed model for each of the four emotions involved in the applied dataset.	81
27	Listeners' preferences for the emotion expression and speech quality between the proposed model and model utilizing the VAE discriminator only from second ablation study. Test involves utterances from all speakers and all emotions in the applied dataset.	85
28	Listeners' preferences for the emotion expression and speech quality between models utilizing the VAE discriminator and the pitch discriminator individually from the second ablation study. Test involves utterances from all speakers and all emotions in the applied dataset.	86

CHAPTER I

INTRODUCTION

Since its introduction in the late 1950s, Text-to-Speech (TTS) systems have seen significant progress. Today, these systems can generate speech that closely resembles human speech, marking a journey through various stages of advancement. Initially relying on conventional methods like concatenative synthesis, the field has evolved towards more sophisticated techniques such as Statistical Parametric Speech Synthesis (SPSS), primarily leveraging machine learning algorithms like Hidden Markov Models (HMMs) and Gaussian Mixture Models (GMMs).

Despite these advancements, the speech produced by earlier methods still bore a noticeable artificial quality. However, the rise of deep learning in 2006 brought about considerable improvements. Speech synthesis researchers, following trends in other fields, began integrating Deep Neural Networks (DNNs) into their models. Initially, DNNs replaced HMMs and GMMs in SPSS models, while the overall framework remained rooted in the SPSS architecture, as illustrated in Figure 1. As discussed in [5], deep learning-based models have addressed many of the limitations associated with traditional machine learning approaches.

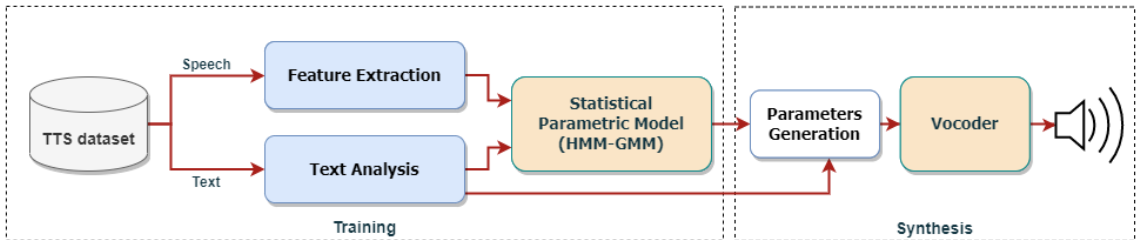


Figure 1: Statistical parametric speech synthesis model structure.

Aside from achieving high speech quality and more human-like speech, researchers

aimed to streamline the framework of text-to-speech models. This was motivated by the complex nature of the SPSS structure, which posed challenges for those without extensive linguistic knowledge and expertise. The advent of deep learning has led to the development of a simplified encoder-decoder structure for TTS models, known as sequence-to-sequence (Seq2Seq) approaches. These proposed methods streamline the conventional TTS structure, which typically involves multiple components, into training a single network. This network converts a set of input text characters or phonemes into a set of acoustic features (mel-spectrograms).

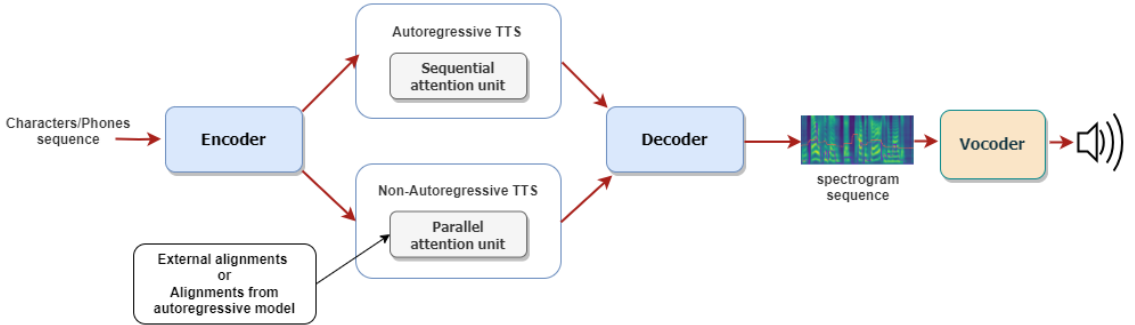


Figure 2: Structure of TTS models based on deep learning. The autoregressive models follow the upper track with sequential attention mechanism, non-autoregressive models follow the lower track with parallel attention unit utilizing alignments from an external aligner or a pretrained autoregressive model.

A primary consideration in these advanced TTS models revolves around the mapping process between input and output sequences, which presents a one-to-many challenge. This is because a single input text can yield multiple speech variations as output. Recent TTS models can be categorized into two groups, illustrated in Figure 2. The first group adopts a sequential (autoregressive) approach, generating mel-spectrograms through soft and automatic attention alignments between input and output sequences, exemplified by the Tacotron model [6, 7]. The second group employs hard alignments between phonemes/characters and mel-spectrograms, resulting in a parallel (non-autoregressive) speech generation process, as demonstrated by the FastSpeech model [8, 9]. This refinement in the TTS model structure has spurred

rapid advancements in the field over the past few years, during which proposed models have achieved speech quality nearly indistinguishable from human speech.

However, human speech is highly expressive, embodying various factors, such as the speaker’s identity, emotion, and speaking style, etc. In addition, there are many applications in which speech synthesis can be utilized, especially expressive speech synthesis. For instance, audiobooks and podcast applications that create audio versions of eBooks and podcasts, translation applications which provide real-time translation of foreign language text, dubbing applications that generate an alternative audio track for a video with different content, speaker, or language, and content creation applications which help produce audio versions of textual content, such as blogs and news articles. E-learning applications that allow for adding voice-over audio to e-learning courses, and conversational AI applications enable machines to communicate with users in a human-like manner, such as AI chatbots and virtual assistants.

Given that spoken language holds significant importance in these applications, it’s essential for users to perceive the interaction as human-like rather than mechanical. Consequently, the speech produced by these applications needs to convey suitable emotion, intonation, stress, and speaking style to align with the ongoing conversation or the specific content type and context of the text being presented.

As a result, there has been a recent attention towards building efficient expressive speech synthesis models as another step forward in achieving human-like speech. Therefore, many studies have been devoted to expressive text-to-speech models (ETTS) as a hot research area, particularly over the last five years. The proposed approaches in general falls under one of two categories including supervised approaches and unsupervised approaches.

Supervised techniques depend on labeled data for training, which pose a notable

challenge given the limited availability of labeled expressive datasets. In contrast, unsupervised methods dispense with the need for labeled data, but they require a style reference or prompt as input during inference, presenting the challenge of providing such a reference or prompt. Furthermore, reference-based approaches encounter the issue of unintended information leakage from the style reference into synthesized speech.

Recently, prompt engineering-based models, characterized by their language modeling-based approaches, have surfaced as the newest trend in the field of speech synthesis. While these approaches may be viewed as the future of the discipline, they are still in their initial stages. Furthermore, their adoption is primarily confined to large companies and research groups due to the substantial data requirements and intensive computational resources they demand.

In this thesis, our objective is to investigate deep neural models or architectures applicable to supervised emotional Text-to-Speech (TTS) models, aiming for improved expressive speech synthesis. Our aim is to enhance the expressiveness of these emotional TTS models while working within the constraints of limited labeled data for model training. We primarily focused on two distinct models, both of which succeeded in enhancing emotion expression in the TTS output without requiring an increase in the training dataset size.

Initially, we explored the integration of a voice conversion model into the TTS framework to enhance emotional expressivity in the synthesized speech. We assessed two different voice conversion models: MaskCycleGAN and Seq2Seq. Both models demonstrated improvements in TTS performance in terms of expressing emotions, with MaskCycleGAN even enhancing speech quality beyond the baseline model.

In the second approach, we introduced two ensembles into the TTS model, each comprising an emotion classifier and a discriminator. While the emotion classifier was trained jointly with the TTS model, the discriminator was employed for adversarial

training on the TTS model. We experimented with two distinct speech features as inputs for each ensemble component: pitch and emotion embedding generated by a variation auto-encoder. Our objective with this architecture was to enhance the generation of these two input features in the synthesized speech. The proposed model with the two ensembles exhibited significantly improved performance over the baseline model in generating speech that better expresses various emotions.



CHAPTER II

BACKGROUND AND MOTIVATION

Expressive speech synthesis (ETTS) has become a hot research area in the last few years where many studies have been devoted to this subject. Based on reviewing papers published since year 2018, we propose to divide ETTS models into two main categories with regard to the learning approach applied in each model. The first category is supervised ETTS approaches where labeled data is utilized. We identified three subcategories under the supervised ETTS approaches including: 1) labels as input features, 2) labels as separate layers or models, and 3) labels for emotion predictors/classifiers.

The second category is the unsupervised ETTS approaches which is grouped into four different subcategories based on the main structure or method used for modeling expressivity in these models. From our observation, most of the proposed methods in the last five years are based on three main early works in this field, namely, reference encoder [10], global style tokens [11], and latent features via variational autoencoders (VAE) [12, 13]. Specifically, proposed models in most of the papers under this category can be considered as an extension or enhancement of one of the three previously mentioned methods. Besides, we identify a fourth subcategory that includes the recent TTS models representing the new trend in the TTS area, which utilizes in-context learning. There is one factor common to all these four unsupervised models, which is that they are all based on using an audio reference/prompt. Additionally, we added a fifth subcategory (named other approaches) in which we include approaches outside the previous four main unsupervised approaches. Figure 3 illustrates the proposed classification schema for the DL-based expressive speech synthesis models.

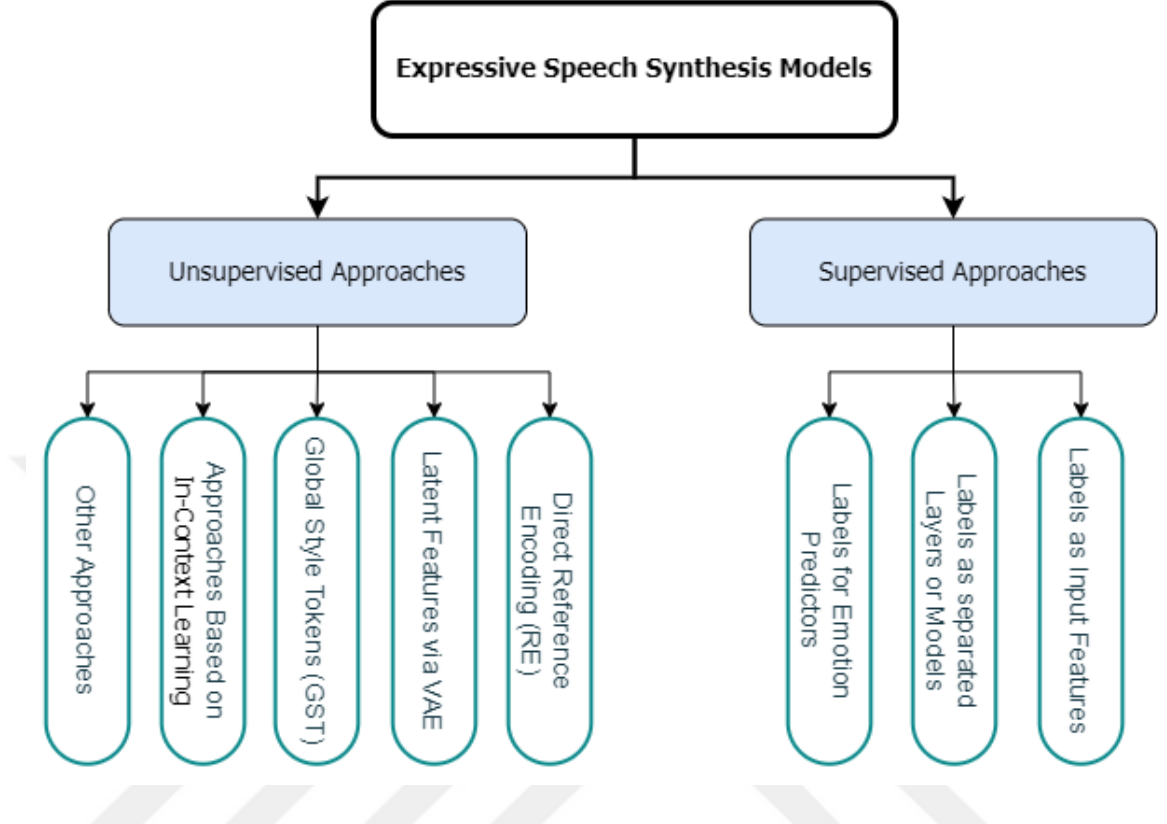


Figure 3: General proposed classification schema of the deep learning based models for expressive speech synthesis.

2.1 Supervised Approaches

Supervised approaches refer to models that are trained on datasets with emotion labels. Those labels guide model training, enabling it to learn accurate weights. Early deep learning-based expressive speech synthesis systems were primarily supervised models that utilized labeled speech exhibiting various emotions (such as sadness, happiness, anger, etc.) or speaking styles (such as talk-show, newscaster, call-center, etc.). Note that the term style has also been used to refer to a set of emotions or a mixture of emotions and speaking styles [14, 15, 16, 17]. Generally, the structure of early conventional TTS models was built upon two primary networks: one for predicting duration and the other for predicting acoustic features. These acoustic features were then converted to speech using vocoders. Both networks receive linguistic features extracted from the input text. In supervised ETTS approaches, speech labels

(emotions and/or styles) are represented in the TTS model as either input features or as separate layers, models, or sets of neurons for each specific label. The following sections explain these three representations in detail.

2.1.1 Labels as Input Features

The most straightforward method for representing emotion labels of annotated datasets as input to the TTS model is by using a one-hot vector. This approach entails using a vector with a size equivalent to the number of available labels. In this vector, a value of (1) is assigned to the index corresponding to the label ID, while all other values are set to (0). Many early ETTS models [18, 19, 20, 21, 22, 14, 23, 24] advocated for this direct representation of emotion labels in order to generate speech encompassing various emotions. The one-hot emotion vector, also referred to as a style/emotion code in some studies [21, 18, 14, 20], is concatenated with the input linguistic features of the model.

When dealing with large number of labels, the one-hot representation becomes both high-dimensional and sparse. Moreover, in other scenarios, merging label vectors with input features instead of concatenation can lead to length mismatch issues. In both situations, the embedding layer offers a solution by creating a continuous representation for each label, known as embedding vectors. Unlike the one-hot vector, which is constrained in size based on the number of labels, an emotion embedding can have any dimension, regardless of the number of available labels.

For instance, in [23], each sample in the training dataset has three separated labels including speaker, style (emotion), and cluster. In this context, the cluster value indicates the consistency in speech quality of a given speaker and style pair. If one-hot vector is used to represent each unique combined label of each sample, the resulting label vector will be high dimensional (which in this case is 67). Therefore, the three one-hot vectors representing the given three labels are combined and passed

as input to an embedding layer to reduce its dimension (in this case 15). On a different note, [25] utilizes an embedding layer to expand concise binary one-hot label vectors to match with the dimensions of the input features to be added together as input to the TTS model.

To address the potential disparities between a talker’s intent and a listener’s perception when annotating emotional samples, in [26], a different methodology for representing labels is introduced. In the context of N emotion classes, each sample from the talker may be perceived by the listener as one of the N emotions. In response to this, the paper suggests the adoption of a singular vector termed the ‘perception vector,’ with N dimensions. This vector represents how samples from a specific emotion class are distributed among the N emotions, based on the listener’s perception. Furthermore, in the context of multiple listeners, each emotion class can be represented as a confusion matrix that captures the diverse perceptions of samples belonging to that emotion class by multiple listeners.

2.1.2 Labels as Separate Layers/Models

In this approach, to represent emotion or style labels in TTS models, each label is associated with either a separate instance of the DNN model, an emotion-specific layers, or a set of emotion-specific neurons within a layer. Initially, the model is trained using neutral data, which typically has larger size. Subsequently, in the first approach, multiple copies of the trained model are fine-tuned using emotion-specific data of small size [20, 14]. In the second approach, instead of creating an individual model for each emotion, only specific model layers (usually the uppermost or final layers) from the employed DNN model are assigned to each emotion [19, 20, 14, 24] as shown by Figure 4. While shared layers are adjusted during training using neutral data, output layers corresponding to each emotion are modified exclusively when the model is trained with data from the respective emotion.

Alternatively, when dealing with limited data for certain emotions/styles, the model can initially undergo training for emotions with large amount of data. Following this step, the weights of the shared layers within the model are fixed, and only the weights of the top layers are fine-tuned using the limited, emotion-specific data [27]. Another method for representing emotion labels involves allocating specific neurons from a layer within the DNN model for each emotion. In this approach, the hidden layers of the model could be expanded by introducing new neurons. Then, as outlined in [28], particular neurons from this expanded set are assigned to represent each distinct emotion. Importantly, the associated weights of these specific neuron subsets are adjusted solely during the processing of data relevant to the corresponding emotion. Furthermore, by substituting the subset of neurons dedicated to a particular emotional class with a different set, the model becomes capable of generating speech imbued with the desired emotional class. This capability holds true even for new speakers who only possess neutral data, and in this case, it is known as expression/emotion transplantation.

2.1.3 Labels for Emotion Predictors/Classifiers

Another common approach to utilize emotion labels is to use them directly or via emotion predictor or classifier to support the process of extracting emotion/prosody embedding. For example, in [29] emotion labels represented as one-hot vectors are used as targets for the weight vectors of GSTs (explained in Section 2.2.3) where a cross entropy loss between the two vectors is added to the total loss function. [30] proposes a joint emotion predictor based on the Generative Pre-trained Transformer (GPT)-3 [31]. The proposed predictor produces two outputs including emotion class and emotion strength based on features extracted from input text by (GPT)-3. A joint emotion encoder is then used to encode the predictor outputs into a joint emotion embedding. The joint emotion predictor is trained with the guidance of the emotion

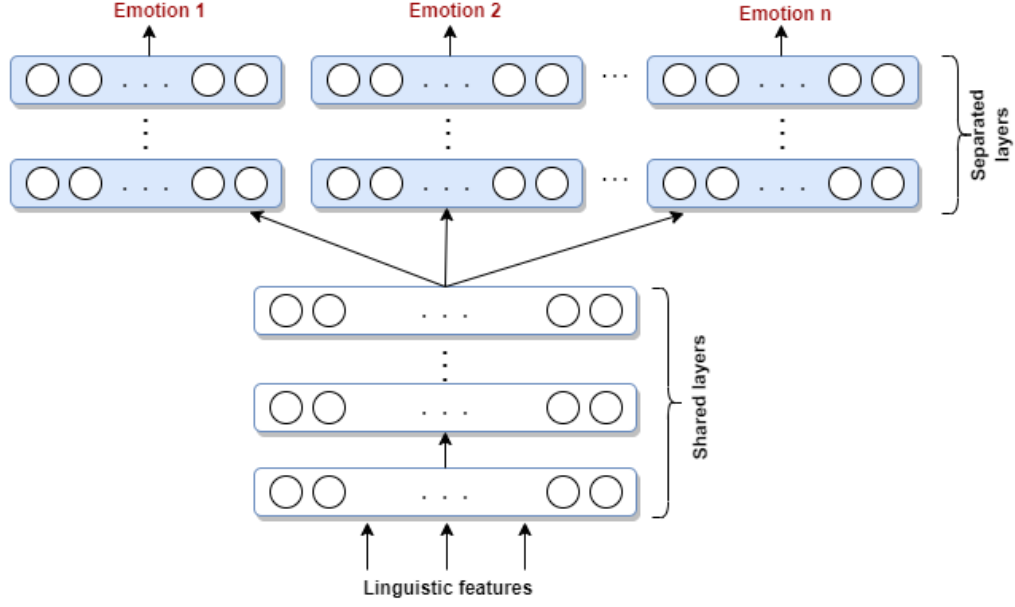


Figure 4: Labels represented as multiple separated layers, the shared layers are trained with data from all emotions, the emotion specific layers are trained with emotion related data only.

labels and emotion strength values obtained via a ranking support vector machine (RankSVM) [32].

In [33], an emotion classifier is used to produce more discriminative emotion embeddings. Initially, the input Mel-spectrogram features from the reference-style audio and those predicted by the proposed TTS model are passed to two reference encoders (explained in Section 2.2.1) to generate reference embeddings. Both embeddings are then fed to two emotion classifiers, which consist of intermediate fully connected (FC) layers. The output of the second FC layer from both classifiers is considered as the emotion embedding. Apart from the loss of the classifiers, an additional loss function is established between the resulting emotion embeddings from the two classifiers. Similarly, an emotion classifier is also employed in [34] to reduce irrelevant information in the generated emotion embedding from an emotion encoder with reference speech (Mel-spectrogram) as input.

Several other studies [35, 34, 36] that support multiple speakers also suggest utilizing a speaker classifier in addition to the emotion classifier. This approach aims to improve the speaker embedding derived from speaker encoders. Moreover, these studies introduce an adversarial loss between the speaker encoder and the emotion classifier using a gradient reversal layer (GRL) [37]. The purpose of this is to minimize the potential transfer of emotion-related information into the speaker embedding. The GRL technique involves updating the weights of the speaker encoder by utilizing the inverse of the gradient values obtained from the emotion classifier during the training process.

2.2 Unsupervised Approaches

Due to the limited availability and challenges associated with collecting or preparing labeled datasets of expressive speech, as discussed in Section 2.3.5, many researchers tend to resort to unsupervised approaches for generating expressive speech. Within these approaches, models are trained to extract speaking styles or emotions from expressive speech data through unsupervised methods. Unsupervised models typically utilize reference speech as an input to the TTS model, which extracts a style or prosody embedding which is then used to synthesize speech resembling the input style reference. In the literature, three primary structures emerge as baseline models for unsupervised ETTS models: including reference encoders, global style tokens, and variational auto-encoders, which are explained in the following three sections. In addition, we identify the recent TTS models that utilize in-context learning as another group of unsupervised approaches. The last subcategory under the unsupervised approaches involves other individual approaches.

2.2.1 Direct Reference Encoding

The main approach, based on a reference or prosody encoder, can be traced back to an early Google paper [10]. The paper suggests using a reference encoder to

produce a low-dimensional embedding for a given style reference audio, which is called a prosody embedding. This encoder takes spectrograms as input to represent the reference audio. The generated prosody embedding is then concatenated with the text embedding derived from the text encoder of a Seq2Seq TTS model such as Tacotron [6, 7]. Figure 5 shows reference encoder integrated to the TTS model.

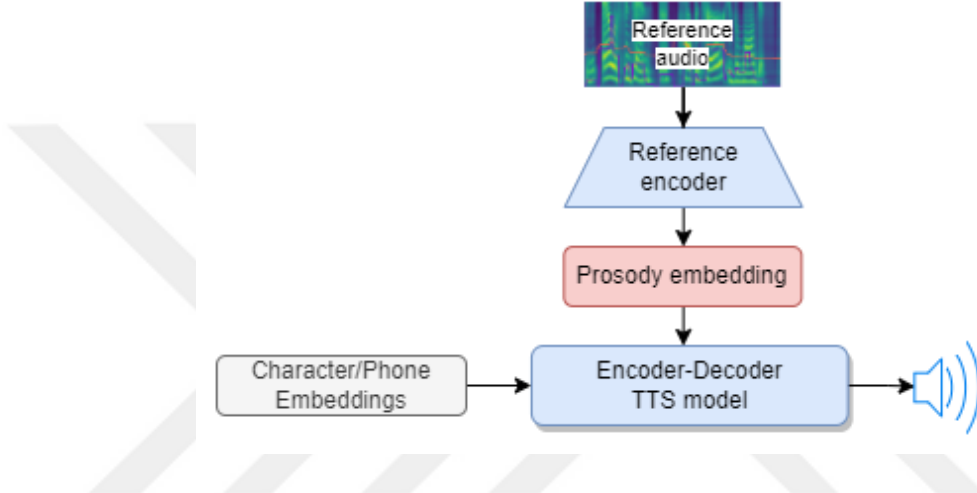


Figure 5: Baseline model(1) for unsupervised approaches: Direct reference encoding integrated to an encoder-decoder TTS model.

Various features have been employed in the literature as inputs for the reference encoder. For example, in the work [38], MFCC features extracted using the openSMILE toolkit [39] are fed into one of the encoders within its style extraction model, which is composed of a multi-modal dual recurrent encoder (MDRE). In another study [40], the reference encoder is proposed as a ranking function model, aimed at learning emotion strength at the phoneme level. This model leverages the OpenSMILE toolkit to extract 384-dimensional emotion-related features from segments of reference audio, derived using a forced alignment model for phoneme boundaries. Furthermore, in work [41], a word-level prosody embedding is generated. This is achieved by extracting phoneme-level F0 features from reference speech using the WORLD vocoder [42] and an internal aligner operating with the input text.

A prosody-aware module is proposed in [43] which extracts other prosody-related

features. The prosody-aware module consists of an encoder, an extractor, and a predictor. The encoder receives the three phoneme-level features including logarithmic fundamental frequency (LF0), intensity, and duration from the extractor as input and generates the paragraph prosody embedding with the assistance of an attention unit. Simultaneously, the predictor is trained to predict these features at inference time based on the input text embedding only.

In Daft-Exprt TTS model [44], the prosody encoder receives pitch, energy and spectrogram as input. The prosody encoder then uses FiLM conditioning layers [45] to carry out affine transformations to the intermediate features of specific layers in the TTS model. A slightly modified version of the FastSpeech2 [9] model is utilized in this work where the phoneme encoder, prosody predictor and the decoder are the conditioned components. The prosody predictor is similar to the variance adaptor of FastSpeech2 but without the length regulator, and it estimates pitch, energy and duration at phoneme-level.

A pre-trained Wav2Vec model [46] has also been utilized for extracting features from the reference waveform. These features serve as input to the reference encoders of the proposed Emo-VITS model [47], which integrates an emotion network into the VITS model [48] to enhance expressive speech synthesis. In fact, the emotion network in the Emo-VITS model comprises two reference encoders. The resulting emotion embeddings from these encoders are then combined through a feature fusion module that employs an attention mechanism. Wav2vec 2.0-derived features from the reference waveform in this work are particularly suitable for attention-based fusion and contribute to reducing the textual content within the resulting embeddings [47].

In contrast, [49] proposes a an image style transfer module to generate input for reference encoder. The concept of image style transfer involves altering the artistic style of an image from one domain to another while retaining the image’s original

content [50]. In this specific research, the style reconstruction module from VGG-19 [51], a deep neural network primarily used for image classification, is employed to extract style-related information from the Mel-spectrogram used as input image. Subsequently, the output of this module is fed into the reference encoder to generate the style embedding.

2.2.2 Latent Features via Variational Auto-Encoders

The goal of TTS models under this is to map input speech from the higher dimensional space to a well-organized and lower-dimensional latent space utilizing variational auto-encoders (VAEs) [52]. VAE is a generative model that is trained to learn the mapping between observed data x and continuous random vectors z in an unsupervised manner. In detail, VAEs learn a Gaussian distribution denoted as the latent space from which the latent vectors representing the given data x can be sampled. A typical variational autoencoder consists of two components. First, the encoder learns the parameters of the z vectors (latent distribution), namely the mean $\mu(x)$ and variance $\sigma^2(x)$, based on the input data x . Second, the decoder regenerates the input data x based on latent vectors z sampled from the distribution learned by the encoder. In addition to the reconstruction loss between the model input and the data, variational autoencoders are also trained to minimize a latent loss, which ensures that the latent space follows a Gaussian distribution.

Utilizing VAEs in expressive TTS models as shown by Figure 6, allows for mapping the various speech styles within the given dataset to be encoded as latent vectors, often referred to as prosody vectors, within this latent space. During inference, these latent vectors can be sampled directly or with the guidance of reference audio from the VAE’s latent space. Furthermore, the latent vectors offer the advantage of disentangling prosody features, meaning that some specific dimensions of these vectors independently represent single prosody features such as pitch variation or speaking

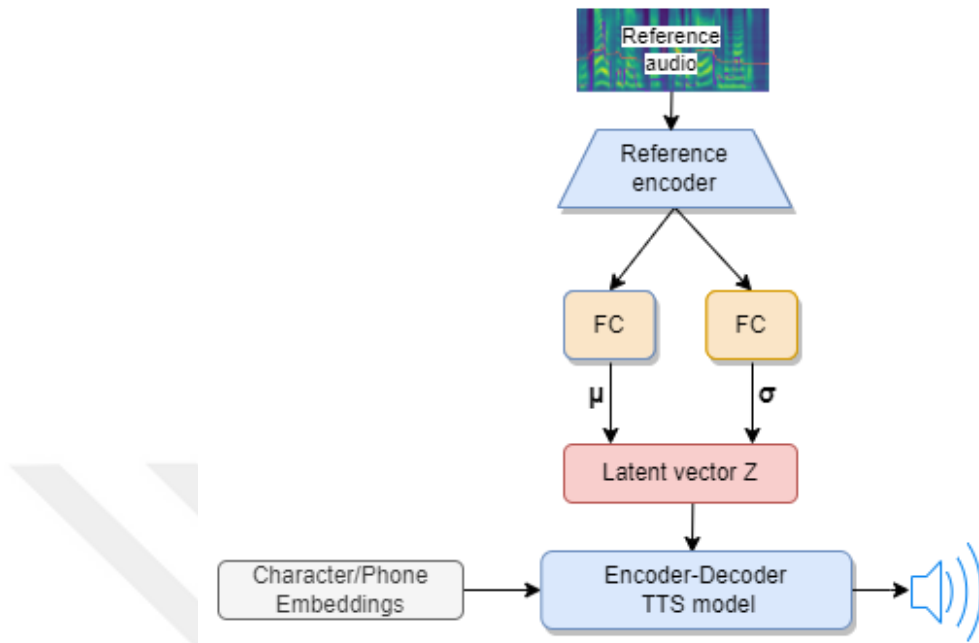


Figure 6: Baseline model(2) for unsupervised approaches: Variational autoencoder (VAE) integrated to encoder-decoder TTS model. VAE learns latent space variables mean $\mu(x)$ and variance $\sigma^2(x)$ and samples latent prosody vector(z) from the learned prosody space.

rate. Disentangled prosody features allow for better prosody control via manipulating the latent vectors with different operations such as interpolation and scaling [13]. The two early papers, [12, 13], can be regarded as the baseline for latent feature-based approaches. The former study [12] introduces VAE within the VoiceLoop model [53], while the latter [13] incorporates VAE into Tacotron2 [7] as an end-to-end TTS model for expressive speech synthesis.

In the same direction of modeling the variation of the prosodic features in expressive speech, studies [54, 55] propose a hierarchical structure for the baseline variational autoencoder, known as Clockwork Hierarchical Variational AutoEncoder (CHiVE). Both the encoder and decoder in the CHiVE model have several layers to capture prosody at different levels based on the input text’s hierarchical structure. Accordingly, linguistic features are also used alongside acoustic features as input to the

model’s encoder. The model’s layers are dynamically clocked at specific rates: sentence, words, syllables, and phones. The encoder hierarchy goes from syllables to the sentence level, while the decoder hierarchy is in the reversed order. The CHiVE-BERT model in [55], differs from the main model in [54] as it utilizes BERT [56] features for input text at the word-level. Since the features extracted by the BERT model incorporate both syntactic and semantic information from a large language model, CHiVE-BERT model is expected to have improved the prosody generation.

Other studies [57, 58] propose Vector-Quantized Variational Auto-Encoder (VQ-VAE) to achieve discretized latent prosody vectors. In vector quantization (VQ) [59], latent representations are mapped from the prosody latent space to a codebook of a limited number of prosody codes. Specifically, during training, the nearest neighbor lookup algorithm is applied to find the nearest codebook vector to the output of the reference encoder and used to condition TTS decoder. To further improve the quality of latent prosody vectors and consequently the expressiveness of the generated speech, DiffProsody[58] proposes a diffusion-based VQ-VAE model. In the proposed model a prosody generator that utilizes a denoising diffusion generative adversarial networks (DDGANS) [60] is trained to generate the prosody latent vectors based only on text and speaker information. At inference time, the prosody generator is used to produce prosody vectors based on input text and with no need for an audio reference which improves both quality and speed of speech synthesis.

While most of the studies in this category follow the baseline model and use mel-spectrograms to represent the reference audio, other studies extract correlated prosody features as input to the VAE. For instance, frame-level F0, energy, and duration features are extracted from the reference speech as basic input for the hierarchical encoder of the CHiVE model [54]. These same features are also used as input for the VAE encoder in work [61], but at the phoneme level. In work [15], multi-resolution VAEs are employed, each with acoustic and linguistic input vectors. The acoustic

feature vectors for each encoder include 70 mel-cepstral coefficients, log F0 value, a voiced/unvoiced value, and 35 mel-cepstral analysis aperiodicity measures.

2.2.3 Global Style Tokens

The Global Style Tokens (GST) approach for expressive synthesis was first introduced in [11]. The paper proposes a framework to learn various speaking styles (referred to as style tokens) in an unsupervised manner within an end-to-end TTS model. The proposed approach can be seen as a soft clustering method that learns soft style clusters for expressive styles in an unlabeled dataset. In detail, GST, as shown by Figure 7, extends the approach introduced in [10] by passing the resulting style embedding from the reference encoder to an attention unit, which functions as a similarity measure between the style embedding and a bank of randomly initialized tokens. During training, the model learns the style tokens and a set of weights, where each style embedding is generated via a weighted sum of the learned tokens. In fact, the obtained weights represent how each token contributes to the final style embedding. Therefore, each token will represent a single style or a single prosody-related feature, such as pitch, intensity, or speaking rate.

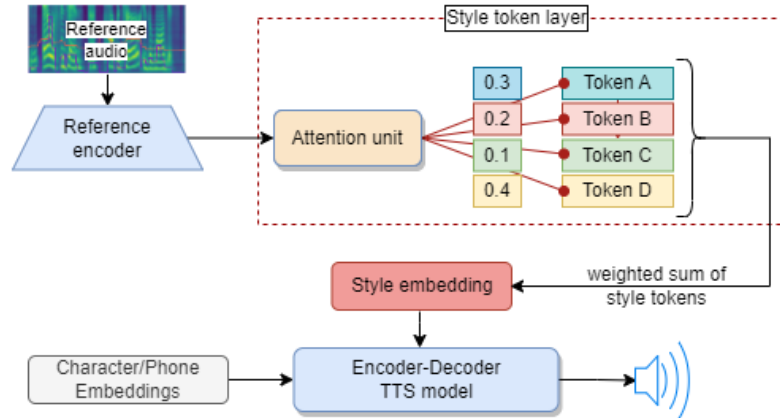


Figure 7: Baseline model(3) for unsupervised approaches: Global style tokens (GSTs) integrated to encoder-decoder TTS model. Style token layer learns a single weight for each token and generates the style embedding from summation of all the tokens multiplied by their corresponding weights.

At inference time, a reference audio can be passed to the model to generate its corresponding style embedding via a weighted sum of the style tokens. Alternatively, each individual style token can be used as a style embedding. In addition, GSTs offer an enhanced control over the speaking style through various operations. These include manual weight refinement, token scaling with different values, or the ability to condition different parts of the input text with distinct style tokens.

The GST-TTS model can be further enhanced by modeling different levels of prosody to improve both expressiveness and control over the generated speech. For instance, [62] proposes a fine-grained GST-TTS model where word-level GSTs are generated to capture local style variations (WSVs) through a prosody extractor. The WSV extractor consists of a reference encoder and a style token layer, as described in [11], along with an attention unit to produce the word-level style token

In [63] a hierarchical structure of multi-layer GSTs with residuals is proposed. The model employs three GST layers, each with 10 tokens, resulting in a better interpretation of the tokens of each level. Upon tokens analysis, it was found that the first-layer tokens learned speaker representations, while the second-layer tokens captured various speaking style features such as pause position, duration, and stress. The third-layer tokens, however, were able to generate higher-quality samples with more distinct and interpretable styles. Similarly, in [64], a multi-scale GST extractor is proposed to extract speaking style at different levels. This extractor extracts style embeddings from the reference mel-spectrogram using three style encoders at global, sentence, and sub word levels, and combines their outputs to form the multi-scale style embedding.

With only a small portion of the training dataset labeled with emotions, [29] proposes a semi-supervised GST model for generating emotional speech. The model applies a cross-entropy loss between the one-hot vectors representing the emotion labels and the weights of GSTs, in addition to the GST-TTS reconstruction loss.

The semi-GST model is trained on a dataset in which only 5% of the samples are labeled with emotion classes, while the rest of the dataset is unlabeled. After training, each style token represents a specific emotion class from the training dataset and can be used to generate speech in the corresponding emotion.

Furthermore, in [65], a speech emotion recognition (SER) model is proposed with the GST-TTS to generate emotional speech while acquiring only a small labeled dataset for training. The paper formulates the training process as reinforcement learning (RL). In this framework, the GST-TTS model is treated as the agent, and its parameters serve as the policy. The policy aims to predict the emotional acoustic features at each time step, where these features represent the actions. The pre-trained SER model then provides feedback on the predicted features through emotion recognition accuracy, which represents the reward. The policy gradient strategy is employed to perform backpropagation and optimize the TTS model to achieve the maximum reward.

In contrast, the Mellotron model [66] introduces a unique structure for the GSTs, enabling Mellotron to generate speech in various styles, including singing styles, based on pitch and duration information extracted from the reference audio. This is achieved by obtaining a set of explicit and latent variables from the reference audio. Explicit variables (text, speaker, and F0 contour) capture explicit audio information, while latent variables (style tokens and attention maps) capture the latent characteristics of speech that are hard to extract explicitly.

2.2.4 Approaches Based on In-Context Learning

These is a group of recent TTS models that are trained on a large amounts of data using in-context learning strategy. During in-context learning (also called prompt engineering), the model is trained to predict missing data based its context. In other words, the model is trained with a list of input-output pairs formed in a way that

represents the in-context learning task. After training, the model should be able to predict the output based on a given input.

For the TTS task, the provided speech reference (referred to as prompt) is considered as part of the entire utterance to be synthesized. The TTS model training task is to generate the rest of this utterance following the style of the provided prompt as shown by Figure 8. By employing this training strategy, recent TTS models such as VALL-E [67], NaturalSpeech 2 [68], and Voicebox [1] are capable of producing zero-shot speech synthesis using only a single acoustic prompt. Furthermore, these models demonstrate the ability to replicate speech style/emotion from a provided prompt [67, 68] or reference [1] to the synthesized speech.

In VALL-E [67], a language model is trained on tokens from Encodec [69], and the input text is used to condition the language model. Specifically, the Encodec model tokenizes audio frames into discrete latent vectors/codes, where each audio frame is encoded with eight codebooks. VALL-E employs two main models: the first one is an auto-regressive (AR) model that predicts the first code of each frame, and the second is non-auto-regressive (NAR) model that predicts the other seven codes of the frame.

Instead of discrete tokens used in VALL-E, NaturalSpeech 2 [68] represents speech as latent vectors from a neural audio codec with residual vector quantizers. The latent vectors are then predicted via a diffusion model, conditioned on input text, pitch from a pitch predictor, and input speech prompt.

Another example of in-context training is Voicebox [1] which is a versatile generative model for speech trained on a large amount of multilingual speech data. The model is trained on a text-guided speech infilling task, which gives it the flexibility to perform various speech tasks such as zero-shot TTS, noise removal, content editing, and diverse speech sampling. Voicebox is modeled as a non-autoregressive (NAR) flow-matching model with the ability to consider future context.



Figure 8: Utilizing in-context learning for training speech synthesis models, adapted from [1].

2.2.5 Other Approaches

This category contains reviewed papers that propose individual techniques or methods which cannot be categorized under any of the previously mentioned unsupervised approaches. For instance, in [70], a neural encoder is introduced to encode the residual error between the predictions of a trained average TTS model and the ground truth speech. The encoded error is then used as a style embedding that conditions the decoder of the TTS model to guide the synthesis process. [71] improves prosody modeling of FastSpeech2 model [9] with an additional variance adaptor for utterance-wise prosody modeling.

As context information is strongly related to speech expressivity, [72] proposes using multiple self-attention layers in Tacotron2 [7] encoder to better capture the context information in the input text. The outputs of these layers in the encoder are combined through either direct aggregation (concatenation) or weighted aggregation using a multi-head attention layer. Additionally, there are some papers that propose using only input text to obtain prosody-related representations/embeddings without any style references, and those are further discussed in Section 2.3.2.4.

2.3 Main Challenges of ETTS Models

In this section, we list and explain the most important challenges that face expressive TTS models and the main solutions that have been proposed in the literature to overcome these challenges.

2.3.1 Irrelevant Information Leakage

One main problem in unsupervised approaches that rely on having a style reference or a prompt, is the leakage of irrelevant information, like speaker or text related information, into the generated style or prosody embedding. This irrelevant information within the speech style can lead to degradation in the quality of the synthesized speech. As a result, many studies have investigated this problem, and several solutions have been proposed as outlined below.

2.3.1.1 Adversarial Training

Adversarial training [37] is one of the widely used techniques to confront the information leakage problem. Typically, a classifier is trained to distinguish the type of unwanted information (such as speaker or content information) that is leaking from the prosody reference audio into the generated prosody embedding. During the training process, the weights of the employed prosody encoder/extractor from the reference audio are modified with gradient inversion of the proposed classifier. In other words, the classifier penalizes the prosody encoder/extractor for any undesired information in its output. A gradient reversal layer (GRL) is usually used to achieve the inversion of the classifier gradients.

Several studies utilize adversarial training to prevent the flow of either speaker or content-related information from the given reference audio to the resulting prosody embedding. For instance, the VAE-TTS model proposed in [73] learns phoneme-level 3-dimensional prosody codes. The VAE is conditioned on speaker and emotion embeddings, besides the tone sequence and mel-spectrogram from the reference audio. Adversarial training using a gradient reversal layer (GRL) is applied to disentangle speaker and tone from the resulting prosody codes. Similarly, adversarial training is introduced to the style encoder of the cross-speaker emotion transfer model proposed in [74] to learn a speaker-independent style embedding, where the target speaker

embedding is provided from a separate speaker encoder.

The STYLER model in [75] employs multiple style encoders to decompose the style reference into several components, including duration, pitch, speaker, energy, and noise. As noise is encoded individually by a separate encoder in the model, other encoders are constrained to exclude noise information by employing either domain adversarial training or residual decoding. In [76], a style encoder that models prosody at the phone-level is trained via adversarial training utilizing a phone identity classifier for the generated style embedding.

On the other hand, [77] proposes adversarial training for the style reference by inverting the gradient of an automatic speech recognition (ASR) model. The proposed model introduces a shared layer between an ASR and a reference encoder-based model. A gradient reversal layer (GRL) is employed by the ASR model to reverse its gradient on the shared layer. Accordingly, the reference encoder parameters are modified so that the ASR model fails to recognize the shared embedding, and thus content leakage to the style embedding from the reference encoder is reduced.

2.3.1.2 Prosody Classifiers

This is a supporting approach used by some studies to produce more discriminative prosody embeddings by passing them to a prosody classifier. This method can be applied when the training data is labeled with emotion or style labels. In the two consecutive studies [33, 35] from the same research group, an auxiliary reference encoder is proposed and located after the decoder of the baseline TTS model [10]. The two reference encoders in the model are followed by emotion classifiers to further enhance the discriminative nature of their resulting embeddings. However, the emotion embedding that is passed to the TTS model is the output of an intermediate hidden layer of the classifiers. In addition to the classification loss, an additional style loss is also applied between the two emotion embeddings from the two employed emotion

classifiers.

In [34], alongside the text encoder, two encoders are introduced to generate embeddings for speaker and emotion from a reference audio. To further disentangle emotion, speaker, and text information, both speaker and emotion encoders are supported with a classifier to predict speaker and emotion labels, respectively. Similarly, in paper [36], a model with two encoders and two classifiers is proposed to produce disentangled embeddings for speakers and emotions from a reference audio.

2.3.1.3 Information Bottleneck

The information bottleneck is a technique used to control information flow via a single layer/network. It helps prevent information leakage as it projects input into a lower dimension so that there is not enough capacity to model additional information and only important information is passed through it. In other words, the bottleneck can be seen as a down-sampling and up-sampling filter that restricts its output and generates a pure style embedding. Several prosody-reference based approaches, as in [78, 79, 80, 81, 75], have employed this technique to prevent the flow of speaker or content-related information from the reference audio to the prosody embedding.

In [81], a bottleneck layer named sieve layer is introduced to the style encoder in GST-TTS to generate pure style embedding. Similarly, in the multiple style encoders model STYLER [75], each encoder involves a channel-wise bottleneck block of two bidirectional-LSTM layers to eliminate content information from encoders' output. Another example is the cross-speaker-style transfer Transformer-TTS model proposed in [80] with both speaker and style embeddings as input to the model encoder. The speaker-style-combined output from the encoder is then passed to a prosody bottleneck sub-network, which produces a prosody embedding that involves only prosody-related features.

The Copycat TTS model [78] is a prosody transfer model via VAE. The model applies three techniques to disentangle the source speaker information from the prosody embedding. One of these techniques is to use a temporal bottleneck encoder [82] within the reference encoder of the model. The prosody embedding that is sampled from the latent space is passed to the bottleneck to reduce speaker identity-related information in the prosody embedding before it flows to the model decoder. Similarly, the model proposed in [79] produces a style embedding with less irrelevant style information by adding a variational information bottleneck (VIB) [83] layer to the reference encoder.

2.3.1.4 Instance Normalization

Batch normalization (BN), first introduced in [84], is utilized in deep neural networks to accelerate the training process and increase its stability. Essentially, a batch normalization layer is added before each layer in deep neural networks to adjust the means and variances of the layer inputs, as illustrated by equation (1):

$$IN(x) = \gamma \left[\frac{x - \mu(x)}{\sigma(x)} \right] + \beta \quad (1)$$

Where γ , β are affine parameters learned from data and μ , σ are the mean and standard deviation which are calculated for each feature channel across the batch size. Instance normalization (IN) also follows equation (1); however, it calculates means and variances across spatial dimensions independently for each channel and each sample (instance). In the field of computer vision, stylization approach is significantly improved by replacing (BN) layers with (IN) layers [85]. Consequently, researchers in the expressive speech field have started to apply IN to extract better prosody representations. For example, an instance normalization (IN) layer is used at the reference encoder in [78], at the prosody extractor in [81], and at the style encoder in [86] to remove style/prosody irrelevant features (such as speaker identity features)

and enhance the learned style/prosody embedding.

2.3.1.5 Mutual Information Minimization

For a pair of random variables, mutual information (MI) is defined as the information obtained on one random variable by observing the other. Specifically, if X and Y are two variables, then $MI(X; Y)$ shown by Venn diagram in Figure 9, can be seen as the KL-divergence between the joint distribution (P_{XY}) and the product of the marginals (P_X, P_Y) as in equation (2). If the two random variables X and Y represent linguistic and style vectors, applying MI minimization between these two vectors helps to produce style vectors with less information from the content vector.

$$MI(X; Y) = DL_{KL}(P_{(X,Y)} \| P_X \otimes P_Y) \quad (2)$$

For example, in [87], the Mutual Information Neural Estimation algorithm (MINE) [88] is employed to estimate the mutual information between the content and style vectors. The algorithm uses a neural network that is trained to maximize the lower bound of the mutual information between the style and content vectors. Simultaneously, the TTS model aims to minimize the reconstruction loss, making the overall problem a max-min problem. Alternatively, in [89], the CLUB method [90], which computes an upper bound as the MI estimator, is used to prevent the leakage of speaker and content information into the style embedding.

A new approach is proposed in [91] for MI estimation and minimization to reduce content/speaker information transfer to the style embedding in a VAE based approach. Typically, the model needs to estimate MI between latent style embeddings and speaker/content embeddings. To avoid the exponentially high statistical

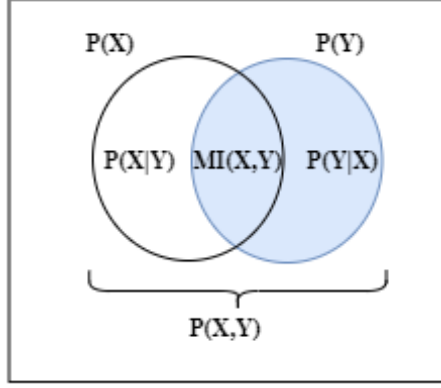


Figure 9: Venn diagram of two random variables X and Y where $P(X)$ and $P(Y)$ represent their entropies, $P(X|Y)$ is the conditional entropy of X given Y and $P(Y|X)$ is the conditional entropy of Y given X , $H(X,Y)$ is the joint entropy of X and Y and $MI(X,Y)$ is their mutual information.

variance of the finite-sampling MI estimator, the paper suggests using a new algorithm for information divergence named Rényi divergence. Two variations from the Rényi divergence family are proposed, including minimizing the Hellinger distance and minimizing the sum of Rényi divergences.

2.3.1.6 Wav2Vec Features

Wav2Vec [46] model converts speech waveform into context-dependent vectors/features. The model is trained via self-supervised or in-context training algorithms which are explained in Section 2.2.4. Features generated by wav2vec and similar models such as HuBERT [92] provide better representations of speech and its lexical and non-lexical information. Therefore, these models are utilized nowadays in different speech processing tasks such as speech recognition, synthesis, and downstream emotion detection.

Some studies such as [93, 47] use Wav2vec 2.0 as a feature extractor to provide input to the reference encoder instead of spectrum features or raw audio waveform. Figure 10 illustrates the framework of the wav2vec technique and how it is utilized as a feature extractor with TTS models. The wav2vec model converts the continuous

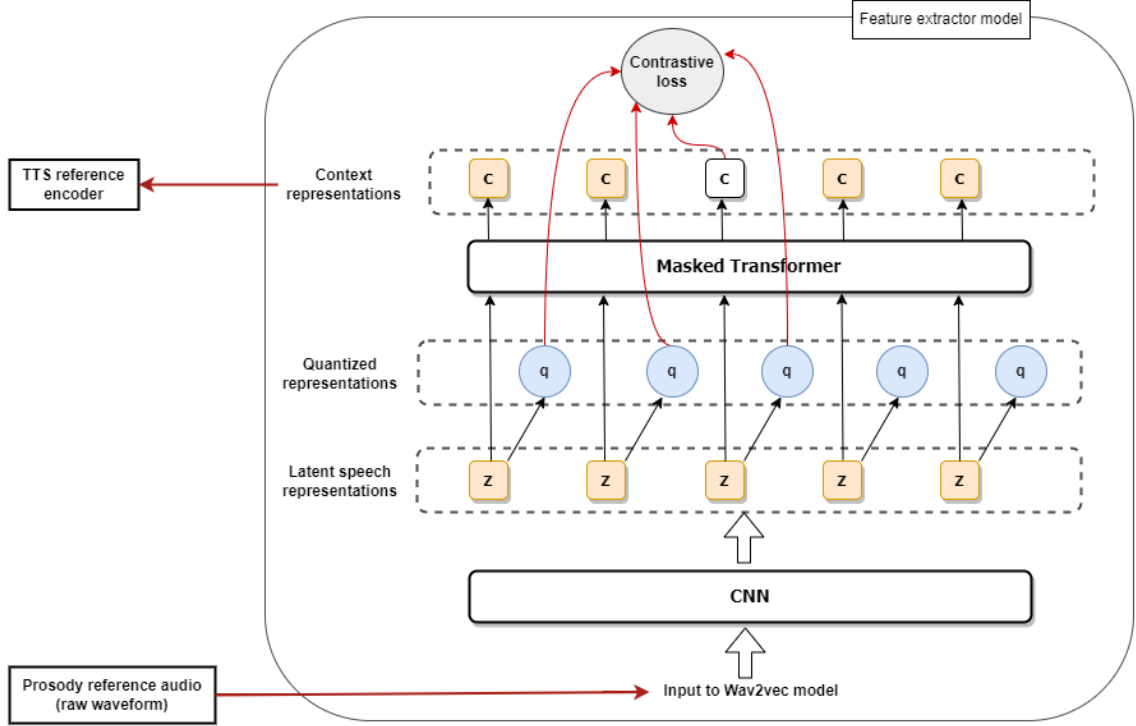


Figure 10: The general framework of wav2vec technique and its utilization as a feature extractor for generating speech representations as input to the TTS model.

audio features into quantized finite set of discrete representations called tokens. This is done using a quantization module that maps the continuous feature vectors into a discrete set of tokens from a learned codebook. As those tokens are more abstract, they reduce the complexity of the features by retaining important features while filtering out all the irrelevant information. Because of that abstraction, it is harder to reconstruct audio from the wav2vec features, which means leakage of linguistic content into feature vectors is significantly lower compared to other features such as MFCCs.

2.3.1.7 Orthogonality Loss

Studies [35, 36] propose a model with two separate encoders to encode speaker and emotion information through speaker and emotion classification loss, along with gradient inversion of the emotion classification loss in the speaker encoder. Additionally, to

disentangle the source speaker information from the emotion embedding, the emotion embedding is made orthogonal to the speaker embedding with an orthogonality loss shown in equation (3). An ablation study in [35] showed that applying an orthogonality constraint helped the encoders learn both speaker-irrelevant emotion embedding and emotion-irrelevant speaker embedding.

$$L_{orth} = \sum_{i=1}^n \|S_i - e_i\|_F^2 \quad (3)$$

where $\|\cdot\|_F$ is the Frobenius norm, e_i is the emotion embedding and s_i is the speaker embedding.

2.3.2 Inference without Reference Audio

A main drawback of the unsupervised approaches (Section 2.2) is that they require a reference audio for the desired prosody or style of the generated speech. However, prosody references are not always available for the desired speaker, style, or text. Besides, using prosody reference introduces the leakage problem as discussed in Section 2.3.1. As a result, different techniques have been proposed that enable unsupervised expressive speech synthesis without prosody references. Some techniques utilize the reference audio at training phase while at inference phase speech synthesis can be done with or without a reference audio. Other techniques depend on input text only to generate prosody embedding at both training and inference phases. In the following three sections, we will describe techniques for inference without reference audio applied with each of the three main unsupervised ETTS approaches. In Section 2.3.2.4, we will discuss some ETTS approaches that are based on text only.

2.3.2.1 Direct Reference Encoding without Reference Audio

In several studies, prosody predictors are trained jointly with the proposed reference encoder to bypass the requirement for reference audio at inference time. The prosody predictors are trained to predict either the prosody embeddings generated by reference

encoders [76, 94, 95, 86], or the acoustic features used as input to reference encoders [43, 41]. As input to these prosody predictors, most studies utilize the phoneme embeddings [76, 43, 41, 86].

Alternatively, features extracted from input text can also be used as input for prosody predictors. In [94], the prosody predictor has a hierarchical structure that utilizes contextual information at both the sentence and paragraph levels to predict prosody embeddings. The input features for this predictor are in the form of 768-dimensional phrase embeddings extracted by the pre-trained language model XLNet [96].

Furthermore, in [97], emotion is modelled at three levels: global, utterance, and syllable (local). The model employs three prosody encoders, each with a predictor trained to predict the corresponding prosody embedding based on the output from the text encoder of the TTS model. In paper [98], the prosody predictor proposed learns multiple mixed Gaussian distributions model (GMM) for prosody representations. As input, the predictor receives two phoneme-level sequences including embeddings from the text encoder and embeddings from a pre-trained language model. Similar work is proposed in [99] where only phoneme embeddings are used as input to the prosody predictor. GMM in both studies is modeled via the mixture density network [100].

2.3.2.2 VAE-Based Approaches without Reference Audio

Sampling from the latent space without reference audio results in less controllability of style. In addition, it can also introduce naturalness degradation and inappropriate contextual prosody with regard to the input text [15, 101]. Therefore, to avoid sampling the latent space without a reference, authors of [102] proposed utilizing the same prosody embedding of the most similar training sentence to input sentence at inference time. The selection process is based on measuring cosine similarity between sentences’ linguistic features using either 1) calculating the syntactic distance between

words in the sentence using constituency trees [103], 2) averaging the contextual word embeddings (CWE) for the words in the sentence using BERT, or 3) combining the previous two methods.

Other studies approach the problem in alternative ways, seeking to enhance the sampling process either through refining the baseline model structure or by incorporating text-based components into the baseline. Regarding the improvement of the baseline structure, study [15] suggests the combination of multiple variational autoencoders to generate latent variables at three distinct levels: utterance-level, phrase-level, and word-level. Furthermore, they apply a conditional prior (CP) to learn the latent space distribution based on the input text embedding. To account for dependencies within the input text, they employ Autoregressive (AR) latent converters to transform latent variables from coarser to finer levels.

An alternative approach is proposed in [104] by replacing the conventional VAE encoder with a residual encoder that leverages phoneme embedding and a set of learnable free parameters as inputs. With this modified structure, the model learns a latent distribution that represents various prosody styles for a specific sentence (i.e., the input text), in addition to capturing potential global biases within the applied dataset (represented by the free parameters).

Similar to 2.3.2.1, some studies propose training a predictor for the latent prosody vectors based on features extracted from the input text [73, 61]. The proposed model in [73] generates fine-grained prosody latent codes of three dimensions at phoneme-level. These prosody codes are then used to guide the training process of a prosody predictor that receives phoneme embeddings as input, in addition to emotion and speaker embeddings as sentence-level conditions. In [61], the predicted mean values of the latent space distribution are employed as prosody codes. Similarly, a prosody predictor is trained to predict these prosody codes using two text-based inputs, including sentence-level embeddings from a pre-trained BERT model and contextual

information considering BERT embeddings of a few of surrounding k sentences given the current sentence.

Alternatively, study [101] proposed training a sampler, i.e., Gaussian parameters, to sample the latent space using features extracted from the input text. The applied text-based features include either BERT representations of a sentence (semantic information), the parsing tree of the sentence (syntactic information) after it is fed to a graph attention network, or the concatenation of outputs from the previous two samplers.

2.3.2.3 GST-Based Approaches without Reference Audio

In the same direction, some GST-TTS models utilize text-based features from pre-trained language models such as BERT to guide expressive speech synthesis at inference time without a reference. For instance, in [17], the training dataset is labeled with short phrases that describe the style of the utterance and are known as style tags. A pre-trained Sentence BERT (SBERT) model is used to produce embeddings for each style tag as input to a style tag encoder. The style embedding from the GST-TTS model is used as ground truth for the style tag encoder. During inference, either a reference audio or a style tag can be used to generate speech.

Alternatively, pre-trained language models are used to extract features from input text and train a prosody predictor to predict the style embedding based on these text-based features [105, 106, 62, 107, 64, 108]. In [105], the baseline model [11] is extended with a prosody predictor module that extracts time-aggregated features from the output of the baseline text encoder. Two pathways are suggested for the targets of the predictor output: either using the weights of the GSTs or the final style embedding. Similarly, in [106], two prosody predictors are investigated, using different inputs from a pre-trained multi-language BERT model. While the first predictor utilizes BERT embeddings for the sub-word sequence of input text, the other predictor employs only

the CLS token from the sentence-level information extracted by the BERT model.

The multi-scale GST-TTS proposed in [64] which employs three style encoders, also introduces three style predictors that employ hierarchical context encoders (HCE). The input to the first predictor is the BERT sub word-level semantic embedding sequence. The attention units in the HCE, however, are used to aggregate the resulting context embedding sequence from lower level as input to higher-level predictors. BERT embeddings are also used in [62] but at word-level and are passed as input to the proposed prosody predictor. The style embedding which is generated via word-level GSTs is used to guide the prosody predictor during model training.

A Context-aware prosody predictor is proposed in [108] which considers both text-side context information and speech-side style information from preceding speech. This predictor comprises two hierarchical components: a sentence encoder and a fusion context encoder. The context-aware input to the predictor includes word-level embeddings from XLNet [96] for each word in the current sentence, as well as the N preceding and following sentences. The sentence encoder focuses on learning low-level word meanings within each sentence, while the fusion context encoder captures high-level contextual semantics between the sentences. Additionally, style embeddings from previous sentences are integrated into the fusion context encoder input to account for speech-side information.

In [107], speech emotion recognition model (SER) is employed in the GST-TTS model as a style descriptor to learn the implicit connection between style features and input text. Deep style features for both synthesized speech and reference speech are obtained from a small intermediate fully connected layer of a pre-trained SER model during training. At inference time only text is used to synthesize expressive speech.

2.3.2.4 *ETTS Approaches Based Only on Text*

This category involves approaches that depend solely on input text to obtain prosody-related representations/embeddings during TTS model training. Several features related to speech prosody have been proposed by various studies for extraction from input text and subsequent transmission to a DNN-based module to generate prosody representations. For instance, the features extracted by the pre-trained language models can capture both semantic and syntactic relationships with the input text, making them effective representations for prosody.

In [109], input text word-level embeddings are extracted by the Embeddings from Language Models (ELMo) model [110] and used to generate context-related embeddings via a context encoder. Similarly, in [111], BERT is employed to extract embeddings for utterance sentences and pass them to a specific context-encoder to aggregate these embeddings and form a final context vector.

Other studies, such as [112, 113, 114], utilize graph representations of input text, which can also reflect semantic and syntactic information about the given text. In [112], the graphical representations of prosody boundaries in Chinese text are passed to a graph encoder based on Graph Neural Networks (GNN) to generate prosodic information for the input text. The prosody boundaries of the Chinese language can be manually annotated or predicted using a pre-trained model.

In contrast, [114] combines BERT-extracted features for input text with its graph dependency tree to produce word-level prosody representations. Specifically, the input text is passed through both BERT and a dependency parsing model to extract the dependency tree for word-level BERT embeddings. A Relational Gated Graph Network (RGGN) is used to convert this dependency tree into word-level semantic representations upon which the decoder of the TTS model is conditioned.

In [113], an emotion lexicon is used to extract word-level emotion features, including VAD (valence, arousal, dominance) and BE5 (joy, anger, sadness, fear, disgust).

Additionally, the [CLS] embedding by BERT for each utterance is also extracted where all the obtained features are passed to a style encoder to produce the style embedding. Alternatively, sentiment features of the input text can be used as prosody embedding to produce expressive speech. For instance, in [115], prosody embedding is produced by the Stanford Sentiment Parser which generates sentiment probabilities based on the tree structure of the input sentence.

Other models train a prosody encoder/predictor with the TTS model to predict some prosody related features utilizing text-based inputs. The text-based input to these prosody predictors in most of the studies is the text’s character/phoneme embeddings [116, 117, 118, 119, 120], while some studies use features extracted from the input text [121, 30] such as features from language models.

As for output of prosody predictors, several prosodic features are proposed. For example, the prosody predictor in [116] predicts a set of utterance-wise acoustic features, including log-pitch, log-pitch range, log-phone duration, log-energy, and spectral tilt. In [117], a continuous pitch representation is produced by a pitch predictor outputs. In studies [118, 120], three prosody-related features are predicted including F0, energy, and duration.

2.3.3 Prosody Controllability

Text-to-speech is a one-to-many mapping problem, i.e., for one piece of text there could be many valid prosody patterns because of speaker-specific variations. Accordingly, providing a kind of controllability over prosody-related features in synthesized speech is essential for generating expressive speech with different variations. However, it’s not always easy to mark-up prosody or even to define boundaries between prosody events, i.e., duration boundaries can vary depending on segmentation, pitch contour prediction is error-prone, and prosody features may not always correlate well with what listeners perceive.

Several studies in literature have addressed the controllability issue in terms of selecting an emotion/style class or intensity level and adjusting prosody-related features at different speech levels. In this section, we discuss studies considering prosody controllability.

2.3.3.1 Modeling Specific Prosody Styles

This group of studies provides individual representations of expressive styles/emotions, enabling the control of prosody in synthesized speech by offering the ability to select from available representations or adjust their values. In some studies [122, 95, 123], style is modeled at a single speech/text level, while in other studies [63, 15, 16] a multi-level or hierarchical model of expressive styles is used to allow for a better capture of prosody variation in expressive speech.

In single-level prosody modeling approaches, [122] is one of the early studies that extends a baseline with fine-grained control over the prosody of synthesized speech. The proposed modification involves adding an embedding network with temporal structure to either the speech-side (frame-level) or text-side (phoneme-level) of the TTS model. Accordingly, the resulting prosody embedding is of variable length, and it is used to condition input to either encoder or decoder based on the position of the embedding network.

Single-level prosody embeddings can be converted into discrete embeddings as in [95, 123]. Discrete prosody representations are easier to control and analyze and provide a better interpretation of prosodic styles. In [95], a word-level prosody embedding is proposed based on decision trees and a GMM. First, a binary decision tree is employed to cluster word-level prosody embeddings from reference audio based on their phonetic identities. Then prosody embeddings of each leaf can be clustered via a GMM model where clusters represent the available prosody tags in the leaf. At inference time, appropriate prosody tags can be selected manually or via a prosody

predictor.

In [123], a character-acting-style extraction module is used to extract different character acting styles from the input speech. Discrete character-level styles are obtained via vector quantization (VQ) [59], which maps them to a codebook. A style predictor based on text can be used at inference to select from these discrete character-acting-styles.

Regarding multi-level prosody modeling, some studies propose enhancing prosody control in the baseline models [10, 13, 11] by modifying their single-level prosody modeling to multiple levels. For instance, [63] proposes a hierarchical structure of [11] with multiple GST layers which were found to yield better tokens interpretation. Baseline model [13] is also extended in [15] with three VAEs to generate three different levels (utterance, phrase, and word) of latent variables with varying time resolutions.

2.3.3.2 Modeling Specific Prosody Features

This group of studies provides individual representations of prosody-related features. Control over prosody of the synthesized speech is provided via selecting or adjusting a specific representations of a specific prosody-related feature. Some studies in this direction model prosody features at the global or utterance-level [75, 71], while other studies propose modeling at fine-grained levels [118, 41, 124, 125, 117], such as phoneme, syllable, or word-level.

The STYLER model [75], for example, employs multiple style encoders to factor speech style into several components, including duration, pitch, speaker, energy, and noise. This structure enables STYLER to generate controllable expressive speech by adjusting each of the individually modeled features. FastSpeech2 [9] provides fine-grained control over pitch range, duration, energy through a phone-level variance adaptor. [71] improves FastSpeech2 with an utterance-wise prosody model using an additional variance adaptor that models five features at the utterance-level: pitch,

pitch range, duration, energy, and spectral tilt. The utterance-wise prosody model enables easier control of prosody while still allowing modification at the phone-level.

In the same direction, [41] proposes two prosody modeling levels: the word-level and utterance-level. while global prosody embedding is the emotion embedding obtained by a reference-based encoder, the local prosody embedding is obtained via a predictor of the F0 features at the word-level. More flexible control of the F0 feature is provided in the controllable deep auto-regressive model (C-DAR) model [124] which allows for F0 contour adjustment by the user. Three strategies are used for adjustment including: 1) context awareness by conditioning with the preceding and following speech, 2) conditioning on some random segments of ground truth F0, and 3) predicting F0 values in reverse order.

Discrete fine-grained representations of prosody features as in [125, 117] via VQ are also applied to limit the number of the obtained representations. VQ in [117] are applied on character-level pitch prosody representations while in [125], it is used to produce discrete syllable-level prosody embeddings. The resulting prosody pitch codes in [117] can be adjusted via a specific bias value to generate speech with different pitch accents. In [125], codes are representing some prosody features such as pitch and duration at syllable-level which allow for prosody adjustment though manual selection from codebook.

2.3.3.3 Modeling Prosody Strength

This group of studies focus on regulating the strength of emotion or prosody. For instance, [126] utilizes the distance between emotion embeddings and the neutral emotion embeddings to identify scalar values for emotion intensity control. Another method for learning emotion strength values in an unsupervised manner is by using ranking functions. Studies [127, 40, 97, 30] utilize a ranking function-based method named relative attributes [32] for this purpose.

[97] trains an emotion strength extractor at the syllable-level based on input speech utilizing the ranking function. Alternatively, the reference encoder in [40] functions as a ranking function to learn a phoneme-level emotion strength (descriptor) sequence. The OpenSMILE [39] tool is used to extract 384-dimensional emotion-related features from phoneme fragments of reference speech as input to the ranking function. Similarly, the proposed ranking function in [127] takes a set of acoustic features extracted from the input speech via OpenSMILE tool but at the utterance-level as input.

In [30], both emotion class and emotion strength value are obtained via a joint emotion predictor based only on the input text. The input to the predictor is features extracted from input text via the Generative Pre-trained Transformer (GPT)-3 [31]. The emotion strength values which are also obtained via [32], are used as ground truth for predictor training.

Another ranking method is proposed in [74] using the ranking support vector machine. The model generates style embeddings via reference encoders and represents each style with its embeddings centroid at inference time. However, a linear SVM is trained with the model to provide the ability for style embedding adjustment. The SVM model learns a hyperplane between neutral emotion and a specific emotion embeddings which is utilized to move(scale) the style vectors based on their distance from the hyperplane.

2.3.3.4 Prosody Clustering

In this section, methods for selecting the appropriate prosody embedding for referenced-based ETTS models are described. In [128, 129], clustering methods are utilized to generate representative prosody embeddings for each emotion class when the GST-TTS model is trained with a labeled dataset. Generally, these models apply clustering on the resulting emotion embeddings in 2D space. In [128], the centroid of each cluster is used as the weights of the GSTs while in [129], the weight vector is obtained

by considering the inter and intra distances between emotion embeddings clusters.

Similarly, clustering algorithms are applied in [130, 131] to achieve discrete prosody embeddings but for two specific prosody-related features. The two studies employ K-means algorithm to cluster F0 and duration features extracted for each phoneme. The centroids of the clusters are then used as discrete F0 and duration values/tokens for each phoneme.

In [132], a cross-domain SER model with the GST-TTS model is proposed to obtain emotion embeddings for an unlabeled dataset. The cross-domain SER model is trained using two datasets including: 1) an SER dataset (source) labeled with emotions, and 2) a TTS dataset (target) that is not labeled. In details, the SER model trains an emotion classifier that generates soft labels for the unlabeled TTS dataset. These soft labels are then used as ground truth to train an extended version of baseline GSTs with an emotion label predictor based on weights of GSTs as input.

2.3.4 Speech Synthesis for Unseen Speakers and Unseen Styles

Building a speech synthesis model that supports multiple speakers or styles can be achieved by training TTS model with a multi-speaker multi-style dataset. However, generating speech for an unseen speaker or style is a challenging task for which several solutions have been proposed in the literature. A popular approach is to fine-tune the averaged TTS model with some samples from the unseen target speaker or style. The fine-tuning process may require a single sample from the unseen speaker or style (referred to as one-shot models) or a few samples (referred to as few-shot models). There are also models that do not require any fine-tuning steps, and these are known as zero-shot TTS models.

In [131], the proposed fine-tuning process focuses on sentences used in the process to ensure phonetic coverage, meaning that each phoneme should appear at least once in these sentences. The proposed model requires about 5 minutes of recordings from

the unseen target speaker to clone the voice and allow for manipulation of some voice features such as F0 and duration.

Another approach to address the problem of unseen data is to employ specific structures in the TTS model. In [133] for instance, a cycle consistency network is proposed with two Variational Autoencoders (VAEs) and two training paths: a paired path and an unpaired path. The unpaired path refers to training scenarios where the reference audio differs from the output (target) speech in terms of text, style, or speaker.

In [134], a U-net structure is proposed for the TTS model to support one-shot speech synthesis for unseen styles and speakers. The U-net structure is used between the style encoder and the mel decoder of the TTS model, with an opposite flow between them. The decoder receives phoneme embedding and produces the Mel-spectrogram as output. In parallel, the style encoder receives the reference audio and produces its linguistic content with guidance from the content (text) encoder.

Lastly, recent TTS models based on in-context learning [68, 67, 1] all share the capability to perform zero-shot speech synthesis, as explained in Section 2.2.4. In fact, the in-context training strategy underlies the ability of these models to synthesize speech given only a style prompt with the input text. Specifically, the synthesis process treats the provided prompt/reference as part of the desired output speech. Therefore, the model’s goal is to predict the rest of this speech in the same style as the given part (prompt) and with the input text.

2.3.5 Lack of Labelled Expressive Datasets

A main limitation in the domain of expressive speech synthesis is the inadequate availability of expressive speech datasets. Although there are several emotional and expressive speech datasets publicly available, they still fall short in terms of size, accuracy, and diversity required to train effective ETTS models. On the other hand,

we noticed numerous internal expressive speech datasets that are utilized in many studies in literature. Some of these datasets are of large size and exhibit good quality, with high diversity, including multiple speakers, styles, and emotions but they are not open to the research community.

As a result, current ETTS models still suffer from performance degradation and poor generalization. Additionally, there are open expressive datasets, such as the Blizzard datasets [135], which are larger in size but lack any labels. Furthermore, widely used TTS datasets, such as LJSpeech [136] with a single speaker, and VCTK [137], and LibriTTS [138] with multiple speakers, are also employed in unsupervised expressive TTS models to address prosodic features generally, control issues, or learning different speakers’ styles.

Constructing an expressive speech dataset is, in fact, a demanding endeavor compared to collecting neutral speech datasets, due to several factors. First of all, differences among speakers in portraying different speech styles or emotions pose the first challenge. Second, there are variations in emotional interpretation among different listeners who annotate the same expressive speech, which can impact the accuracy and consistency of these datasets. Moreover, the wide range of human emotions and speaking styles introduces further complexities.

Additionally, when considering different languages and multiple speakers, the challenge becomes more intricate. However, with the new trend that introduces the language modeling approach to the field of speech synthesis, TTS models are trained on a large amount of data using in-context learning strategy. These models can provide diversity in speakers, speaking styles, and prosodies, however, they require large amount of data for training and consequently high computational resources.

2.3.6 Inefficient Performance Evaluation Metrics

While evaluation metrics applied for general TTS models focus on speech quality in terms of intelligibility and naturalness, the assessment of ETTS models' performance goes beyond that, focusing on other aspects. Evaluation metrics of ETTS models need to measure aspects like emotion or style expressiveness, prosodic features, and controllability over all these aspects.

In the literature there are some metrics that have been proposed to evaluate performance of ETTS models. For instance, emotion and style classifiers/recognition (SER) models as in [128, 30, 129, 74] which are used to measure classification accuracy, reflecting the efficiency of the proposed model in generating emotional speech. Furthermore, visualization and plotting of different prosodic features, variables, or embeddings are also employed in several studies [127, 125, 132, 118, 41, 131] to evaluate the expressivity of generated samples and compare different approaches or synthesizing scenarios. Additionally, ablation studies as in [75, 74] have also been conducted to measure the effectiveness of each component in the proposed model and how it affects the expressivity of generated speech.

Despite existence of such metrics to assess the ETTS models performance, no general and standard metrics have been identified to facilitate the comparison process among different approaches. Furthermore, since the evaluation of expressive speech is more sophisticated and sensitive compared to TTS models, there is still a high demand for more accurate metrics capable of capturing various aspects of expressiveness in speech. Additionally, with the increased attention on building controllable ETTS models, the need arises for efficient evaluation metrics for controllability related aspects.

2.4 Thesis Motivation

Although TTS models have made significant strides, approaching human-like speech synthesis, expressive TTS models still require considerable refinement to achieve effective expressive speech synthesis and overcome the aforementioned challenges in this field. Language modeling-based approaches could potentially revolutionize the field of ETTS, addressing many of the challenges mentioned earlier, but they are still in their early stages in the field. Furthermore, developing such in-context learning based approaches depends heavily on availability of huge amount of data and high computational facilities which limit work on those approaches to large companies.

Based on literature, the unsupervised ETTS approaches are constrained with availability of style reference/prompt audio at inference time. As a result, several studies have been working on overcoming this challenge by replacing features based on the reference audio with input text-based features. On the other side, the supervised ETTS approaches are based on labelled speech datasets. However, building labelled expressive/emotional speech datasets is a challenging process with several obstacles.

Accordingly, in this thesis we are investigating the capability of certain deep neural-based approaches to improve the performance of supervised ETTS models. We aim at enhancing the generated speech with these approaches in terms of emotion expression, while utilizing only limited amount of data.

We investigate two main approaches including: 1) voice conversion models, and 2) ensembles of discriminators and emotion classifiers to the ETTS model. In the first approach we propose to employ voice conversion models as emotion booster to the synthesized speech from ETTS models.

In the second method, we suggest incorporating emotion classifiers and discriminators (adversarial training) into the ETTS model for enhanced emotional speech synthesis. Existing literature employs emotion classifiers and adversarial training in

various manners to booster the ETTS model. Our research introduces the use of ensembles consisting of these two components, each processing different input features. We propose employing two types of input features: firstly, pitch features which is highly associated with speech expressiveness according to existing literature. Secondly, emotion-related features extracted by a variational autoencoder as an efficient network for extracting representative feature.



CHAPTER III

VOICE CONVERSION FOR EMOTIONAL SPEECH SYNTHESIS

Voice conversion (VC) models are employed to alter the voice of one individual to mimic another's, maintaining the original linguistic content. Conversely, Emotional Voice Conversion (EVC) involves transforming speech from one emotional state to another, all within the same speaker and utilizing the same textual input.

We propose to combine an emotion voice conversion model to the expressive TTS model to enhance its emotion expressivity in the generated speech by this model as illustrated by Figure 11. In other words, we propose to utilize pairs of acoustic features (i.e., mel-spectrogram) for emotional speech from ETTS model and the original emotional dataset and use them as training pairs for an EVC model. The voice converter in this case, is actually working as emotion booster on the synthesized speech from the ETTS model.

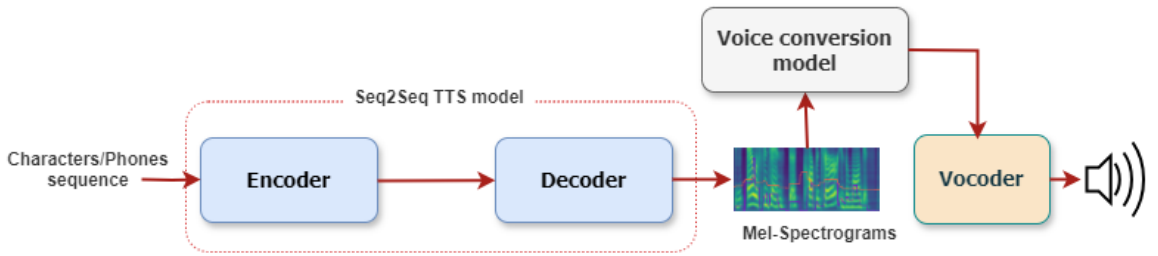


Figure 11: First proposed model for enhanced emotional speech synthesis. The model combines an emotional voice conversion model to ETTS model where output of ETTS is used as input to the voice converter.

At first, the ETTS model need to be trained separately with the applied dataset to be able to generate synthetic data. After that, the trained ETTS model is used to generate a synthetic copy of the dataset from textual part of the data. The resulting

synthetic data with the original speech data are then used to train an EVC model for better synthesized emotional speech.

As a result, we will need to train a single EVC model to convert synthetic data of each emotion to the corresponding emotion original data. Moreover, if we use multi-speaker multi-emotion dataset, then we should train an EVC model for each speaker-emotion pair of the data. We propose to use two of the latest voice conversion models which we will describe in the following two sections.

3.1 *CycleGAN-Based Voice Conversion Model*

Voice conversion models typically undergo training using parallel data from both sides of the conversion process. Essentially, this means that the training dataset for voice conversion should include pairs of identical utterances spoken by two distinct speakers or expressed in two different emotions by the same speaker. However, gathering parallel speech data is challenging and costly. Consequently, researchers have endeavored to develop voice conversion models trained with non-parallel data to address this hurdle. CycleGAN-based voice conversion models represent a category of converters proficient in achieving the conversion objective of voice conversion models without relying on parallel utterances.

CycleGAN-VC [2] is the base for this group of VC models that is based on two-direction mapping in the training process between the source ($x \in X$) and target ($y \in Y$) utterances as shown in Figure 12, and three loss functions, i.e., adversarial loss [139], cycle-consistency loss [140] and identity-mapping loss.

In adversarial loss as shown in equation (4), generator $G_{X \rightarrow Y}$ learns to convert source x to target y that is indistinguishable by discriminator D_Y . In contrast, discriminator D_Y tries not to be deceived by generator $G_{X \rightarrow Y}$ by correctly classifying fake from real targets. Therefore, the generator $G_{X \rightarrow Y}$ aims at minimizing adversarial loss in equation (4) while discriminator D_Y aims to maximize it. Similarly, for inverse

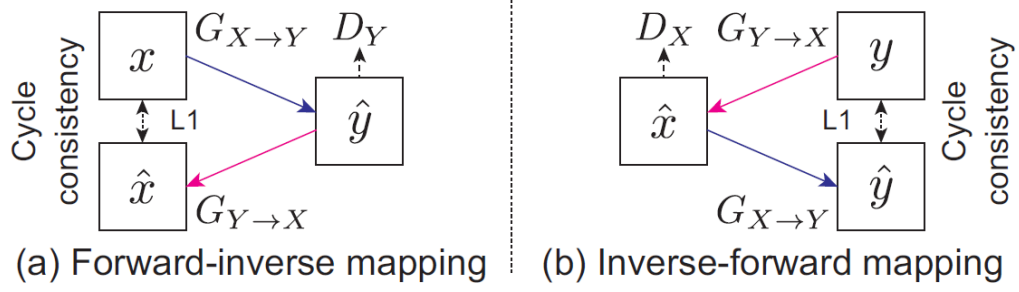


Figure 12: The two-direction mapping applied by CycleGAN training process utilizing non-parallel data. Reprinted from [2]

mapping, another adversarial loss is applied between the generator $G_{Y \rightarrow X}$ and the discriminator D_X .

$$L_{adv}(G_{X \rightarrow Y}, D_Y) = E_{y \sim P_Y(y)}[\log D_Y(y)] + E_{x \sim P_X(x)}[\log(1 - D_Y(G_{X \rightarrow Y}(x)))] \quad (4)$$

The cycle consistency loss given by equation (5) is introduced to stabilize training and to grantee the correct pairs (x, y) for forward and backward mappings $(G_{X \rightarrow Y}, G_{Y \rightarrow X})$:

$$L_{cyc}(G_{X \rightarrow Y}, G_{Y \rightarrow X}) = E_{x \sim P_X(x)}[\|G_{Y \rightarrow X}(G_{X \rightarrow Y}(x)) - x\|_1] + E_{y \sim P_Y(y)}[\|G_{X \rightarrow Y}(G_{Y \rightarrow X}(y)) - y\|_1] \quad (5)$$

The conversion process is enhanced with the third loss, i.e., identity loss that aims at preserving linguistic information during mapping process:

$$L_{id}(G_{X \rightarrow Y}, G_{Y \rightarrow X}) = E_{x \sim P_X(x)}[\|G_{Y \rightarrow X}(x) - x\|_1] + E_{y \sim P_Y(y)}[\|G_{X \rightarrow Y}(y) - y\|_1] \quad (6)$$

CycleGAN-VC2 [141] on the other hand, improves the adversarial loss of CycleGAN-VC with an additional step (named two-step adversarial loss) as a solution to reduce the over smoothing caused by $L1$ in the cycle-consistency loss. The additional step introduces a new discriminator for both adversarial losses and identifies adversarial loss on the reconstituted x and y through circular mapping $(x \rightarrow y \rightarrow x, y \rightarrow x \rightarrow y)$. CycleGAN-VC2 also introduces some improvements on both generator and discriminator structures.

In both models CycleGAN-VC and CycleGAN-VC2, application is limited to mel-cepstrum as input and it lacks capability to capture the time-frequency features, while most recent and efficient vocoders [142, 143] work on time-frequency structures, i.e., mel-spectrograms.

MaskCycleGAN-VC [3] is an extension to CycleGAN-VC2 to overcome this drawback by introducing a novel way for training the model known as filling in frames (FIF). In this type of training process, some frames of the input mel-spectrograms are hidden with temporal mask and the model tries to utilize the neighboring frames to fill in the missing (masked) ones as shown by Figure 13. Efficiency of MaskCycleGAN-VC over previous models in this group of VC comdels, motivated us to use it in our proposed approach for enhancing emotion expressivity in synthesized speech by ETTS models by VC models.

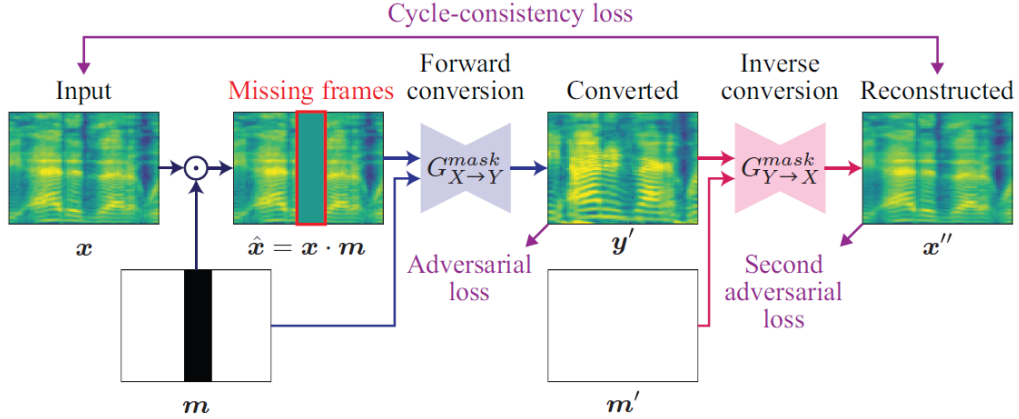


Figure 13: MaskCycleGAN-VC pipeline with filling in frames training (FIF) for the forward-inverse mapping. Reprinted from [3]

3.2 Seq2Seq-Based Voice Conversion Model

This is another efficient voice conversion model that is trained on limited amount of emotional data through 2-stage training strategy [4]. The model consist of main five components including: a text encoder, a seq2seq automatic speech recognition (ASR) encoder, a style encoder, a classifier, and a seq2seq decoder as shown by Figure 14.

We can say that the text encoder and the seq2seq decoder form together a seq2seq autoregressive TTS model. In fact, the structure of the decoder here follows the structure of well-known autoregressive TTS model called Tacotron [6, 7].

The idea of this model, is to leverage a TTS model for training an emotional VC with a limited amount of emotional data. The training process is done in two main stages: 1) style initialization, and 2) emotion training. In the first stage, the model is trained on a multi-speaker neutral TTS large dataset to enable the model to learn how to distinguish between speakers styles and linguistics content in the input speech.

In the second stage, the trained style encoder, having learned different speaking styles from the first stage, requires only a small amount of emotional data to learn emotional speaking styles. Consequently, the model is trained on a limited multi-emotion dataset to comprehend the various emotions present. This second stage allows the model to convert speech from one emotion to another using only minimal emotional data.

In detail, during the style initialization stage, both the text encoder and the ASR encoder focus on learning to predict linguistic features from input speech and text. Simultaneously, the style encoder learns to predict the style embedding of each speaker from input speech. The classifier in this stage acts as a speaker classifier based on linguistic features for enhancing the generation of speaker-independent linguistic embeddings by applying adversarial training. In this stage, speaker id is used as one-hot-vector for supervised training for both style encoder and speaker classifier.

In the second stage (emotion training), the style encoder transitions into an emotion encoder, fine-tuned with a limited emotion-labeled dataset to predict emotion embedding. Here, the classifier functions as an emotion classifier, working on removing emotion-related information from linguistic embeddings through adversarial training. In this stage, emotion id is used as one-hot-vector for supervised training for both emotion encoder and emotion classifier.

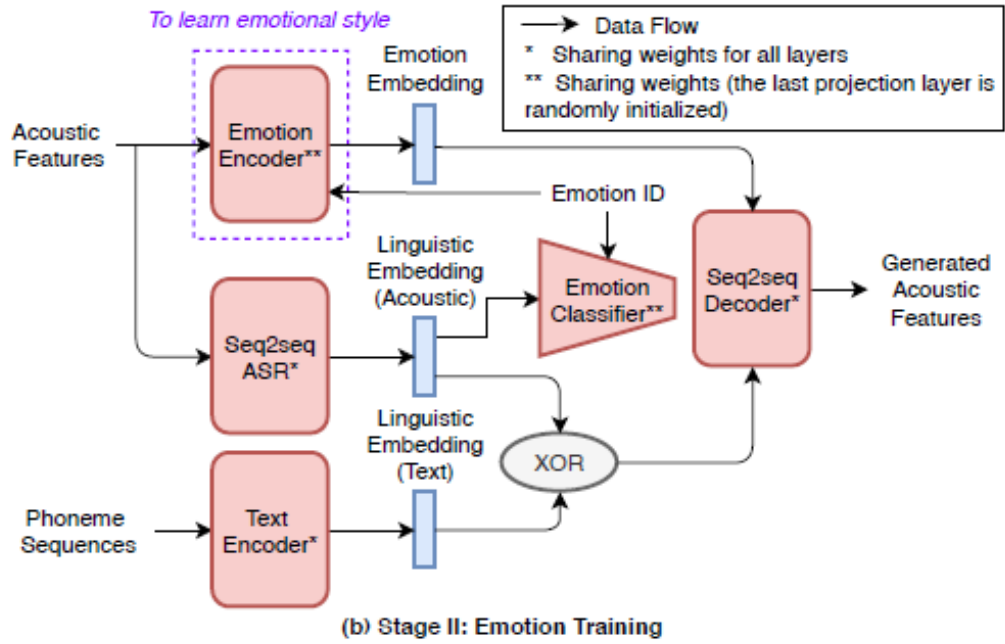
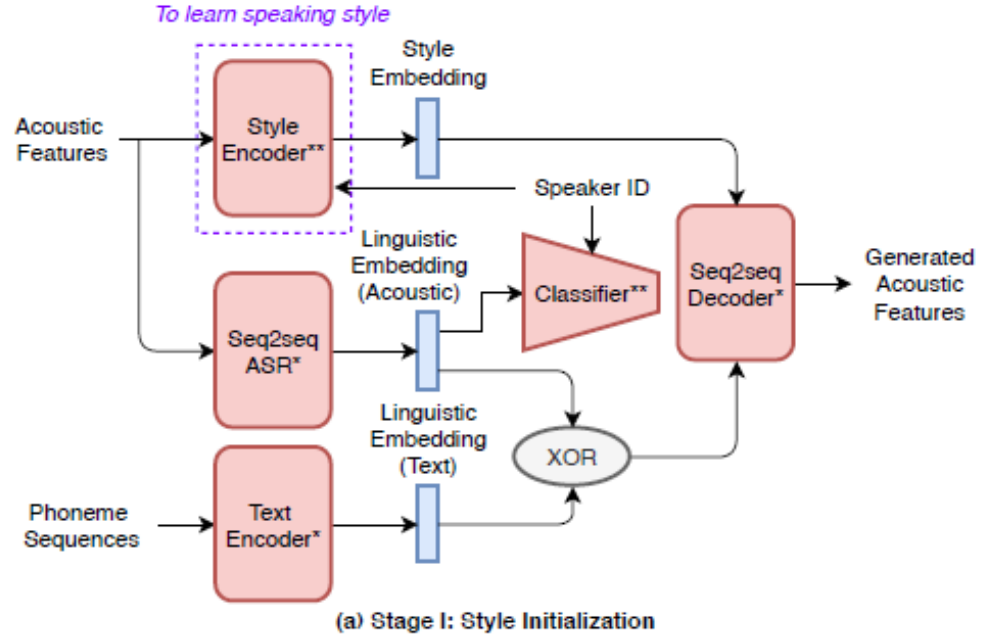


Figure 14: The 2-stage training of Seq2Seq voice conversion model. Repainted from [4]

At inference time, the emotion encoder takes first a set of reference utterances for the target emotion and generate an averaged emotion embedding representing

this emotion. The seq2seq ASR processes the source utterance requiring emotion conversion, extracting its linguistic features to produce the linguistic embedding. These two embeddings are then passed to the seq2seq decoder to generate the source utterance but in the target emotion.



CHAPTER IV

ENSEMBLES OF DISCRIMINATORS AND EMOTION CLASSIFIERS FOR EMOTIONAL SPEECH SYNTHESIS

In this chapter, we describe our proposed model that integrates two ensembles of emotion classifiers and discriminators, into a supervised ETTS model for better expressive/emotional speech synthesis. The classifiers are trained jointly with ETTS model while discriminators are used for applying adversarial training considering ETTS model as a generator.

The classifiers and the discriminators receive different inputs that are based on acoustic features (i.e., mel-spectrograms) generated by the ETTS model. The first classifier and discriminator receives a pitch sequence generated by a pitch extractor based on the synthetic mel-spectrograms from ETTS model as input. The second classifier and discriminator are fed with an emotion embedding produced by a variational auto-encoder (VAE) that also produces these emotion embeddings utilizing the synthetic mel-spectrograms as input. Figure 15 illustrates the proposed model with its different integrated components.

The main goal behind this structure of the proposed model is to enhance expressiveness of its synthesised speech utilizing joint training with emotion classifiers and also adversarial training with discriminators. The features used input to the components of the two proposed ensembles, are selected specifically as they are emotion-related and hence can booster the model capability of synthesizing emotional speech. In addition, employing two types of emotion-related features enrich model learning with variety prosodic features.

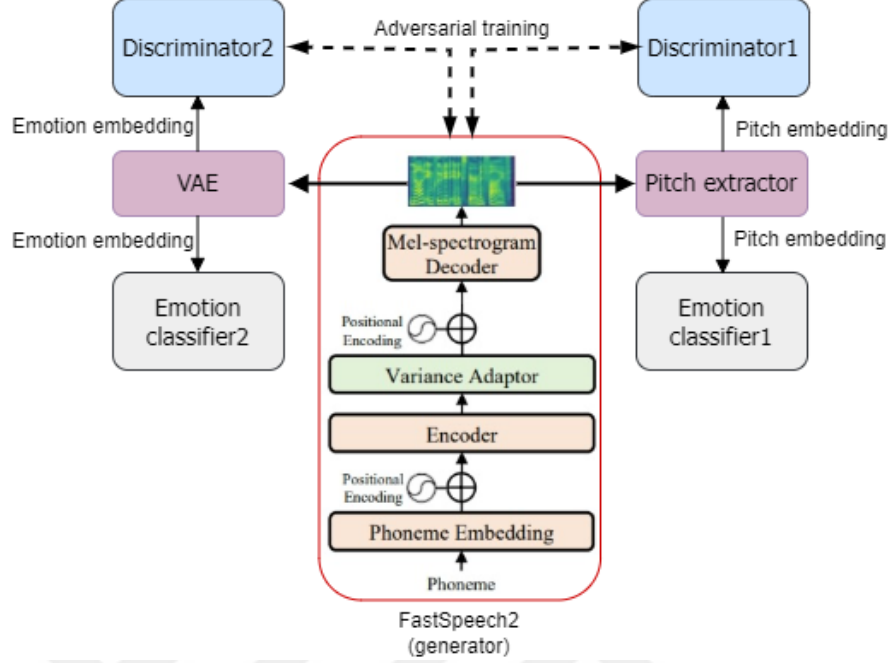


Figure 15: Proposed model of expressive TTS model with ensembles of discriminators and emotion classifiers.

4.1 *ETTS Model*

The backbone of our proposed model as an ETTS model is FastSpeech2 [9] which is a non-autoregressive model with encoder-decoder structure. Details of FastSpeech2 model are provided in Section (5.1.1). It is worth to mention here, that we have extended input of FastSpeech2 with two values representing speaker and emotion labels beside the input text to support multi-speaker and multi-emotion training dataset. The loss function of FastSpeech2 includes three loss terms for three predictors of three different features (i.e., pitch, energy and duration). Besides, a reconstruction loss term is added to the total loss for reconstructing the model output acoustic features (i.e., mel-spectrograms). FastSpeech2 loss function is shown by equation (7):

$$L_{fastspeech} = L_{mel} + L_{duration} + L_{pitch} + L_{energy} \quad (7)$$

Our proposed model introduces basically two ensembles each of an emotion classifier and a discriminator. The input to the first ensemble is pitch feature extracted from

the output of the baseline model. The second ensemble receives emotion embedding from a VAE model utilizing the output of the baseline model. In fact, our motivation behind using emotion classifiers and discriminators is to enhance the baseline model learning capability of expressiveness related features in two different ways.

Starting with the first ensemble, the two components of this ensemble aim at encouraging the baseline model to learn better pitch feature in the generated speech. Pitch feature is actually one of the most important and speech expressivity related feature that has been utilized in several studies [125, 124, 117, 75] for better expressive synthesized speech. When training the emotion classifier jointly with the baseline model, it will encourage the model to generate speech with better representative pitch feature that helps the classifier to predict the correct emotion. Discriminator on the other hand, applies adversarial training on the baseline model, which improves the model capability (as a generator) to produce speech with pitch feature similar to the ground truth feature.

Similarly, the second ensemble components enhanced the baseline model capability to produce expressive speech but this time focusing on another feature as input to the ensemble. The feature employed by this ensemble is an emotion embedding generated by a VAE model. The emotion embedding can be seen as a more generic feature summarizing more than one expressivity related features. As explained in Section 4.3, the VAE model is trained to generate emotion embeddings/vectors that involve specifically emotion discriminative features. The VAE model will utilize the output of the baseline model as its input. Accordingly, joint training of an emotion classifier using emotion embedding as an input, will motivate the baseline model to enhance emotion related features in the synthesized speech as output. In addition, adversarial training by the discriminator in this ensemble, will encourage the baseline model to generate speech from which the extracted emotion embedding is similar to the one extracted from the ground truth speech.

In general, the proposed model introduces six different components to the selected base ETTS model including: two emotion classifiers, two discriminators, a pitch predictor and a variation auto-encoder. In the following sub sections we will describe the details of each of the these different components and how they are integrated to base ETTS model.

4.2 *Pitch Extractor*

The pitch extractor is used to extract a frame-level pitch sequence from the synthetic mel-spectrograms generated by FastSpeech2 decoder. The extracted pitch is used as input to one emotion classifier and one discriminator that forms together the first proposed ensemble. To further utilize the pitch extractor for our target to enhance expressivity, we extended the total loss function of FastSpeech2 with an additional loss term ($L_{post-pitch}$) between the extracted and the ground truth frame-level pitch sequences. The extended loss function of FastSpeech2 is shown in equation (8):

$$L_{fastspeech} = L_{mel} + L_{duration} + L_{pitch} + L_{energy} + L_{post-pitch} \quad (8)$$

4.3 *Variational Autoencoder*

Variational auto-encoder (VAE) [144] is a generative model that is utilized to map its input from a higher dimensional space to a well-organized and lower-dimensional latent space. It has encoder-decoder structure and it is trained to generate latent representative vectors for its encoder. Latent vectors are then passed to the decoder to re-generate the VAE input. In other words, VAEs learns a Gaussian distribution (latent space) from which latent vectors can be sampled to represent the input data. We provide further details of VAE model structure in Section (5.1.2).

The training objective of VAEs consists of two main terms: reconstruction loss and KL-divergence loss as shown by equation (9). While reconstruction loss ensures ability of VAE to regenerate its input from latent vectors, the KL-divergence loss

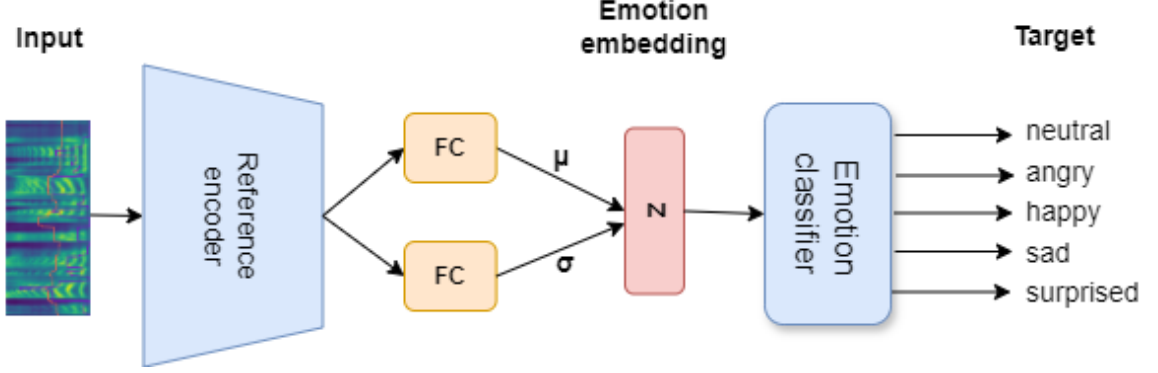


Figure 16: Modified variational autoencoder structure replacing the decoder with emotion classifier to learn emotion representative latent vectors.

ensures that the learned latent space follows a Gaussian distribution by minimizing KL-divergence between the two distributions.

$$L_{VAE} = L_{reconst} + L_{KL-div} \quad (9)$$

In our proposed model, we used a VAE model to learn representative latent vectors/embeddings for mel-spectrograms generated by the ETTS model. However, as we aim at learning learn emotion representative vectors, we replaced the VAE decoder with an emotion classifier and the reconstruction loss with a classification loss. Figure 16 illustrate the modified VAE architecture that we used in our proposed model. Equation (10) shows VAE loss that we applied to train the VAE model in our experiments.

$$L_{VAE} = L_{classification} + L_{KL-div} \quad (10)$$

4.4 Emotion Classifiers

The proposed model employs two different emotion classifiers based on output of the applied ETTS model. Each classifier predicts the emotion of the ETTS output based on extracting specific features from this output. Acquiring two different features,

specifically emotion-related features, as inputs to the classifiers aims at enhancing the ETTS model capability of learning variations of emotion-related prosody.

The first classifier predicts emotion labels based on the extracted pitch features by the pitch extractor (explained in Section (4.2)). As pitch features are strongly related to emotional speech expressivity, the first emotion classifier will push the ETTS model in the direction of more correct and accurate pitch generation of synthesized speech and hence better emotion expressivity.

The second classifier on the other hand, employs a variational auto-encoder to generate emotion-related features to be used as input. The VAE will first receive the generic acoustic features which are the output of ETTS model, and encode them into an emotion vector/embedding representing more specific emotion features. Emotion embeddings are then passed to the second classifier to predict emotion labels that are in this case, indirectly dependent on the output of the ETTS model. In particular, this classifier motivates the ETTS model to generate more expressive mel-spectrograms as these can produce more distinctive emotion embeddings when forwarded to the VAE network.

4.5 *Discriminators*

Another direction for improving emotion expressivity of the synthesized speech from the target ETTS model, is by applying adversarial training in model training. Adversarial training stands on a min-max game between two adversarial networks named Generator (G) and Discriminator (D). During the game, the generator G tries to learn generating target output y of the given input x from the training pairs in the dataset.

The discriminator on the other hand as the name indicates, is a classifier that learns how to discriminate between the real targets y and the fake ones z generated by the generator during the training process. The generator and the discriminator

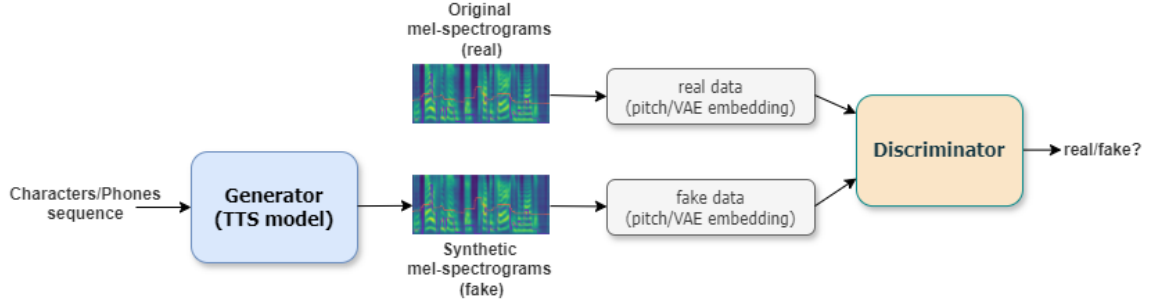


Figure 17: Applying adversarial training on TTS model which represent the generator in this process. The first discriminator receives pitch sequence as input. The second discriminator receive emotion embedding from VAE.

form together one model know as Generative Adversarial Networks (GANs) [139]. The loss function of the GAN model is binary cross entropy loss as shown by equation (11). During GAN training, the generator updates its parameters to minimize the loss function while the discriminator updates its parameters to maximize it.

$$\min_G \max_D \mathbb{E}_{x \sim p_{\text{data}}(x)} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [1 - \log D(G(z))] \quad (11)$$

Similar to emotion classifiers, we propose to use two discriminators utilizing the applied ETTS model as a generator as in Figure 17. In GAN game, the discriminator goal is to enhance the generator ability to produce better quality output. We used two discriminators with same structure but with different inputs from the generator (the ETTS model in this case). Using these two emotion-related features as inputs encourages the two discriminators to support the ETTS model (generator) to synthesize better speech in particular emotion prosody-related features in synthesized speech.

In fact, discriminator are binary classifiers that evaluates how close its input from the original (real) data. As our focus in this model is to improve the speech prosodic features, we employed one discriminator to enhance the pitch feature while the other discriminator improves more general emotion-related features. Specifically, pitch sequence and emotion embedding extracted from the ETTS output mel-spectrograms,

are fed to the two discriminators, respectively. Similar to the previously mentioned emotion classifiers, the frame-level pitch sequences are extracted by the pitch extractor (Section 4.2) and the emotion embeddings are generated by the VAE model (Section 4.3). The final loss of the proposed model is shown by equation (12):

$$\begin{aligned}
L_{model} = & L_{fastspeech} + \lambda_1 L_{adv-pitch} + \lambda_2 L_{adv-vae} \\
& + \beta_1 L_{cls-pitch} + \beta_2 L_{cls-vae}
\end{aligned} \tag{12}$$

Where $L_{fastspeech}$ is as in equation (8), L_{cls} is the loss of each emotion classifier and L_{adv} is the adversarial loss or the generator loss with respect to each corresponding discriminator. The symbols λ_1 , λ_2 , β_1 , and β_2 are the corresponding weights for each loss term and are identified experimentally as hyper-parameters.

CHAPTER V

RESULTS AND DISCUSSION

5.1 *Experimental Setup*

5.1.1 FastSpeech 2

FastSpeech [8] is a non-autoregressive TTS model mainly based on feed-forward transformer network (FFT) [145] and can synthesize speech with a high inference speed and quality. FastSpeech has an encoder-decoder structure that converts phonemes/characters to mel-spectrograms where both components consist of stacked FFT blocks. These two components are connected to each other via a length regulator in the middle, which handle the length differences between phonemes encoder and mel-spectrograms decoder. The length regulator works based on provided ground-truth durations of the phonemes to match them to the lengths of the output. At the same time, length regulator trains a duration predictor to predict these durations for the output acoustic features at inference time. FastSpeech first version depends on an auto-regressive teacher model in order to provide these ground-truth durations, which affects synthesized speech quality negatively.

FastSpeech2 [9] in the other hand, overcome this disadvantage by removing the teacher model and introducing a variance adapter to enhance FastSpeech architecture. The variance adapter beside the duration predictor used by length regulator, involves a predictor for pitch and energy features. The ground-truth values needed for training pitch and energy predictors can be extracted directly from the training dataset. As for duration predictor, the ground-truth durations in this case, are extracted via an external aligner such as Montreal Forced Aligner (MFA) [146]. In fact, durations extracted by external aligners have shown to be more accurate than those gained by

the complicated teacher model in the first version of FastSpeech. The loss function of FastSpeech2 is shown in equation (13). As for vocoder, we fine-tuned the pre-trained $16kHz$ universal HiFi-GAN vocoder [142] with ESD dataset using the official implementation repository [147] of HiFi-GAN.

$$L_{FS2} = L_{mel} + L_{duration} + L_{pitch} + L_{energy} \quad (13)$$

5.1.2 Variational Auto-Encoders

Variational auto-encoders [144] are generative models used for unsupervised learning. Their generative process is based on learning relation between continuous random latent variables z and the observed data x . The true posterior distribution $p_\theta(z|x)$ with parameters θ is intractable inference because of the intractable marginal likelihood $p_\theta(x)$. As a solution to this issue, VAE uses approximate posterior distribution $q_\phi(z|x)$ with ϕ parameters. The learning process can be seen as minimizing the distance between two distributions: $p_\theta(z|x)$ and $q_\phi(z|x)$ using the efficient algorithm Kullback–Leibler divergence [148] as shown in equation (14). After substituting true posterior according Bayes rule and re-arrangement of equation (14), the solution turns into maximizing the variational lower bound to the marginal log-likelihood of data. This alternative objective function is known as Evidence Lower BOund (ELBO) which is calculated with respect to both sets of parameters θ and ϕ as in equation (15).

$$D_{kl}(p_\theta(z|x), q_\phi(z|x)) = E_{q_\phi(z|x)}[\log \frac{q_\phi(z|x)}{p_\theta(z|x)}] \quad (14)$$

$$L(\theta, \phi; x) = E_{q_\phi(z|x)}[\log p_\theta(x|z)] - KL[q_\phi(z|x) || p_\theta(z)] \quad (15)$$

Where both prior distribution of latent variables z and the approximate posterior are modeled as Gaussian distributions as follows: $p(z) = N(z|0, I)$ and $q_\phi(z|x) = N(z|\mu_\phi(x), \sigma_\phi^2(x)I)$.

When using deep neural networks to build the model, two networks are used. The first network is the encoder which learns $\mu(x)$ and $\sigma^2(x)$ parameters via the given

data x . The second network is the decoder which decodes the latent variables z sampled from distribution learnt by encoder to reconstruct input data x . However, the stochastic part of the model (sampling z) obstruct the back-propagation process necessary for learning model parameters. To overcome this problem, the model uses a trick known as (re-parametrization trick) in which z is calculated as a fixed value based on μ and σ with addition small value $\varepsilon \sim N(0, I)$:

$$z = \mu + \sigma \odot \varepsilon \quad (16)$$

The two terms of the objective function shown in equation (15) known as reconstruction loss and KL-loss, respectively. A common problem during VAEs training is the early collapse of KL-loss to very small values (nearly zero), which affect the learning efficiency of the model. An annealing technique is used in this case for this part of the loss function to prevent its early collapse. The technique is two steps: first, to multiply kl-loss with a weight value that starts from 0 and increases gradually to 1 during training. Second, to consider kl-loss in the loss function every k iterations only and not in every iteration.

5.1.3 Baseline Model

As a baseline model for our proposed enhancements on ETTS models we used two main models: 1)FastSpeech2 with emotion embedding (FS2-EMBD), and 2)FastSpeech2 with variation autoencoder and emotion embedding (FS2-VAE-EMBD). Both models are supervised with emotion label which is used as input to the model. However, FS2-VAE-EMBD is also a reference-based model that requires emotion reference audio as input to VAE. As our applied dataset is multi-speaker, we also added speaker label as input to both models.

To implement FastSpeech2 in both models, we followed the implementation of this repository [149], while for VAE in FS2-VAE-EMBD model, we used implementation

in this [150] repository. We also applying gradient reversal layer (GRL) [37] with FS-VAE-EMBD to reduce leakage of speaker related information from reference audio to the generated style embedding from VAE encoder (reference encoder). For the first baseline model, the loss function is the same as FastSpeech2 model shown in equation (13). The loss function for the second baseline model (FS2-VAE-EMBD) includes additional term from the VAE loss which is KL-loss as shown by equation (17):

$$L_{BL2} = L_{FS2} + L_{GRL} + \lambda L_{KL} \quad (17)$$

Where L_{FS2} is FastSpeech2 loss function (or reconstruction loss). The term L_{GRL} is the adversarial loss used by the gradient reversal layer, and L_{KL} is the KL-loss of VAE as explained in Section (5.1.2). The hyperparameter λ_{KL} is used as control weight of term L_{KL} in the total loss and it is set to 0.001 in our experiments.

5.1.4 Dataset

To train the proposed model, we used the emotional speech dataset (ESD) which is a multi-speaker and multi-emotion dataset [151]. The script of ESD dataset consist of 350 different short utterances, where each utterance is recorded by 10 native English and 10 native Mandarin speakers (5 males and 5 females) in 5 different emotions (Neutral, happy, angry, sad and surprise). ESD audio clips are sampled at 16 kHz and the dataset is publicly available. We only use the English part of the dataset with an approximate size of 13 hours. We used about 0.9 of the data to train our proposed model while the rest (≈ 0.1) is used for model validation.

5.1.5 General Settings

All experiments for models in Section 5.2 are conducted on machine with 2x NVIDIA GeForce GTX 1080 GPU. For models in Section 5.3, we run the experiments on NVIDIA GeForce RTX 4090 GPU. We followed the baseline implementation (Section

5.1.3) for the selected values of the main hyper-parameters including learning rate 10^{-5} , batch size 16, and Adam optimizer.

5.2 Voice Conversion for Emotional Speech Synthesis

As we mentioned in chapter (3) our first proposed model for emotional speech synthesis is to combine a voice conversion model with the ETTS model. We investigated two main voice conversion models including MaskCycleGAN-VC and Seq2Seq-VC explained in Sections 13 and 14 respectively. In the following sections we will describe implementation, evaluation tests and results for these two proposed models.

5.2.1 MaskCycleGAN Voice Conversion with FastSpeech2

To implement the MaskCycleGAN-VC model, we used this [152] unofficial implementation of the model. As mentioned in Chapter 13, the emotional VC model can be trained only with the synthetic data from a single emotion and a single speaker. For the purpose of testing our proposed model, we trained four different VC models representing synthetic data from the two baselines mentioned in (5.1.3) each trained twice with data from two emotions of two different speakers. The selected speakers include speaker 17 (female) with surprised emotion and speaker 20 (male) with sad emotion. To evaluate these models, we basically conducted two listening tests that assess performance of baseline models individually and when combined with a VC in terms of emotion expression and speech quality. 10 English speakers participated in our tests where each rater listened in total to 56 different samples.

In the first test, our goal is to compare performance of the baseline models first in terms of emotion expression and speech quality. The test consists of several parallel samples from the two baseline models and raters are asked to choose samples that: 1)best expresses emotion and 2)has best quality, separately. The second test is XAB preference test to evaluate the enhancement made by the VC over each of the baseline models. In this test, we provide listeners with the original sample from the dataset

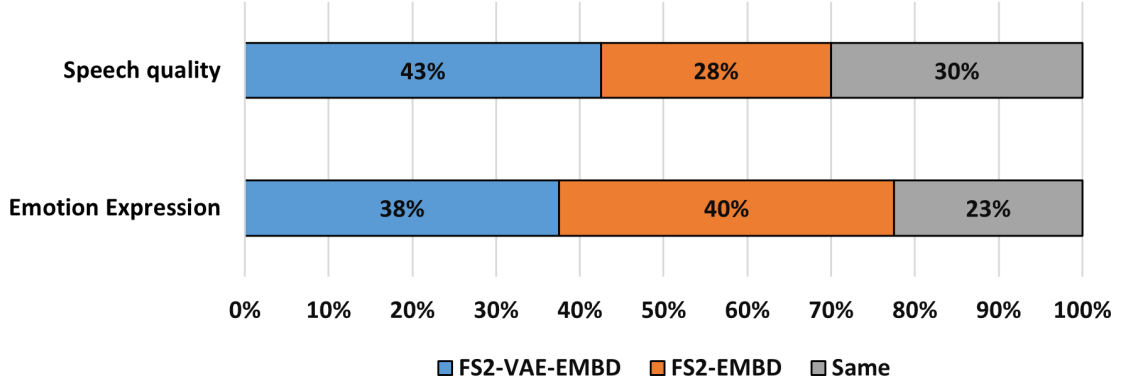


Figure 18: Results of emotion expression and speech quality preference test for the two proposed baseline models.

as a reference and two synthetic samples from the baseline model before and after being combined with the VC model. The participants in this test are asked to choose sample that: 1)sounds more similar to a given emotion reference and 2)has a better speech quality, separately.

In Figure 18, we show results of the first evaluation test which reflects the performance comparison between the two baseline models. The results show that in terms of quality, the second baseline model (FS2-VAE-EMBD) performs better which can be interpreted as a result parallel style transfer that is applied with the this baseline model at either training phase or inference phase. In parallel style transfer, the text of the audio used as a style reference is same as the input text to ETTS model to be synthesized. As mentioned before (Section 2.3.1), leakage of text related information from the reference audio into the synthesized speech is a main problematic issue with reference based ETTS models. However, in parallel style transfer this issue could be a supporting factor to the ETTS model to synthesize better (quality) speech.

In terms of expressing emotions, the first baseline (FS2-EMBD) exhibits a slightly better performance. This can be attributed to a slight potential conflict encountered by the second baseline (FS2-VAE-EMBD) between the provided style reference and the emotion embedding. While the emotion embedding encourages the model to

assimilate an average style for each emotion, the style embedding from VAE steers the model towards acquiring a specific style aligned with the given reference. Additionally, upon reviewing various samples from the original dataset, we observed samples where speakers portrayed single emotion samples differently or even misrepresented one emotion for another. In such scenarios, the baseline with emotion embedding may yield better results as it learns an averaged emotion embedding across all samples of a particular emotion. Besides, ESD dataset consist of mostly short sentences and thus emotion embedding can be considered enough to generate speech in the desired emotion, while with long sentences second baseline is expected to perform better. Generally speaking, the comparable evaluations of emotion expression between the two baselines suggests a certain degree of performance similarity between the two models when considering labeled data.

In Figure 19, we show results of the second test (XAB) that evaluates the performance of the baseline models with and without VC optimization. Obviously, we can see that VC improves performance of the two models in both sides emotion expression and speech quality. This ensures the feasibility of our proposal for enhancing emotion expression in emotional TTS systems by combining them with a VC model. As for speech quality, results are actually aligned with the first test results where second baseline is able to generate speech with better quality.

In terms of emotion expression, the second baseline model (FS2-VAE-EMBD) performance is improved with VC more then the first baseline (FS2-EMBD). We expect that this difference in emotion expression is actually because speech generated from the second baseline (FS2-VAE-EMBD) is more similar to the ground truth (reference) then speech from the first baseline. The VC model then improves emotion expression in speech from both baseline models. As second test is XAB test, the selection of listeners is based on similarity to the given reference. The second baseline model is probably able to imitate the style/emotion of the given reference sample

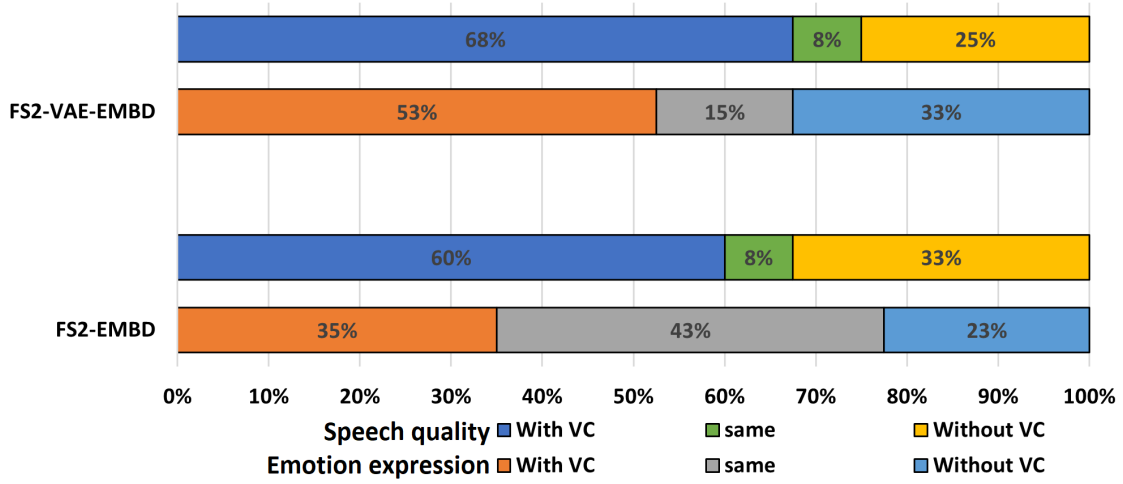


Figure 19: Results of emotion expression and speech quality (XAB) preference test for each of the proposed baseline models with and without MaskCycleGAN voice converter.

better than the other model because of the style reference given as input.

To summarize tests' results, it becomes evident that the second baseline model outperforms the first when a good reference for the desired style or emotion is available. Conversely, the first baseline model derives an averaged emotion embedding from each emotion set of samples, relying solely on the provided emotion label as input. Interestingly, this approach proves more effective in certain cases, where the generated speech appears to more accurately represent the intended emotion compared to scenarios where a reference is provided, as demonstrated in the first test.

In fact, there are some insights that can be also drawn from the tests' results, particularly, they encourage for further exploration in this direction to improve emotional speech synthesis. One potential avenue for improvement could involve experimenting with other voice conversion models within the context of ETTS systems, specifically those supporting many-to-many voice conversion such as [153, 154]. This approach would alleviate the need to train a separate VC model for each emotion data in the training dataset. On the other hand, the tests' results suggest that the impact of MaskCycleGAN-VC optimization is more pronounced on the quality of the generated

speech than on emotion expression. This interesting finding paves the way for exploring alternative ways to leverage VC with TTS models to enhance the overall quality of synthesized speech.

5.2.2 Seq2Seq Voice Conversion with FastSpeech2

We implemented the Seq2Seq-VC model using its official implementation from this repository [155]. We conducted two listening test to achieve our evaluation goals for this model including: 1)comparing performance of the baseline models extended with this VC model and 2)comparing performance of Seq2Seq-VC and MaskCycleGAN-VC when used with our ETTS baseline model. Similar to MaskCycleGAN-VC, we needed to train a single model to convert speech from synthetic data by the baseline to the original data for each single speaker and single emotion in the dataset. For evaluation purpose, we trained two models that convert synthetic speech from each baseline model to the original corresponding speech in the dataset for speaker 20 (male) for emotion sad.

In the first test we provided listener with samples from the two trained Seq2Seq-VC models and asked them to selected samples which better express emotion and samples with better quality, separately. As shown in Figure 20, the performance of Seq2Seq-VC with baselines is similar to MaskCycleGAN-VC where the second baseline (FS2-VAE-EMBD) performs better then the first baseline (FS2-EMBD) but with only slight difference in both emotion expression and speech quality.

Moving to the second test, its objective is to compare the performance of the two applied voice conversion models when integrated with our baseline model for emotional speech synthesis. Since both VC models exhibited improved performance with the second baseline (FS2-VAE-EMBD), the focus of this test is to evaluate the performance of these two models. Therefore, in this test, participants are presented with samples from both VC models combined to the second baseline, and are tasked

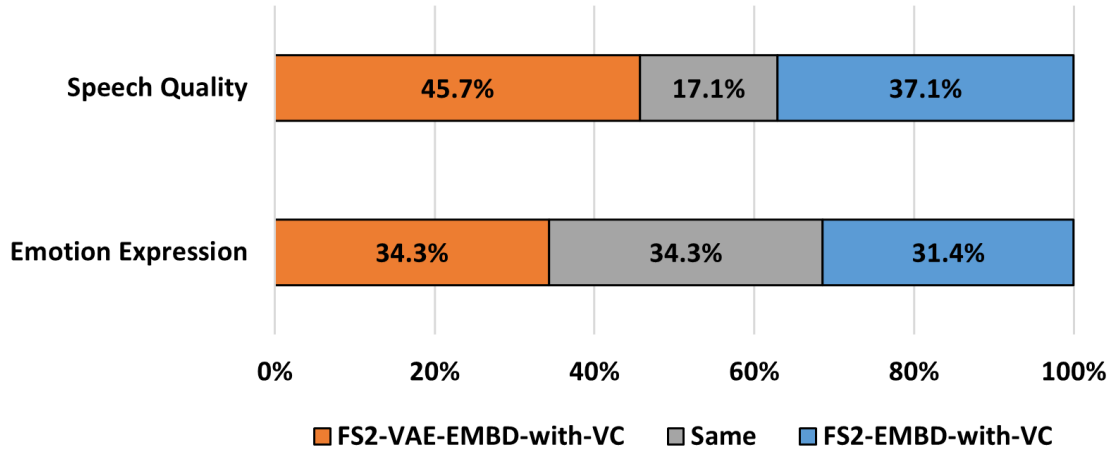


Figure 20: Results of emotion expression and speech quality (AB) preference test for the two proposed baseline models with Seq2Seq voice converter.

with selecting samples that demonstrate better emotion expression and speech quality separately. As depicted in Figure 21, the results of the test indicate the superior performance of MaskCycleGAN-VC in terms of speech quality compared to Seq2Seq-VC. However, the two models exhibit relatively similar performance in terms of emotion expression, with Seq2Seq-VC having a slight edge.

In fact, when listening to samples from the baseline model combined with Seq2Seq-VC, quality degradation can be spotted directly. Basically, the quality problems of Seq2Seq-VC are repeated or skipped words which are strongly related to autoregressive TTS models [8]. Moreover, these quality issue are also affecting the expressivity capability of the model as they prevent the listener from hearing the emotion in speech correctly and clearly. Interestingly, even though of these quality artifacts, the model is able to show comparable performance to MaskCycleGAN-VC in expressing emotions. These results actually can be indicating that Seq2Seq-VC is expected to perform better in both evaluation sides (quality and emotion expression), if the autoregressive TTS is replaced with a non-autoregressive TTS to overcome the related quality problems.

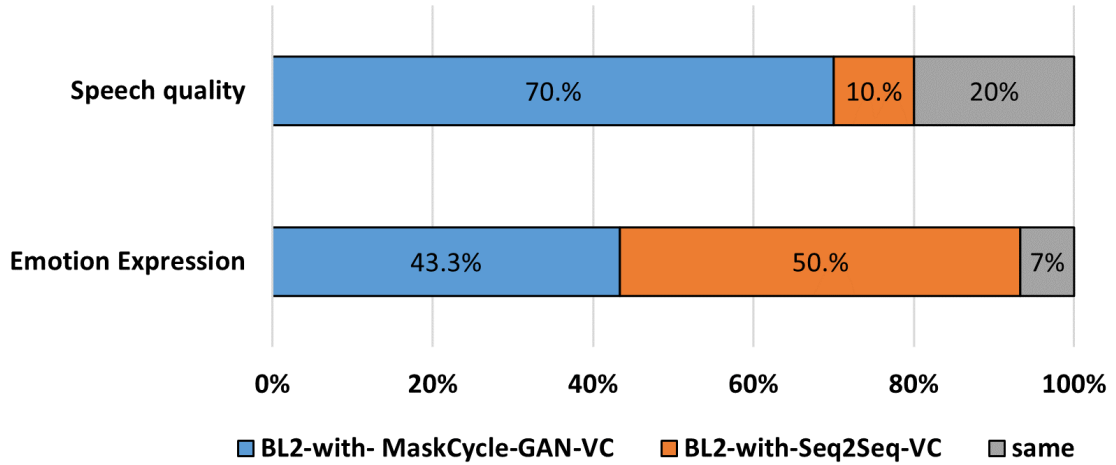


Figure 21: Results of emotion expression and speech quality preference test for second baseline model (FS2-VAE-EMBD) with MaskCycleGAN-VC and Seq2Seq-VC voice converters.

5.3 *Ensembles of Discriminators and Emotion Classifiers for Emotional Speech Synthesis*

5.3.1 Model Implementation

To implement this model, we selected our first baseline model (FS2-EMBD) as the backbone ETTS for the proposed model. We preferred to use this baseline model as it has a more simple structure which makes it more flexible to adding the proposed ensembles components in terms of implementation and training. In fact, the two baselines have similar performance, but the second baseline structure is more complicated with the VAE network and the GRL layer. Extending this baseline model with the proposed ensembles of classifiers and discriminators will probably result in a more complicated model with a complex loss function. Additionally, the complex structure of this baseline might prevent easy controlling the different added components to the model and verifying the effect of each component.

To implement the pitch extractor, we used a similar structure to the pitch predictor of FastSpeech2 with two 1D convolutional layers. The pitch extractor is initially

trained with a combination of original and synthetic mel-spectrograms that are extracted from the original ESD audios and the synthetic audios generated by the baseline model, respectively. In addition, for better pitch extraction, unvoiced pitch frames are excluded during the training process of the pitch extractor. Figure (22) illustrates pitch contour extracted from ground truth and synthetic mel-spectrograms in two training cases of the pitch extractor. As the figures show the pitch extractor performs better when it is trained on combination of original and synthetic data and with exclusion of unvoiced pitch frames.

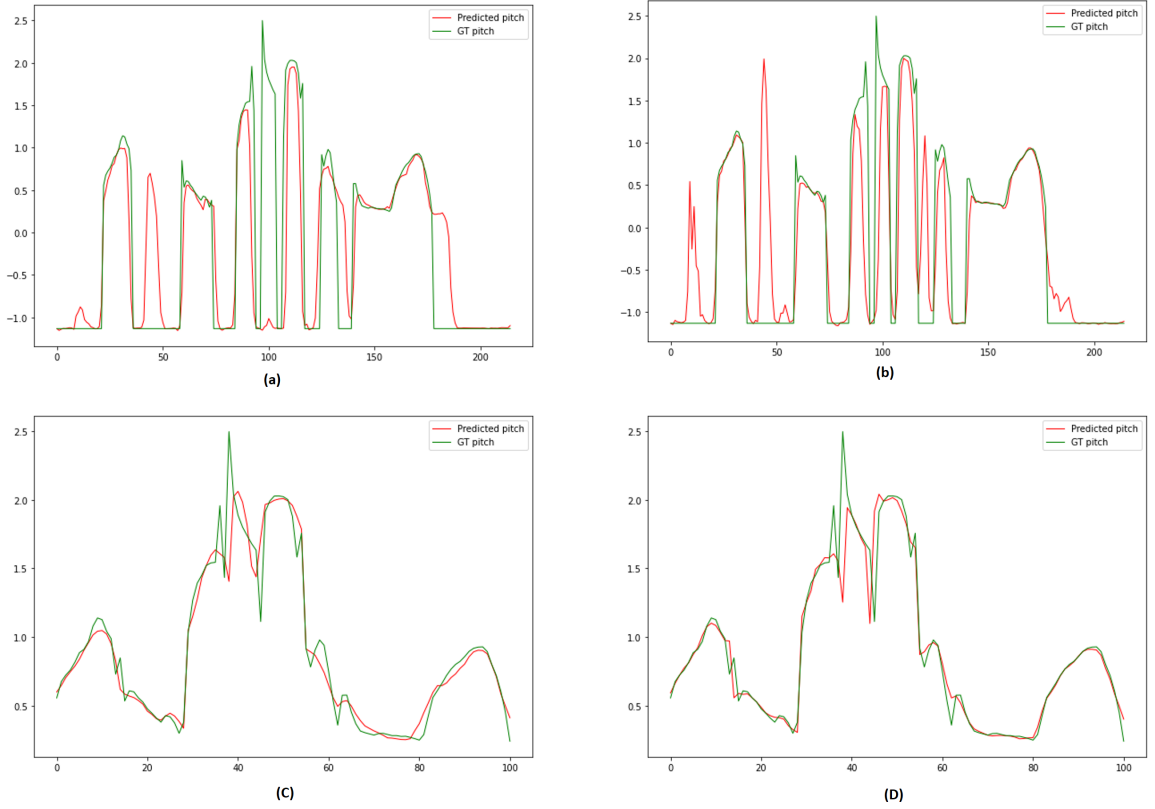


Figure 22: Pitch contour extracted by the trained pitch extractor. (a) and (b) represent pitch extracted when the extractor is trained on original data only while (c) and (d) show pitch extracted when the extractor is trained on both original and synthetic data. (a) and (c) represent pitch extracted from synthetic mel-spectrograms while (b) and (d) show pitch extracted from original mel-spectrograms.

As for emotion classifiers, we conducted several experimental training to choose the efficient architecture for each classifier. The first emotion classifier which takes

pitch sequences as input achieves best prediction accuracy with structure of three LSTM layers each of 256 units. The second emotion classifier is trained within a VAE network [144] replacing its decoder (as explained in Section 4.3). The architecture of this classifier involves three 1D convolutional layers. As for the structure and setup of VAE model, we followed the VAE implantation of this repository [150] after replacing the decoder with our second emotion classifier. Both classifiers apply log softmax as loss function.

The two classifiers are first trained individually before being trained jointly within the proposed model. Initial training of the first classifier is done on a mix of original and synthetic data from the original dataset and the baseline model, respectively. The second classifier, however, is trained on inputs from the original training dataset only which was enough to achieve high classification accuracy by this classifier even for synthetic inputs from the baseline model. Table 1) lists the classification accuracy achieved by each of the two emotion classifiers at the pre-training phase.

Table 1: Classification accuracy for the two proposed emotion classifiers at the initial training phase.

Emotion Classifier Name	Classification Accuracy
Pitch-based Classifier	64%
Emotion VAE embedding-based Classifier	87.9%

The performance of the second classifier is actually based on the emotion embeddings generated by the VAE encoder. These embeddings are probably involving distinctive emotion-related features. To evaluate the performance of the trained VAE, we applied K-means algorithm to cluster all the emotion embeddings generated by the VAE encoder for both synthetic and original mel-spectrograms of the validation dataset into five clusters. When checking emotion labels of embeddings in each of the resulting clusters, we found that each cluster involves mostly embeddings of same emotion label. This assures that the VAE encoder is able to generate distinctive

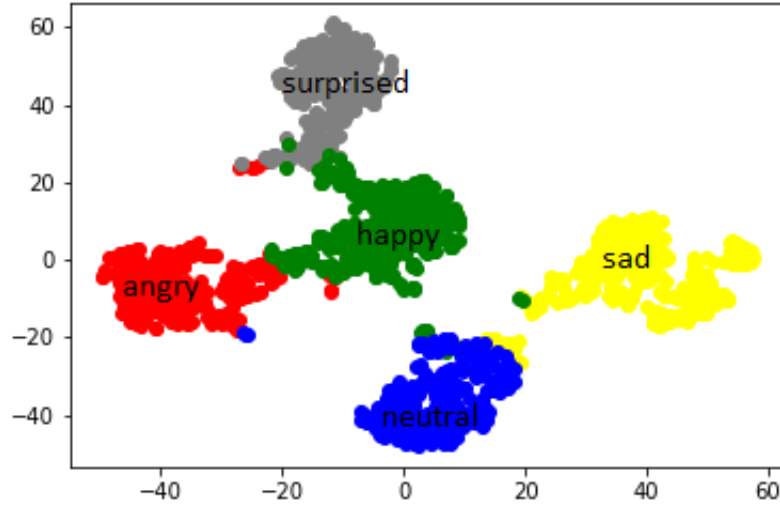


Figure 23: K-mean clusters of emotion embeddings from encoder of VAE with where embedding dimension is reduced by t-SNE algorithm.

emotion embeddings and hence support the second classifier task.

Furthermore, to visualize the resulting k-means clusters, we plotted the emotion embeddings of each cluster in $2D$ space. As the emotion embedding dimension is 32, it is first reduced into two dimensions via t-distributed stochastic neighbor embedding (t-SNE) [156]. Figure 23) illustrates how the generated clusters for the VAE emotion embeddings forms well grouped and separated clusters where each cluster includes mostly embeddings from the same emotion.

For the structure of the applied discriminators in our proposed model, we experimentally decided to choose the discriminator structure used to build the multi-scale discriminator in the HiFi vocoder [142] with least-squares loss function. The two discriminators have the same structure and differ only in the single input received by each discriminator. It is also worth to mention that input of each discriminator is combined with speaker and emotion embeddings while inputs of the emotion classifiers are combined with the speaker embeddings for better performance.

At model training stage, all the model the previous components are first initialized

from the initial training versions of each component except for the discriminators. The backbone model (FastSpeech2) is also initialized from the trained baseline model. The VAE model is also initialize with weights from its initial training process, but during the entire model training phase, it is freezed. Beside avoiding complex loss function, the VAE is freezed as it already works good on synthetic data from the baseline model although it is trained on original ESD data only. As shown in the proposed model final loss function (equation 12), we use four different hyperparameters as controlling weights that identify the participation amount of each component loss term in the total loss. Based on experiments these values (0.2, 0.2, 0.1, 1.0) are selected for the four hyperparameters in equation (12) respectively.

5.3.2 Model Objective Evaluation

Performance evaluation is one of the main challenges that faces researchers when developing ETTS models as discussed in Section (2.3.6). The different objective metrics that are used for evaluating general TTS models assess general aspect of speech quality and naturalness. However, evaluation of synthesized speech from ETTS models needs more accurate and sensitive metrics that assess style and emotion related features and hens reflect the targeted performance of the ETTS models.

on the other hand, speech expressiveness is a combination of different features that in total reflect a specific speaking style or emotion. Therefore, proposed objective metrics for ETTS models in literature which focus on only specific speech feature, such as pitch-based metrics, may not reflect the actual performance of the ETTS model as they ignore other prosody-related features. In addition, many of these metrics are based on comparison with the ground truth data, however, in the case of emotional speech, this can involve misleading information for these metrics. In other words, in emotional speech datasets, speakers differ in acting different speech styles or emotions when building emotional datasets and in some cases they could fail/mix

emotions expressions.

Accordingly, we decided, as done in several studies in literature [128, 30, 129, 74], to use a different objective evaluation metric by training a speech emotion classifier or recognition model (SER) to assess the performance of our proposed model. To the best of our knowledge, SER based on wav2vec features represent the the-state-of-the-art in this field. in Chapter (2) we provided more explanation about wav2vec features in Section (2.3.1.5).

For the purpose of evaluating our model, we fine-tuned a pre-trained wav2vec-based SER model using our dataset [151]. We used this [157] implementation from huggingface which fine-tune the SER model on a multi-emotion dataset known as Toronto emotional speech set (TESS) [158]. The classification accuracy achieved by the SER model with this dataset is (88%). After fine-tuning the pre-trained wav2vec-based SER model with the training set from our applied dataset (ESD), the model achieved classification accuracy of (97.6%) on the testing set from ESD.

To evaluate the performance of our proposed model, we first synthesized speech samples of the testing part of ESD dataset using the baseline model and our proposed model. After that, we classified the emotion of the synthesized samples from both models via the fine-tuned SER model. As shown in Table (2) the proposed model is able to improve classification accuracy by about (3%).

Table 2: Classification accuracy with wav2vec-based SER model for baseline model and the proposed model.

Samples from	Classification Accuracy
Original dataset	97.6%
Baseline model	91.4%
Proposed model	94.6%

In Figure 24, we present the confusion matrix of the SER model classification results of the synthetic waveforms from our proposed model. The confusion matrix

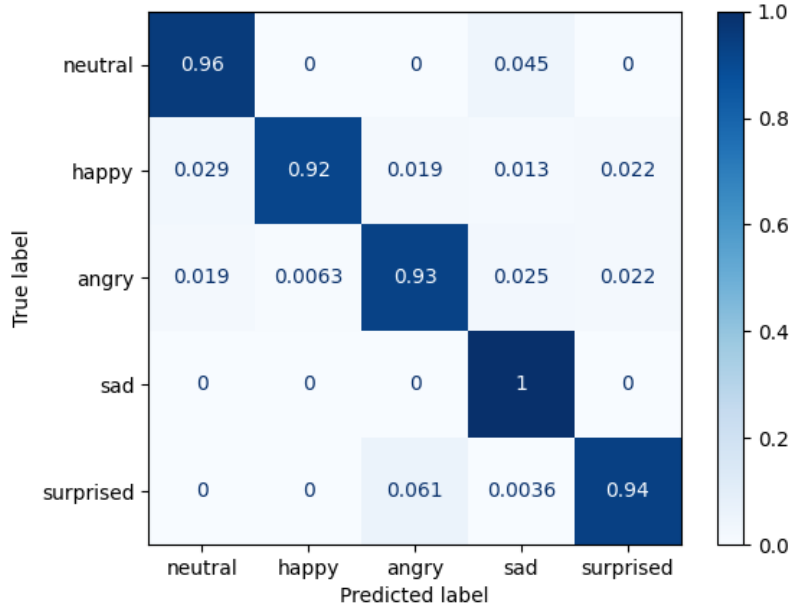


Figure 24: Confusion matrix for the Wav2Vec-based emotion classifier results on test samples from the proposed model.

indicates that the proposed model works best for emotion (sad) and less for other emotions. In fact, these results are aligned with our conclusion after listening to random samples from the ESD dataset. typically, sad emotion samples are mostly represented with low pitch and slow speaking speed by most of ESD speakers. on the other side, happy, angry and surprised samples are expressed as mix of high and low pitch and variable speaking speeds. Therefore, it is expected that the classifier task to learn features distinguishing between emotions is more easy for sad emotion compared to the other emotions.

The confusion matrix also shows that most wrong predictions for neutral emotion samples are predicted as sad emotion which is expected as the two emotions are similar. This is also aligned with the clustering results of emotion embeddings from the VAE encoder shown by Figure (23) where a small portion of neutral samples are clustered as part of sad emotion cluster. Moreover, the intermediate location of happy emotion cluster in the figure between other clusters and its overlapping with

all of them is also aligned with its predictions by the SER model.

5.3.3 Model Subjective Evaluation

We conducted two listening tests for more accurate evaluation of the proposed model in terms of emotion expression and speech quality. Both tests are preference test (AB test) between the baseline model and our proposed model. The first test evaluates expression of emotions while the second test assesses quality of speech generated by the proposed model compared to the baseline model.

In the first test, we provided listeners with parallel samples from the two models and asked them to select samples that better express the given emotions. Similarly, in second test, listeners are presented with samples from the two considered models and are asked to choose samples of better quality. 10 different English speakers participated in both tests and in total each rater listened to 60 different samples. Tests involved samples from the four different emotions for each of the ten speakers in the applied validation dataset.

During the process of selecting the test samples, we noticed several samples with different emotion labels but they express neutral emotion or wrong emotion. When referring to ESD original dataset, we figured out that the dataset has several problematic issues in terms of how speakers act the different emotions. In particular, some speakers act emotions as neutral, or mix acting of an emotion with another or even act differently for a specific emotion samples from the same speaker. On the other hand, some selected samples showed very low intelligibility and hence were difficult to be evaluated for emotion expression.

Accordingly, when selecting test samples, we first chose random samples from all emotions for each speaker from both models. We then compared the selected samples (personally) and based on our evaluation, our model performed better for emotion expression in about 47% of the samples and the baseline performed better in about

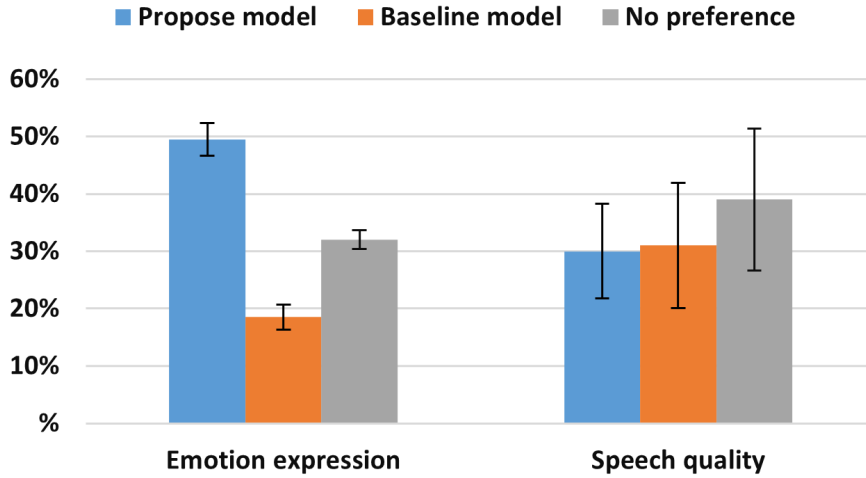


Figure 25: Listeners’ preferences for the emotion expression and speech quality between the baseline model and the proposed model. 95% confidence intervals are shown. Test involves utterances from all speakers and all emotions in the applied dataset.

20% of the samples. After that, we selected the final listening test samples for both models based on these ratios. In other words, we selected about 45% of the test samples from samples in which our model almost works better and about 20% of the test samples in which the baseline almost works better. We tried to considered only samples that are expressing emotions in a correct way and are of intelligibility that do not prevent listeners from recognizing emotions.

The Results of the two preference AB tests are summarized in Figure 25. In the first test, the listeners rated the proposed model as better in expressing emotions than the baseline in about half of the test samples as shown by Figure 25. The figure also shows considerable ratio ($\approx 32\%$) of the samples where listeners find that both models perform similar with regard to emotion expression. The baseline was rated better than the proposed model in about $\approx 19\%$ of the test samples. In general, results of evaluating emotion expression confirm the validity of our proposal regarding enhancing the ETTS model by utilizing emotion classifiers and adversarial training based on different prosody-related features.

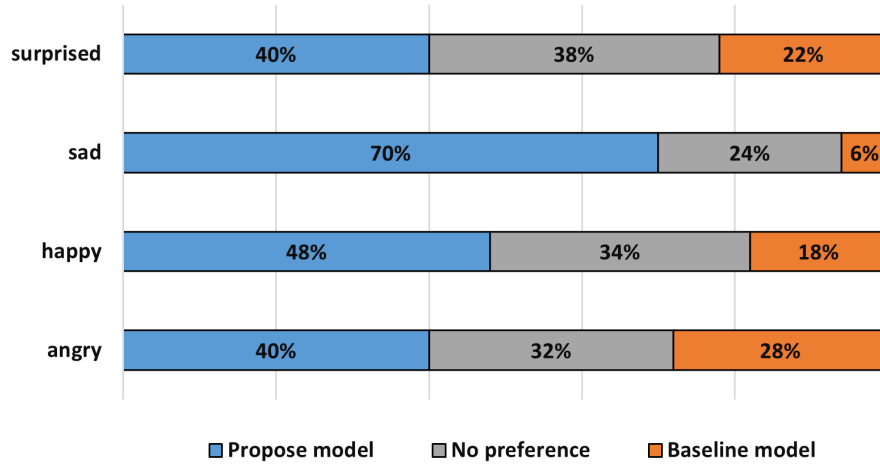


Figure 26: Listeners’ preferences for the emotion expression between the baseline model and the proposed model for each of the four emotions involved in the applied dataset.

In Figure 26 the the result of the preference test for emotion expression are shown for each emotion separately, where the proposed model works best for emotion sad and show almost similar performance for other emotions. Further details ratios for the preferred samples from the proposed model are show in Table 3. In total, sad emotion represents about 70% of the preferred samples from the proposed model in this test. These results is in line with the objective evaluation results in which emotion sad achieves the highest ratio of correct predictions made by the SER model on the test dataset synthesized by our proposed model.

Table 3: Emotions’ ratios from the preference test results of the proposed model. Ratios are calculated with regard to the total number of preferred samples by listeners from the proposed model.

Emotion type	Preference ratio
sad	35.35%
Happy	24.24%
Angry	20.20.2%
Surprised	20.20%

Furthermore, when analysing listeners’ responses, we noticed that there isn’t

agreement between the listeners in the selected samples specially for emotions happy, angry and surprised. Our personal evaluation of such samples, showed that improvement made by the proposed model in such samples is slight and thus can be missed by listeners. This issue can be attributed to the previously mentioned problems with the ESD dataset. In other words, we can say that the mistakes in acting different emotions by speakers in ESD dataset, have made the proposed model task to improve emotion expression more complicated and of low effect specially for emotions happy, angry and surprised.

As for results from the speech quality test, listeners rated most of the test samples as of same quality from both models. They also selected a ratio of about 30% of test samples as of better quality for each of the two evaluated models. These results indicate similar speech quality from both models, and hence the proposed integrated components to the baseline ETTS model are able to improve the model capability of expressing emotions while preserving the quality of generated speech.

To assess the significance of difference between the proposed model and the baseline model, paired t-test at 95% confidence interval is applied on the results of both listening tests. The results of the paired t-test showed significance performance of the proposed model in both sides of the listening test: emotion expression and speech quality.

5.3.4 Ablation Study

We conducted two ablation studies to investigate the effect of the different components that we integrated to the baseline and to check that the contribution of each of them is not redundant by another one. In the first ablation study we re-run the training process of the proposed model four times from a specific checkpoint and for specific number of iterations with one of the proposed components excluded each time. We then evaluated the model performance through the SER model when applied on

synthetic data from the re-trained models. As Table 4 shows, removing each of the four proposed components while keeping the others decreases the model performance with small ratio. However, the most degradation in the model performance results in when removing the pitch discriminator, i.e., adversarial training based on the pitch feature.

Table 4: Comparison of proposed model performance via wav2vec-based SER model when excluding each of the components in the proposed integrated ensembles to the ETTS model.

Component excluded	Classification accuracy
Emotion classifier based on pitch	94.3%
Emotion classifier based on VAE embedding	94.4%
Adversarial training based on pitch	93.8%
Adversarial training based on VAE embedding	94.2%
Model with all proposed components	94.6%

The results in Table 4 indicate that adversarial training based on the pitch feature has the highest contribution to improvement of the baseline model performance. At them time, the classifier based on pitch feature showed less effect on proposed model performance compared to the other three components. To further clarify the results of the first study, we conducted a second ablation study. In the second study, we repeated the training process of the proposed model from scratch four times considering only one of the proposed components each time while the other three components are removed. Similarly, we used the SER model to measure the performance of the four re-trained models and the results are listed in Table 5 for each of these models.

Surprisingly, results of the second ablation study showed a enhanced performance of the models integrated to each of the ensembles' components individually, except for the pitch-based classifier, compared to the first ablation study. Moreover, the model utilizing only the discriminator based on VAE emotion embedding is even able achieve a slightly higher classification accuracy over the proposed model.

Table 5: Comparison of proposed model performance via wav2vec-based SER model when including only one of the components in the proposed integrated ensembles to the ETTS model.

Component included	Classification accuracy
Emotion classifier based on pitch	94.2%
Emotion classifier based on VAE embedding	94.5%
Adversarial training based on pitch	94.4%
Adversarial training based on VAE embedding	94.8%
Model with all proposed components	94.6%

The good performance of the models with single component can be attributed to the low complexity level of these models compared to the comprehensive model. However, the difference in SER accuracy values between the ablation studies' models themselves and also compared to the comprehensive model are actually very slight. Therefore, we conducted another two preference (AB) listening tests to verify these results.

In the initial test, we contrasted samples from the proposed model with those from the model solely employing the VAE discriminator from the second ablation study. The aim was to assess whether the latter model indeed outperforms the proposed comprehensive model in terms of both emotion expression and speech quality. As depicted in the results presented in Figure 27, the proposed model exhibits superior performance in emotion expression compared to the model integrating only the VAE discriminator, with a notable degree of similarity between the two models. Regarding speech quality, both models demonstrated comparable performance. These findings suggest that the objective results of the ablation study do not accurately capture the true disparities between the compared models. In essence, these results imply that the SER model, as an objective metric, may not be sufficiently precise, particularly in discerning slight differences in emotion expression.

In the second listening test, our objective was to compare two input features (pitch

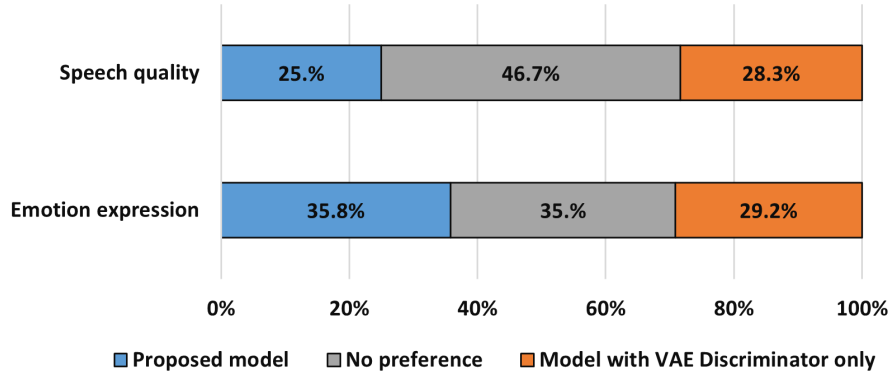


Figure 27: Listeners’ preferences for the emotion expression and speech quality between the proposed model and model utilizing the VAE discriminator only from second ablation study. Test involves utterances from all speakers and all emotions in the applied dataset.

and emotion embedding from VAE) used in the two proposed ensemble components. Since discriminators generally outperformed emotion classifiers in the objective results of two ablation studies, we decided to compare models integrating individual discriminators from the second ablation study. Similarly, this test evaluates samples from the two selected models in terms of emotion expression and speech quality. The results, summarized in Figure 28, indicate no significant difference between the two features in terms of emotion expression, with only a slight improvement in speech quality observed in the model based on emotion embedding from VAE.

In summary, the minor variations detected in SER scores during the ablation studies validate the effectiveness of each proposed component in enhancing the comprehensive model’s overall performance. Nevertheless, models utilizing these components separately do not attain the same level of performance as when they are integrated into a single model, particularly with respect to emotional expressiveness. Regarding the features employed, our preliminary tests indicated no clear superiority between the two types of input features incorporated in our proposed model.

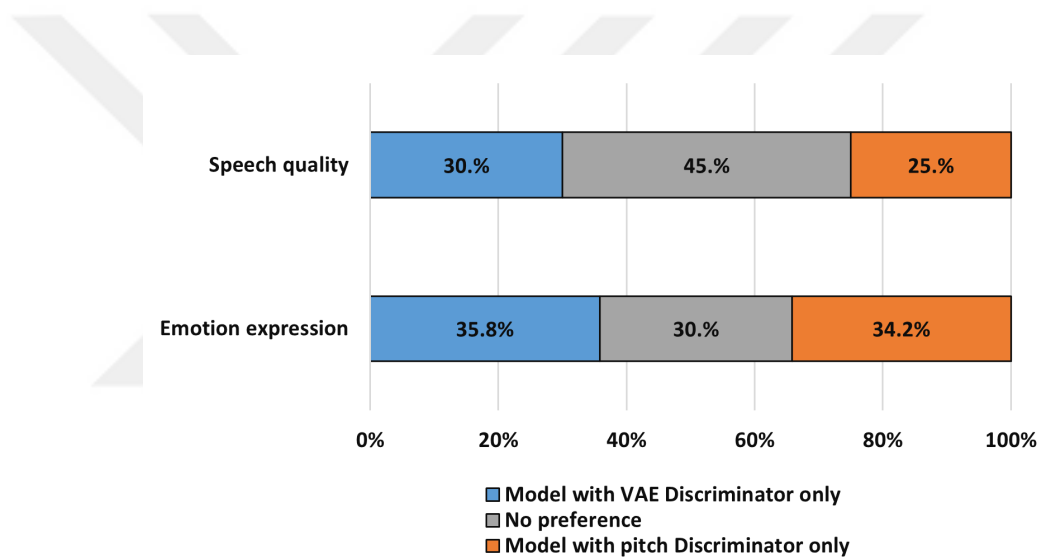


Figure 28: Listeners’ preferences for the emotion expression and speech quality between models utilizing the VAE discriminator and the pitch discriminator individually from the second ablation study. Test involves utterances from all speakers and all emotions in the applied dataset.

CHAPTER VI

CONCLUSIONS

Text-to-speech models have been through different development stages until they reached the level of producing speech that is almost human-like in terms of naturalness and intelligibility. The following stage that are considered by researchers in this field, is to generate expressive speech. In this stage, TTS models need to reflect other speech factors such as the speaker's identity, emotion, and speaking style for more human-like user interaction in the different applications of TTS models.

Generating speech in different emotions is one side of expressive speech synthesis. Emotional TTS models are mostly supervised and trained on datasets labeled with different emotions or speaking styles. However, a main challenge in this area, is the availability of these annotated datasets with multiple emotions. In fact, building a labelled emotional dataset is a challenging task as it involves many problematic issues such as wrong acting of emotions or different acting of an emotion by different speakers or even a single speaker. As a result, emotional speech datasets are not widely available and usually of small sizes which consequently complicate the development of emotional TTS models.

In this thesis, we investigated three different modules to be combined or integrated to emotional speech synthesis model for better emotion expression while preserving the limited amount of labelled training data. Two ETTS models are proposed: in the first model, we combine a voice conversion model to the ETTS model, while in the second model, two ensembles of emotion classifiers and discriminators are integrated and trained with TTS model. In both models we used same amount of labelled training data used to train the baseline TTS model.

In the first proposed model, we investigate two different emotional voice conversion models to be combined with the well-known TTS model, i.e., FastSpeech2, including: MaskCycleGAN and Seq2Seq voice conversion models. Our goal is to enhance emotion expression in synthesized speech from the baseline model via a voice converter from synthetic speech to ground truth speech of a specific emotion. Evaluation tests confirmed our proposal of emotional voice converter capability to booster emotion expression in generated speech by the TTS models. However, the quality of the produced speech is based on the efficiency of applied voice converter. While emotional Seq2Seq voice conversion models perform better in terms of emotion expression, converted speech by MaskCycleGAN model is of better quality and slightly less emotion expression capability.

In the second proposed model, the two integrated ensembles are fed with two different emotion-related features including: the pitch feature and emotion embedding from a VAE model. Objective and subjective evaluations of the model reflected improvement in expressing emotions by the proposed model while preserving speech quality from the TTS model. However, preparation and detailed analysis of evaluation tests' results revealed that problematic issues in the applied dataset has affected the results. Several emotion acting mistakes by the dataset speakers have made the proposed model task harder and consequently produced indistinctive emotional samples specially for emotion angry, happy and surprised.

Ablation studies are also conducted to evaluate the effectiveness of each component in the proposed two connected ensembles to the baseline model. In general, Ablation studies confirmed the effectiveness of each of the integrated components in enhancing the the overall performance of the TTS model. Moreover, comparison between the two applied input features showed comparable performance of the two features.

For future work, in the first proposed model, we suggest investigating of other many-to-many voice conversion models for enhancing expressiveness of generated

speech from TTS models. Many-to-many voice converters will solve the problem training a separate model for each single pair of speaker and emotion. A following step in this direction could be the possibility of training an End-to-End TTS model with a voice conversion as one comprehensive ETTS model.

As for the second proposed model, a promising future task is to train the model with a more efficient and of good quality dataset. This step will also help to perform more accurate evaluation of the proposed model. Another direction for future work is to investigate other emotion-related features such as energy and duration to be modeled for better emotion expressivity in the synthesized speech. Test-based features is another interesting type of features that can be investigated in the proposed model specially features from language models that are widely used nowadays for different AI models. Finally, we propose investigation of our model application on data from other languages such as Arabic or Turkish and identification of the required modifications in such cases.

REFERENCES

- [1] M. Le, A. Vyas, B. Shi, B. Karrer, L. Sari, R. Moritz, M. Williamson, V. Manohar, Y. Adi, J. Mahadeokar, *et al.*, “Voicebox: Text-guided multilingual universal speech generation at scale,” *arXiv preprint arXiv:2306.15687*, 2023.
- [2] T. Kaneko and H. Kameoka, “Cyclegan-vc: Non-parallel voice conversion using cycle-consistent adversarial networks,” in *2018 26th European Signal Processing Conference (EUSIPCO)*, pp. 2100–2104, IEEE, 2018.
- [3] T. Kaneko, H. Kameoka, K. Tanaka, and N. Hojo, “Maskcyclegan-vc: Learning non-parallel voice conversion with filling in frames,” in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5919–5923, IEEE, 2021.
- [4] K. Zhou, B. Sisman, and H. Li, “Limited Data Emotional Voice Conversion Leveraging Text-to-Speech: Two-Stage Sequence-to-Sequence Training,” in *Proc. Interspeech 2021*, pp. 811–815, 2021.
- [5] H. Ze, A. Senior, and M. Schuster, “Statistical parametric speech synthesis using deep neural networks,” in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 7962–7966, IEEE, 2013.
- [6] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, *et al.*, “Tacotron: Towards end-to-end speech synthesis,” *arXiv preprint arXiv:1703.10135*, 2017.
- [7] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerry-Ryan, *et al.*, “Natural tts synthesis by conditioning wavenet on mel spectrogram predictions,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4779–4783, IEEE, 2018.
- [8] Y. Ren, Y. Ruan, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, “Fastspeech: Fast, robust and controllable text to speech,” *Advances in neural information processing systems*, vol. 32, pp. 3171–3180, 2019.
- [9] Y. Ren, C. Hu, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, “Fast-speech 2: Fast and high-quality end-to-end text to speech,” *arXiv preprint arXiv:2006.04558*, 2020.
- [10] R. Skerry-Ryan, E. Battenberg, Y. Xiao, Y. Wang, D. Stanton, J. Shor, R. Weiss, R. Clark, and R. A. Saurous, “Towards end-to-end prosody transfer for expressive speech synthesis with tacotron,” in *international conference on machine learning*, pp. 4693–4702, PMLR, 2018.

- [11] Y. Wang, D. Stanton, Y. Zhang, R.-S. Ryan, E. Battenberg, J. Shor, Y. Xiao, Y. Jia, F. Ren, and R. A. Saurous, “Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis,” in *International Conference on Machine Learning*, pp. 5180–5189, PMLR, 2018.
- [12] K. Akuzawa, Y. Iwasawa, and Y. Matsuo, “Expressive speech synthesis via modeling expressions with variational autoencoder,” *arXiv preprint arXiv:1804.02135*, 2018.
- [13] Y.-J. Zhang, S. Pan, L. He, and Z.-H. Ling, “Learning latent representations for style control and transfer in end-to-end speech synthesis,” in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6945–6949, IEEE, 2019.
- [14] S. Suzić, T. Nosek, M. Sečujski, D. Pekar, and V. Delić, “Dnn based expressive text-to-speech with limited training data,” in *2019 27th Telecommunications Forum (TELFOR)*, pp. 1–6, IEEE, 2019.
- [15] Y. Hono, K. Tsuboi, K. Sawada, K. Hashimoto, K. Oura, Y. Nankaku, and K. Tokuda, “Hierarchical multi-grained generative model for expressive speech synthesis,” *arXiv preprint arXiv:2009.08474*, 2020.
- [16] T. Cornille, F. Wang, and J. Bekker, “Interactive multi-level prosody control for expressive speech synthesis,” in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8312–8316, IEEE, 2022.
- [17] M. Kim, S. J. Cheon, B. J. Choi, J. J. Kim, and N. S. Kim, “Expressive text-to-speech using style tag,” *arXiv preprint arXiv:2104.00436*, 2021.
- [18] S. Suzic, T. V. Delic, S. Ostrogonac, S. Duric, and D. J. Pekar, “Style-code method for multi-style parametric text-to-speech synthesis,” *SPIIRAS Proceedings*, vol. 5, no. 60, p. 216, 2018.
- [19] K. Inoue, S. Hara, M. Abe, N. Hojo, and Y. Ijima, “An investigation to transplant emotional expressions in dnn-based tts synthesis,” in *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pp. 1253–1258, IEEE, 2017.
- [20] L. Xue, X. Zhu, X. An, and L. Xie, “A comparison of expressive speech synthesis approaches based on neural network,” in *Proceedings of the Joint Workshop of the 4th Workshop on Affective Social Multimedia Computing and first Multi-Modal Affective Computing of Large-Scale Multimedia Data (ASMMC-MMAC’18)*, pp. 15–20, ACM, 2018.
- [21] S. Suzić, T. Delić, D. Pekar, V. Delić, and M. Sečujski, “Style transplantation in neural network based speech synthesis,” *Acta Polytechnica Hungarica*, vol. 16, no. 6, pp. 171–189, 2019.

- [22] H. Choi, S. Park, J. Park, and M. Hahn, “Multi-speaker emotional acoustic modeling for cnn-based speech synthesis,” in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6950–6954, IEEE, 2019.
- [23] M. Secujski, D. Pekar, S. Suzic, A. Smirnov, and T. V. Nosek, “Speaker/style-dependent neural network speech synthesis based on speaker/style embedding,” *J. Univers. Comput. Sci.*, vol. 26, no. 4, pp. 434–453, 2020.
- [24] K. Inoue, S. Hara, M. Abe, N. Hojo, and Y. Ijima, “Model architectures to extrapolate emotional expressions in dnn-based text-to-speech,” *Speech Communication*, vol. 126, pp. 35–43, 2021.
- [25] R. Li, Z. Wu, Y. Huang, J. Jia, H. Meng, and L. Cai, “Emphatic speech generation with conditioned input layer and bidirectional lstms for expressive speech synthesis,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5129–5133, IEEE, 2018.
- [26] J. Lorenzo-Trueba, G. E. Henter, S. Takaki, J. Yamagishi, Y. Morino, and Y. Ochiai, “Investigating different representations for modeling and controlling multiple emotions in dnn-based speech synthesis,” *Speech Communication*, vol. 99, pp. 135–143, 2018.
- [27] X. Wu, L. Sun, S. Kang, S. Liu, Z. Wu, X. Liu, and H. Meng, “Feature based adaptation for speaking style synthesis,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5304–5308, IEEE, 2018.
- [28] J. Parker, Y. Stylianou, and R. Cipolla, “Adaptation of an expressive single speaker deep neural network speech synthesis system,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5309–5313, IEEE, 2018.
- [29] P. Wu, Z. Ling, L. Liu, Y. Jiang, H. Wu, and L. Dai, “End-to-end emotional speech synthesis using style tokens and semi-supervised training,” in *2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pp. 623–627, IEEE, 2019.
- [30] H.-W. Yoon, O. Kwon, H. Lee, R. Yamamoto, E. Song, J.-M. Kim, and M.-J. Hwang, “Language Model-Based Emotion Prediction Methods for Emotional Speech Synthesis Systems,” in *Proc. Interspeech 2022*, pp. 4596–4600, 2022.
- [31] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Nee-lakantan, P. Shyam, G. Sastry, A. Askell, *et al.*, “Language models are few-shot learners,” *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

- [32] D. Parikh and K. Grauman, “Relative attributes,” in *2011 International Conference on Computer Vision*, pp. 503–510, IEEE, 2011.
- [33] T. Li, S. Yang, L. Xue, and L. Xie, “Controllable emotion transfer for end-to-end speech synthesis,” in *2021 12th International Symposium on Chinese Spoken Language Processing (ISCSLP)*, pp. 1–5, IEEE, 2021.
- [34] K. He, C. Sun, R. Zhu, and L. Zhao, “Multi-speaker emotional speech synthesis with limited datasets: Two-stage non-parallel training strategy,” in *2022 7th International Conference on Intelligent Computing and Signal Processing (ICSP)*, pp. 545–548, IEEE, 2022.
- [35] T. Li, X. Wang, Q. Xie, Z. Wang, and L. Xie, “Cross-speaker emotion disentangling and transfer for end-to-end speech synthesis,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 1448–1460, 2022.
- [36] T. Li, X. Wang, Q. Xie, Z. Wang, M. Jiang, and L. Xie, “Cross-speaker emotion transfer based on prosody compensation for end-to-end speech synthesis,” *arXiv preprint arXiv:2207.01198*, 2022.
- [37] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, “Domain-adversarial training of neural networks,” *The journal of machine learning research*, vol. 17, no. 1, pp. 2096–2030, 2016.
- [38] Y. Gao, W. Zheng, Z. Yang, T. Kohler, C. Fuegen, and Q. He, “Interactive text-to-speech system via joint style analysis,” *arXiv preprint arXiv:2002.06758*, 2020.
- [39] F. Eyben, F. Weninger, F. Gross, and B. Schuller, “Recent developments in opensmile, the munich open-source multimedia feature extractor,” in *Proceedings of the 21st ACM international conference on Multimedia*, pp. 835–838, 2013.
- [40] Y. Lei, S. Yang, and L. Xie, “Fine-grained emotion strength transfer, control and prediction for emotional speech synthesis,” in *2021 IEEE Spoken Language Technology Workshop (SLT)*, pp. 423–430, IEEE, 2021.
- [41] C. Kim, S. Y. Um, H. Yoon, and H. G. Kang, “Fluenttts: Text-dependent fine-grained style control for multi-style tts,” in *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, vol. 2022, pp. 4561–4565, 2022.
- [42] M. Morise, F. Yokomori, and K. Ozawa, “World: a vocoder-based high-quality speech synthesis system for real-time applications,” *IEICE TRANSACTIONS on Information and Systems*, vol. 99, no. 7, pp. 1877–1884, 2016.

- [43] L. Xue, F. K. Soong, S. Zhang, and L. Xie, “Paratts: Learning linguistic and prosodic cross-sentence information in paragraph-based tts,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 2854–2864, 2022.
- [44] J. Zaïdi, H. Seuté, B. van Niekerk, and M.-A. Carbonneau, “Daft-exprt: Cross-speaker prosody transfer on any text for expressive speech synthesis,” *arXiv preprint arXiv:2108.02271*, 2021.
- [45] E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville, “Film: Visual reasoning with a general conditioning layer,” in *Proceedings of the AAAI conference on artificial intelligence*, 2018.
- [46] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, “wav2vec 2.0: A framework for self-supervised learning of speech representations,” *Advances in neural information processing systems*, vol. 33, pp. 12449–12460, 2020.
- [47] W. Zhao and Z. Yang, “An emotion speech synthesis method based on vits,” *Applied Sciences*, vol. 13, no. 4, p. 2225, 2023.
- [48] J. Kim, J. Kong, and J. Son, “Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech,” in *International Conference on Machine Learning*, pp. 5530–5540, PMLR, 2021.
- [49] S. Moon, S. Kim, and Y.-H. Choi, “Mist-tacotron: end-to-end emotional speech synthesis using mel-spectrogram image style transfer,” *IEEE Access*, vol. 10, pp. 25455–25463, 2022.
- [50] L. A. Gatys, A. S. Ecker, and M. Bethge, “Image style transfer using convolutional neural networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2414–2423, 2016.
- [51] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.
- [52] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” *arXiv preprint arXiv:1312.6114*, 2013.
- [53] Y. Taigman, L. Wolf, A. Polyak, and E. Nachmani, “Voiceloop: Voice fitting and synthesis via a phonological loop,” *arXiv preprint arXiv:1707.06588*, 2017.
- [54] T. Kenter, V. Wan, C.-A. Chan, R. Clark, and J. Vit, “Chive: Varying prosody in speech synthesis with a linguistically driven dynamic hierarchical conditional variational network,” in *International Conference on Machine Learning*, pp. 3331–3340, PMLR, 2019.
- [55] T. Kenter, M. K. Sharma, and R. Clark, “Improving prosody of rnn-based english text-to-speech synthesis by incorporating a bert model,” in *Proc. Interspeech 2020*, p. 4412–4416, 2020.

- [56] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” *arXiv preprint arXiv:1810.04805*, 2018.
- [57] Y. Wang, Y. Xie, K. Zhao, H. Wang, and Q. Zhang, “Unsupervised quantized prosody representation for controllable speech synthesis,” in *2022 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6, IEEE, 2022.
- [58] H.-S. Oh, S.-H. Lee, and S.-W. Lee, “Diffprosody: Diffusion-based latent prosody generation for expressive speech synthesis with prosody conditional adversarial training,” *arXiv preprint arXiv:2307.16549*, 2023.
- [59] A. Van Den Oord, O. Vinyals, *et al.*, “Neural discrete representation learning,” *Advances in neural information processing systems*, vol. 30, pp. 6306–6315, 2017.
- [60] Z. Xiao, K. Kreis, and A. Vahdat, “Tackling the generative learning trilemma with denoising diffusion gans,” *arXiv preprint arXiv:2112.07804*, 2021.
- [61] N.-Q. Wu, Z.-C. Liu, and Z.-H. Ling, “Discourse-level prosody modeling with a variational autoencoder for non-autoregressive expressive speech synthesis,” in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7592–7596, IEEE, 2022.
- [62] Y.-J. Zhang and Z.-H. Ling, “Extracting and predicting word-level style variations for speech synthesis,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 1582–1593, 2021.
- [63] X. An, Y. Wang, S. Yang, Z. Ma, and L. Xie, “Learning hierarchical representations for expressive speaking style in end-to-end speech synthesis,” in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 184–191, IEEE, 2019.
- [64] S. Lei, Y. Zhou, L. Chen, Z. Wu, S. Kang, and H. Meng, “Towards expressive speaking style modelling with hierarchical context information for mandarin speech synthesis,” in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7922–7926, IEEE, 2022.
- [65] R. Liu, B. Sisman, and H. Li, “Reinforcement learning for emotional text-to-speech synthesis with improved emotion discriminability,” *arXiv preprint arXiv:2104.01408*, 2021.
- [66] R. Valle, J. Li, R. Prenger, and B. Catanzaro, “Mellotron: Multispeaker expressive voice synthesis by conditioning on rhythm, pitch and global style tokens,” in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6189–6193, IEEE, 2020.

- [67] C. Wang, S. Chen, Y. Wu, Z. Zhang, L. Zhou, S. Liu, Z. Chen, Y. Liu, H. Wang, J. Li, *et al.*, “Neural codec language models are zero-shot text to speech synthesizers,” *arXiv preprint arXiv:2301.02111*, 2023.
- [68] K. Shen, Z. Ju, X. Tan, Y. Liu, Y. Leng, L. He, T. Qin, S. Zhao, and J. Bian, “Naturalspeech 2: Latent diffusion models are natural and zero-shot speech and singing synthesizers,” *arXiv preprint arXiv:2304.09116*, 2023.
- [69] A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, “High fidelity neural audio compression,” *arXiv preprint arXiv:2210.13438*, 2022.
- [70] X. Wu, Y. Cao, M. Wang, S. Liu, S. Kang, Z. Wu, X. Liu, D. Su, D. Yu, and H. Meng, “Rapid style adaptation using residual error embedding for expressive speech synthesis,” in *Interspeech*, pp. 3072–3076, 2018.
- [71] T. Raitio, J. Li, and S. Seshadri, “Hierarchical prosody modeling and control in non-autoregressive parallel neural tts,” in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7587–7591, IEEE, 2022.
- [72] F. Yang, S. Yang, Q. Wu, Y. Wang, and L. Xie, “Exploiting deep sentential context for expressive end-to-end speech synthesis,” *arXiv preprint arXiv:2008.00613*, 2020.
- [73] C. Lu, X. Wen, R. Liu, and X. Chen, “Multi-speaker emotional speech synthesis with fine-grained prosody modeling,” in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5729–5733, IEEE, 2021.
- [74] S. Jo, Y. Lee, Y. Shin, Y. Hwang, and T. Kim, “Cross-speaker emotion transfer by manipulating speech style latents,” in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, IEEE, 2023.
- [75] K. Lee, K. Park, and D. Kim, “Styler: Style factor modeling with rapidity and robustness via speech decomposition for expressive and controllable neural text to speech,” *arXiv preprint arXiv:2103.09474*, 2021.
- [76] D. Tan and T. Lee, “Fine-grained style modeling, transfer and prediction in text-to-speech synthesis via phone-level content-style disentanglement,” *arXiv preprint arXiv:2011.03943*, 2020.
- [77] K. Zhang, C. Gong, W. Lu, L. Wang, J. Wei, and D. Liu, “Joint and adversarial training with asr for expressive speech synthesis,” in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6322–6326, IEEE, 2022.

- [78] S. Karlapati, A. Moinet, A. Joly, V. Klimkov, D. Sáez-Trigueros, and T. Drugman, “Copycat: Many-to-many fine-grained prosody transfer for neural text-to-speech,” *arXiv preprint arXiv:2004.14617*, 2020.
- [79] S. Liang, C. Miao, M. Chen, J. Ma, S. Wang, and J. Xiao, “Unsupervised learning for multi-style speech synthesis with limited data,” in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6583–6587, IEEE, 2021.
- [80] S. Pan and L. He, “Cross-speaker style transfer with prosody bottleneck in neural speech synthesis,” *arXiv preprint arXiv:2107.12562*, 2021.
- [81] X. Dai, C. Gong, L. Wang, and K. Zhang, “Information sieve: Content leakage reduction in end-to-end prosody for expressive speech synthesis,” *arXiv preprint arXiv:2108.01831*, 2021.
- [82] K. Qian, Y. Zhang, S. Chang, X. Yang, and M. Hasegawa-Johnson, “Autovc: Zero-shot voice style transfer with only autoencoder loss,” in *International Conference on Machine Learning*, pp. 5210–5219, PMLR, 2019.
- [83] A. A. Alemi, I. Fischer, J. V. Dillon, and K. Murphy, “Deep variational information bottleneck,” *arXiv preprint arXiv:1612.00410*, 2016.
- [84] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *International conference on machine learning*, pp. 448–456, pmlr, 2015.
- [85] D. Ulyanov, A. Vedaldi, and V. Lempitsky, “Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6924–6932, 2017.
- [86] Z. Lyu and J. Zhu, “Enriching style transfer in multi-scale control based personalized end-to-end speech synthesis,” in *2022 12th International Conference on Information Science and Technology (ICIST)*, pp. 114–119, IEEE, 2022.
- [87] T.-Y. Hu, A. Shrivastava, O. Tuzel, and C. Dhir, “Unsupervised style and content separation by minimizing mutual information for speech synthesis,” in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3267–3271, IEEE, 2020.
- [88] M. I. Belghazi, A. Baratin, S. Rajeshwar, S. Ozair, Y. Bengio, A. Courville, and D. Hjelm, “Mutual information neural estimation,” in *International conference on machine learning*, pp. 531–540, PMLR, 2018.
- [89] D. Yang, S. Liu, R. Huang, G. Lei, C. Weng, H. Meng, and D. Yu, “Instructtts: Modelling expressive tts in discrete latent space with natural language style prompt,” *arXiv preprint arXiv:2301.13662*, 2023.

- [90] P. Cheng, W. Hao, S. Dai, J. Liu, Z. Gan, and L. Carin, “Club: A contrastive log-ratio upper bound of mutual information,” in *International conference on machine learning*, pp. 1779–1788, PMLR, 2020.
- [91] D. Paul, S. Mukherjee, Y. Pantazis, and Y. Stylianou, “A universal multi-speaker multi-style text-to-speech via disentangled representation learning based on rényi divergence minimization,” in *Interspeech*, pp. 3625–3629, 2021.
- [92] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, “Hubert: Self-supervised speech representation learning by masked prediction of hidden units,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3451–3460, 2021.
- [93] L.-W. Chen and A. Rudnicky, “Fine-grained style control in transformer-based text-to-speech synthesis,” in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7907–7911, IEEE, 2022.
- [94] S. Lei, Y. Zhou, L. Chen, Z. Wu, S. Kang, and H. Meng, “Towards expressive speaking style modelling with hierarchical context information for mandarin speech synthesis,” in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7922–7926, IEEE, 2022.
- [95] Y. Guo, C. Du, and K. Yu, “Unsupervised word-level prosody tagging for controllable speech synthesis,” in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7597–7601, IEEE, 2022.
- [96] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, “Xlnet: Generalized autoregressive pretraining for language understanding,” *Advances in neural information processing systems*, vol. 32, pp. 5753–5763, 2019.
- [97] Y. Lei, S. Yang, X. Wang, and L. Xie, “Msemotts: Multi-scale emotion transfer, prediction, and control for emotional speech synthesis,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 853–864, 2022.
- [98] Z. Zeng, J. Wang, N. Cheng, and J. Xiao, “Prosody learning mechanism for speech synthesis system without text length limit,” *arXiv preprint arXiv:2008.05656*, 2020.
- [99] C. Du and K. Yu, “Rich prosody diversity modelling with phone-level mixture density network,” *arXiv preprint arXiv:2102.00851*, 2021.
- [100] F. Richter, *Mixture Density Networks*, pp. 171–198. Wiesbaden: Deutscher Universitätsverlag, 2003.
- [101] S. Karlapati, A. Abbas, Z. Hodari, A. Moinet, A. Joly, P. Karanasou, and T. Drugman, “Prosodic representation learning and contextual sampling for

- neural text-to-speech,” in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6573–6577, IEEE, 2021.
- [102] S. Tyagi, M. Nicolis, J. Rohnke, T. Drugman, and J. Lorenzo-Trueba, “Dynamic prosody generation for speech synthesis using linguistics-driven acoustic embedding selection,” *arXiv preprint arXiv:1912.00955*, 2019.
 - [103] Y. Shen, Z. Lin, C.-W. Huang, and A. Courville, “Neural language modeling by jointly learning syntax and lexicon,” *arXiv preprint arXiv:1711.02013*, 2017.
 - [104] I. Vallés-Pérez, J. Roth, G. Beringer, R. Barra-Chicote, and J. Droppo, “Improving multi-speaker tts prosody variance with a residual encoder and normalizing flows,” *arXiv preprint arXiv:2106.05762*, 2021.
 - [105] D. Stanton, Y. Wang, and R. Skerry-Ryan, “Predicting expressive speaking style from text in end-to-end speech synthesis,” in *2018 IEEE Spoken Language Technology Workshop (SLT)*, pp. 595–602, IEEE, 2018.
 - [106] L. Zhao, J. Yang, and Q. Qin, “Enhancing prosodic features by adopting pre-trained language model in bahasa indonesia speech synthesis,” in *2020 3rd International Conference on Algorithms, Computing and Artificial Intelligence*, pp. 1–6, 2020.
 - [107] R. Liu, B. Sisman, G. Gao, and H. Li, “Expressive tts training with frame and style reconstruction loss,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 1806–1818, 2021.
 - [108] S. Lei, Y. Zhou, L. Chen, Z. Wu, S. Kang, and H. Meng, “Context-aware coherent speaking style prediction with hierarchical transformers for audiobook speech synthesis,” in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, IEEE, 2023.
 - [109] N. Prateek, M. Łajszczak, R. Barra-Chicote, T. Drugman, J. Lorenzo-Trueba, T. Merritt, S. Ronanki, and T. Wood, “In other news: A bi-style text-to-speech model for synthesizing newscaster voice with limited data,” *arXiv preprint arXiv:1904.02790*, 2019.
 - [110] J. Sarzynska-Wawer, A. Wawer, A. Pawlak, J. Szymanowska, I. Stefaniak, M. Jarkiewicz, and L. Okruszek, “Detecting formal thought disorder by deep contextualized word representations,” *Psychiatry Research*, vol. 304, p. 114135, 2021.
 - [111] G. Xu, W. Song, Z. Zhang, C. Zhang, X. He, and B. Zhou, “Improving prosody modelling with cross-utterance bert embeddings for end-to-end speech synthesis,” in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6079–6083, IEEE, 2021.

- [112] A. Sun, J. Wang, N. Cheng, H. Peng, Z. Zeng, L. Kong, and J. Xiao, “Graphpb: Graphical representations of prosody boundary in speech synthesis,” in *2021 IEEE Spoken Language Technology Workshop (SLT)*, pp. 438–445, IEEE, 2021.
- [113] Y. Wu, X. Wang, S. Zhang, L. He, R. Song, and J.-Y. Nie, “Self-supervised context-aware style representation for expressive speech synthesis,” *arXiv preprint arXiv:2206.12559*, 2022.
- [114] Y. Zhou, C. Song, J. Li, Z. Wu, Y. Bian, D. Su, and H. Meng, “Enhancing word-level semantic representation via dependency structure for expressive text-to-speech synthesis,” *arXiv preprint arXiv:2104.06835*, 2021.
- [115] I. Jauk, J. Lorenzo Trueba, J. Yamagishi, and A. Bonafonte C  vez, “Expressive speech synthesis using sentiment embeddings,” in *Interspeech 2018: 2-6 September 2018, Hyderabad*, pp. 3062–3066, International Speech Communication Association (ISCA), 2018.
- [116] T. Raitio, R. Rasipuram, and D. Castellani, “Controllable neural text-to-speech synthesis using intuitive prosodic features,” *arXiv preprint arXiv:2009.06775*, 2020.
- [117] C. Gong, L. Wang, Z. Ling, S. Guo, J. Zhang, and J. Dang, “Improving naturalness and controllability of sequence-to-sequence speech synthesis by learning local prosody representations,” in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5724–5728, IEEE, 2021.
- [118] D. S. R. Mohan, V. Hu, T. H. Teh, A. Torresquintero, C. G. Wallis, M. Staib, L. Foglianti, J. Gao, and S. King, “Ctrl-p: Temporal control of prosodic variation for speech synthesis,” *arXiv preprint arXiv:2106.08352*, 2021.
- [119] G. Pamisetty and K. Sri Rama Murty, “Prosody-tts: An end-to-end speech synthesis system with prosody control,” *Circuits, Systems, and Signal Processing*, vol. 42, no. 1, pp. 361–384, 2023.
- [120] T. H. Teh, V. Hu, D. S. R. Mohan, Z. Hodari, C. G. Wallis, T. G. Ibarrondo, A. Torresquintero, J. Leoni, M. Gales, and S. King, “Ensemble prosody prediction for expressive speech synthesis,” in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, IEEE, 2023.
- [121] Y. Zou, S. Liu, X. Yin, H. Lin, C. Wang, H. Zhang, and Z. Ma, “Fine-Grained Prosody Modeling in Neural Speech Synthesis Using ToBI Representation,” in *Proc. Interspeech 2021*, pp. 3146–3150, 2021.
- [122] Y. Lee and T. Kim, “Robust and fine-grained prosody control of end-to-end speech synthesis,” in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5911–5915, IEEE, 2019.

- [123] W. Nakata, T. Koriyama, S. Takamichi, Y. Saito, Y. Ijima, R. Masumura, and H. Saruwatari, “Predicting VQVAE-based Character Acting Style from Quotation-Annotated Text for Audiobook Speech Synthesis,” in *Proc. Interspeech 2022*, pp. 4551–4555, 2022.
- [124] M. Morrison, Z. Jin, J. Salamon, N. J. Bryan, and G. J. Mysore, “Controllable neural prosody synthesis,” *arXiv preprint arXiv:2008.03388*, 2020.
- [125] G. Zhang, Y. Qin, and T. Lee, “Learning syllable-level discrete prosodic representation for expressive speech generation,” in *Interspeech*, pp. 3426–3430, 2020.
- [126] C.-B. Im, S.-H. Lee, S.-B. Kim, and S.-W. Lee, “Emoq-tts: Emotion intensity quantization for fine-grained controllable emotional text-to-speech,” in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6317–6321, IEEE, 2022.
- [127] X. Zhu, S. Yang, G. Yang, and L. Xie, “Controlling emotion strength with relative attribute for end-to-end speech synthesis,” in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 192–199, IEEE, 2019.
- [128] O. Kwon, I. Jang, C. Ahn, and H.-G. Kang, “An effective style token weight control technique for end-to-end emotional speech synthesis,” *IEEE Signal Processing Letters*, vol. 26, no. 9, pp. 1383–1387, 2019.
- [129] S.-Y. Um, S. Oh, K. Byun, I. Jang, C. Ahn, and H.-G. Kang, “Emotional speech synthesis with rich and granularized control,” in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7254–7258, IEEE, 2020.
- [130] A. Vioni, M. Christidou, N. Ellinas, G. Vamvoukakis, P. Kakoulidis, T. Kim, J. S. Sung, H. Park, A. Chalamandaris, and P. Tsiakoulis, “Prosodic clustering for phoneme-level prosody control in end-to-end speech synthesis,” in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5719–5723, IEEE, 2021.
- [131] N. Ellinas, M. Christidou, A. Vioni, J. S. Sung, A. Chalamandaris, P. Tsiakoulis, and P. Mastorocostas, “Controllable speech synthesis by learning discrete phoneme-level prosodic representations,” *Speech Communication*, vol. 146, pp. 22–31, 2023.
- [132] X. Cai, D. Dai, Z. Wu, X. Li, J. Li, and H. Meng, “Emotion controllable speech synthesis using emotion-unlabeled dataset with the assistance of cross-domain speech emotion recognition,” in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5734–5738, IEEE, 2021.

- [133] L. Xue, S. Pan, L. He, L. Xie, and F. K. Soong, “Cycle consistent network for end-to-end style transfer tts training,” *Neural Networks*, vol. 140, pp. 223–236, 2021.
- [134] R. Li, D. Pu, M. Huang, and B. Huang, “Unet-tts: Improving unseen speaker and style transfer in one-shot voice cloning,” in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8327–8331, IEEE, 2022.
- [135] cstr, “The blizzard challenge.” <https://www.cstr.ed.ac.uk/projects/blizzard/>. [Accessed 15-Sep-2023].
- [136] K. Ito and L. Johnson, “The lj speech dataset.” <https://keithito.com/LJ-Speech-Dataset/>, 2017.
- [137] cstr, “Voice cloning toolkit.” <https://datashare.ed.ac.uk/handle/10283/3443>. [Accessed 15-Sep-2023].
- [138] H. Zen, R. Clark, R. J. Weiss, V. Dang, Y. Jia, Y. Wu, Y. Zhang, and Z. Chen, “Libritts: A corpus derived from librispeech for text-to-speech,” in *Interspeech*, pp. 1526–1530, 2019.
- [139] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” *Advances in neural information processing systems*, vol. 27, pp. 2672–2680, 2014.
- [140] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired image-to-image translation using cycle-consistent adversarial networks,” in *Proceedings of the IEEE international conference on computer vision*, pp. 2223–2232, 2017.
- [141] T. Kaneko, H. Kameoka, K. Tanaka, and N. Hojo, “Cyclegan-vc2: Improved cyclegan-based non-parallel voice conversion,” in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6820–6824, IEEE, 2019.
- [142] J. Kong, J. Kim, and J. Bae, “Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 17022–17033, 2020.
- [143] R. Yamamoto, E. Song, and J.-M. Kim, “Parallel wavegan: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram,” in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6199–6203, IEEE, 2020.
- [144] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” *arXiv preprint arXiv:1312.6114*, 2013.

- [145] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” *Advances in neural information processing systems*, vol. 30, pp. 5998–6008, 2017.
- [146] M. McAuliffe, M. Socolof, S. Mihuc, M. Wagner, and M. Sonderegger, “Montreal forced aligner: Trainable text-speech alignment using kaldi,” in *Interspeech*, vol. 2017, pp. 498–502, 2017.
- [147] K. Jungil, K. Jaehyeon, and B. Jaekyoung, “hifi-gan.” <https://github.com/jik876/hifi-gan>. [Accessed 15-Apr-2024].
- [148] T. Van Erven and P. Harremos, “Rényi divergence and kullback-leibler divergence,” *IEEE Transactions on Information Theory*, vol. 60, no. 7, pp. 3797–3820, 2014.
- [149] C.-M. Chien, J.-H. Lin, C.-y. Huang, P.-c. Hsu, and H.-y. Lee, “FastSpeech2.” <https://github.com/ming024/FastSpeech2>. [Accessed 15-Apr-2024].
- [150] C. Jinhan, “tacotron2-vae.” <https://github.com/jinhan/tacotron2-vae>. [Accessed 15-Apr-2024].
- [151] K. Zhou, B. Sisman, R. Liu, and H. Li, “Emotional voice conversion: Theory, databases and esd,” *Speech Communication*, vol. 137, pp. 1–18, 2022.
- [152] H. Hikaru, D. Terbe, and H. Karl, “Maskcyclegan-vc.” <https://github.com/GANTastic3/MaskCycleGAN-VC>. [Accessed 15-Apr-2024].
- [153] T. Kaneko, H. Kameoka, K. Tanaka, and N. Hojo, “Stargan-vc2: Rethinking conditional methods for stargan-based voice conversion,” *arXiv preprint arXiv:1907.12279*, 2019.
- [154] H. Kameoka, T. Kaneko, K. Tanaka, and N. Hojo, “Stargan-vc: Non-parallel many-to-many voice conversion using star generative adversarial networks,” in *2018 IEEE Spoken Language Technology Workshop (SLT)*, pp. 266–273, IEEE, 2018.
- [155] Z. Kun, S. Berrak, and H. Li, “seq2seq-evc.” <https://github.com/KunZhou9646/seq2seq-EVC?tab=readme-ov-file>. [Accessed 15-Apr-2024].
- [156] L. Van der Maaten and G. Hinton, “Visualizing data using t-sne,” *Journal of machine learning research*, vol. 9, no. 11, pp. 2579–2605, 2008.
- [157] D. Brian, “wav2vec2-base-toronto-emotional-speech-set.” https://huggingface.co/DunnBC22/wav2vec2-base-Toronto_emotional_speech_set. [Accessed 15-Apr-2024].
- [158] L. Eu Jin, “Toronto emotional speech set (tess).” <https://www.kaggle.com/datasets/ejlok1/toronto-emotional-speech-set-tess>. [Accessed 15-Apr-2024].

VITA

Huda Barakat received her B.S. degree in computer science from the University of Science and Technology, Sana'a, Yemen, in 2001 and her M.Sc. degree in Computer Science from the University of Malaya, Kuala Lumpur, Malaysia, in 2012. She has been working as a system analyst and programmer in different companies in Yemen since her graduation. Since 2009, she has been working as a senior IT specialist in the IT department at the secretariat of the cabinet of the capital, Yemen. Her research interests include information retrieval, natural language processing, and digital speech processing.