FEW-SHOT VOICE AND FOREIGN ACCENT CONVERSION AND ITS APPLICATIONS IN PRONUNCIATION TRAINING

A Dissertation

by

SHAOJIN DING

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Ricardo Gutierrez-Osuna
Committee Members,	Ruihong Huang
	Theodora Chaspari
	Tie Liu
Head of Department,	Scott Schaefer

May 2021

Major Subject: Computer Science

ABSTRACT

Voice Conversion (VC) aims to transform the speech of a source speaker to sound as if a target speaker had produced it. As a closely related but more challenging research problem, foreign accent conversion (FAC) [1] aims to create a new voice that has the voice identity of a given non-native (L2) speaker and the accent of a native (L1) speaker. Prior VC and FAC approaches require a considerable amount of speech data from each target speaker for model training, which can be tedious to collect and demotivating for users when applying to real-world scenarios. This dissertation aims to address these problems by introducing three few-shot learning approaches for VC and FAC:

- Few-shot VC based on sparse representation
- Zero-shot VC based on sequence-to-sequence (seq2seq) model
- Zero-shot FAC based on seq2seq model

In the first approach, I develop a novel sparse representation for VC that requires as much as one minute of speech from each target speaker. The proposed approach consists of two complementary components: a Cluster-Structured Dictionary Learning module to learn a dictionary capturing the speakers' characteristics and a Cluster-Selective Objective Function to compute the sparse representation carrying linguistic content. The approach outperforms previous methods that use Gaussian Mixture Model (GMM) and sparse representation, improving both acoustic quality and voice identity of the VC syntheses.

In the second approach, I create a seq2seq model for zero-shot VC that reduces the amount of target speech required from minutes to seconds. The model transforms a linguistic content representation (e.g., from phonetic posteriorgrams) to Mel-spectrogram, conditioned on the target speaker embedding. Moreover, I propose an adversarial training scheme that reduces the speaker-dependent cues from the phonetic posteriorgram. The approach can synthesize more natural speech than conventional methods based on GMM and sparse representations. I also show that the adversarial training scheme further improves the voice identity of the synthesized speech.

In the third approach, I generalize the zero-shot VC model for use in FAC. Compared to zero-shot VC, this approach has an additional accent encoder that generates an accent embedding, which is consumed by the seq2seq model. As in the second approach, the seq2seq model transforms the linguistic content representation to Mel-spectrogram, conditioned on the desired speaker embedding, but now also on the accent embedding. I show via perceptual studies that the proposed approach reduces the accentedness of the syntheses compared to a state-of-the-art seq2seq based FAC approach, while retaining the acoustic quality and voice identity. More importantly, it reduces the required speech from each L2 speaker from hours to seconds.

During the time of conducting this research, I also acted with a leading role in developing Golden Speaker Builder, a web application that uses FAC algorithms for pronunciation training. I will describe the design and implementation of this web application and the user experience feedback collected from the users who participated in the user studies.

DEDICATION

To my parents and fiancé

ACKNOWLEDGEMENTS

First of all, I would like to sincerely appreciate my advisor Dr. Ricardo Gutierrez-Osuna, an insightful and brilliant scholar. It is my great honor and pleasure to work with you during the five years, and thanks for your guidance and valuable feedback that pushed my work to a higher level. I would not have done this dissertation without your support. I also would like to thank for the help and feedback from my committee members Dr. Theodora Chaspari, Dr. Ruihong Huang, and Dr. Tie Liu.

I appreciate the chance to work with my fantastic collaborators at Iowa State University, Dr. John Levis, Dr. Evgeny Chukharev-Hudilainen, Dr. Sinem Sonsaat, Alif Silpachai, Ivana Lučić. Thanks for your previous advice and your devotion to GSB development and studies.

To my colleagues, Avinash, Genna, Roger, María, Adam, Chris, Dennis, Guanlong, Zelun, Tianlong, Anurag, Sudip, Nitin, and Mu, I cherish the time to work with you talented individuals, and thank you for your help and encouragement in the past five years. To my friends, Jiaqi Liu, Di Lu, Shifan Li, I really missed the time when we played basketball as a team in REC and wish you all the best in your future endeavors after graduation.

Finally, I would like to thank my parents, Mr. Feng Ding and Mrs. Shuangling Sun, and my fiancé Shuang Li for your unconditional love and accompaniment. You helped me through a good and dark time, and you earn this degree right along with me. I will try my best to take care of all of you in the future.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

The dissertation committee members are Dr. Ricardo Gutierrez-Osuna (chair), Dr. Theodora Chaspari, Dr. Ruihong Huang, and Dr. Tie Liu. Other people who contributed to this dissertation research include Dr. Guanlong Zhao, Christopher Liberatore, Dr. Sinem Sonsaat, Alif Silpachai, Ivana Lučić Rehman, Dr. Evgeny Chukharev-Hudilainen, and Dr. John Levis.

Funding Sources

This dissertation was supported by two National Science Foundation research grants (1619212 and 1623750).

TABLE OF CONTENTS

Page

LIST OF FIGURES

Page

LIST OF TABLES

1. INTRODUCTION

Voice Conversion (VC) aims to transform the speech of a source speaker to sound as if a target speaker had produced it. A typical VC system first decouples speech utterances into two representations: (1) their linguistic content and (2) the speaker's voice identity, and then combines the linguistic content from source speech with the voice identity of a target speaker to produce the VC speech. Foreign accent conversion (FAC) is closely related to VC but more challenging – it creates a new voice that has the voice identity of a given non-native (L2) speaker and the accent of a native (L1) speaker. In contrast with VC, FAC has to decouple the speech utterances into three representations: (1) their linguistic content, (2) the speaker's voice identity, and (3) their accent. The synthesized FAC speech incorporates the linguistic content and accent from an L1 speaker along with the voice identity of an L2 speaker.

VC and FAC find use in a number of applications, such as personalized text-to-speech synthesis [1], speaking assistance [2], speech enhancements [3], and especially pronunciation training [4]. In pronunciation training, FAC can create a "golden speaker" for the L2 speaker: their own voice, but with a native accent [4-7]. The "golden speaker" well-matches the voice characteristics of the L2 speaker, which has been shown to be more effective for L2 speakers to practice with than a poor-matched voice [8, 9].

Various approaches have been proposed to perform VC and FAC. In terms of VC, methods based on Gaussian Mixture Model (GMM) [10, 11] and Deep Neural Networks (DNN) [12-20] are commonly used and can achieve good performance. For FAC, a variety of techniques have been proposed, including voice morphing [4, 21, 22], frame pairing [23, 24], articulatory synthesis [25, 26], and sequence-to-sequence (seq2seq) modeling [27, 28]. However, previous VC and FAC approaches require a considerable amount of speech data (from 100s to 1,000s utterances) from each target speaker for model training. Thus, when using these conventional methods in real-world applications such as pronunciation training, L2 learners need to record a large number of utterances and then wait for model training before using the system. This can be tedious and demotivating for learners and reduces the efficiency of the pronunciation-training process.

This dissertation aims to address the data requirements of previous VC and FAC methods. In the first part of this dissertation (Chapters 3, 4, and 5), I propose three different state-of-the-art VC and FAC models that only use a few speech utterances from the target speaker during training (i.e., few-shot learning [29]). In the second part of this dissertation (Chapter 6), I bridge the gap between FAC techniques and the need for pronunciation training – I developed an interactive web application, Golden Speaker Builder[1], which enables L2 speakers to build their "golden speaker" voice using FAC models and practice with it online.

In the first work (Chapter 3), I propose a few-shot VC model based on a novel sparse representation . Sparse-representation based VC models [30-32] require much smaller training corpora (~20 utterances from the target speaker) [31] and are more robust to noisy speech than GMMs [30]. A typical sparse-representation based VC model consists of a dictionary construction step and a sparse coding. These models assume that the sparse representation carries linguistic content, and the dictionary captures the speakers' characteristics. However, conventional dictionary-construction

[1] https://goldenspeaker.engl.iastate.edu

and sparse-coding algorithms rarely meet this assumption. The result is that the sparse code is no longer speaker-independent, which leads to lower voice-conversion performance. To address the problem, I propose a Cluster-Structured Sparse Representation (CSSR) algorithm that improves the speaker-independence of the representations, and thus, the acoustic quality and voice identity of VC syntheses. CSSR consists of two complementary components: a Cluster-Structured Dictionary Learning module that groups atoms in the dictionary into clusters, and a Cluster-Selective Objective Function that encourages each speech frame to be represented by atoms from a small number of clusters.

In the second work (Chapter 4), I propose a zero-shot VC approach based on a seq2seq model, which only requires one utterance from the target speaker during inference. The proposed seq2seq model (which I term PPG2speech synthesizer) has an encoder-decoder structure, which transforms a sequence of Phonetic-Posteriorgram (PPG) to a sequence of speech features (e.g., Mel-spectrogram), conditioned on the corresponding speaker embedding (e.g., i-vector [33], d-vector [34]). In practice, however, I noticed that PPGs still carry speaker identity information such as accent, intonation, and speaking rate [17] that can leak into the voice conversions, which can degrade the voice identity of VC syntheses. To resolve this problem, I propose a new training procedure that includes an adversarial speaker classifier jointly trained with the PPG2speech synthesizer, improving the speaker independence of the hidden representation and the voice identity of VC syntheses. During inference, the model can be directly applied to generate voice conversions to arbitrary target speaker given a few seconds of audio (i.e., the amount of audio needed to compute a speaker embedding/fingerprint), without the need to have any model re-training or adaptation process.

In the third work (Chapter 5), I propose a zero-shot FAC approach based on the seq2seq model. As described earlier, FAC characterizes speech utterances using three representations: their linguistic content, the speakers' voice identity, and their accent. To capture the three aspects of an utterance, I use three independent models: (1) a speaker-independent acoustic model to extract a linguistic content representation sequence (denoted as a bottleneck feature vector), (2) a speaker encoder to generate a speaker embedding, and (3) an accent encoder to obtain an accent embedding. Then, I train a novel seq2seq model to synthesize speech using the linguistic content representation and accent embedding from an L1 speaker along with the speaker embedding of an L2 speaker. Once the model is trained, it can generate accent conversions to arbitrary L2 speakers given a few seconds of audio that is used to extract their speaker embeddings, as described in the second work.

The fourth and last part of this dissertation (Chapter 6) focuses on filling the gap between FAC models and the need for pronunciation training. With the assistance of my collaborators Christopher Liberatore, Dr. Guanlong Zhao, Dr. John Levis, Dr. Evgeny Chukharev-Hudilainen, I developed Golden Speaker Builder, a web application for L2 speakers to synthesize their "golden speaker" voice using FAC models and practice with it. To the best of my knowledge, this is the first interface that applies FAC technique to the use of pronunciation training. This tool can beneficially promote future research progress on computer-assisted pronunciation training and provide a more efficient way for L2 learners to improve their pronunciation.

In summary, this dissertation research consists of four main objectives:

1) **Few-shot VC based on sparse representation:** Develop a few-shot voice conversion system using structured dictionary learning and sparse coding algorithms.

2) **Zero-shot VC based on seq2seq model:** Develop a zero-shot voice conversion system using state-of-the-art seq2seq model and adversarial learning.

3) **Zero-shot FAC based on seq2seq model:** Develop a zero-shot foreign accent conversion system using state-of-the-art seq2seq model.

4) **Golden speaker builder:** Develop a web application that applies foreign accent conversion systems to pronunciation training.

The research in this dissertation has the following major contributions.

- Objective 1) improves the acoustic quality and voice identity of VC syntheses, and it only needs minutes of speech from the target speaker during model training.
- Objective 2) improves the voice identity of VC syntheses. More importantly, it eliminates the need for using utterances from the target speaker during training.
- Objective 3) extends the idea in objective 2) to FAC, and it significantly improves the foreign accentedness ratings of the syntheses compared to the previous FAC approaches.
- Objective 4) provides a valuable resource for future research on computer-assisted pronunciation training and provide a more efficient way for L2 learner to improve their pronunciation.

All the works in this dissertation were submitted to or published in top-tier peer-reviewed venues. Initial findings from Objective 1) were published at *Interspeech 2018*; an improved method was published by the *IEEE/ACM Transactions on Audio, Speech, and Language Processing (TASLP)*. Findings from Objective 2) were published at *Interspeech 2020*. Results from Objective 3) were submitted to *TASLP*.

Finally, the description of the web application developed in Objective 4) was published by *Speech Communication*, and the web application is available at https://goldenspeaker.engl.iastate.edu/.

The rest of this proposal is organized as follows. Chapter 2 reviews the related background for this dissertation. Chapter 3, 4, and 5 describe the detailed work for each objective. Chapter 7 summarizes this dissertation work and points out potential future directions.

2. BACKGROUND

2.1. Voice conversion

Most conventional VC systems, such as those based on GMMs, DNNs, and sparse representations, require time-aligned parallel corpora. GMM-based methods [10, 11] learn the joint distribution of source and target spectral features and then estimate the target spectral features through least-squares regression. DNN-based methods map the source spectral features directly into the target space through various network structures such as restricted Boltzmann machines [12], auto-encoders [35], feed-forward neural networks [13], and recurrent neural networks [36]. Sparse representation methods [30, 31] first build exemplar dictionaries for a source and a target speaker. At runtime, they use sparse coding to extract a speaker-independent code from the source speech and then combine it with the target dictionary to generate VC speech.

To avoid the laborious process of collecting parallel corpora, several non-parallel VC techniques have been proposed in recent years. These include the INCA algorithm [37], DNNs [14, 15], sparse representations [38], and phonetic posteriorgrams [16]. More recently, several studies have proposed many-to-many VC approaches based on Variational Autoencoders (VAE) [39-44] and the PPG-to-speech synthesizer [45-49]. Hsu *et al.* [39, 50] first proposed to use a VAE for many-to-many VC. Their VAE consists of an encoder and a decoder. During training, the encoder learns a speaker-independent latent embedding from input speech signals, and the decoder reconstructs the input speech signals given the latent embedding and the corresponding speaker embedding. During inference, they replace the speaker embedding with that of a target speaker to produce VC syntheses. A number of subsequent studies have been conducted to improve performance through various techniques, such as using auxiliary classifiers [43], WaveNet vocoder adaption [44], and using discrete latent space [40, 51]. Other studies [45-47] have used a PPG-to-speech synthesizer approach to perform many-to-many VC. The PPG-to-speech synthesizer is a neural network that takes PPGs as an input, and predict spectra conditioned on the speaker embedding of the target speaker. Early many-to-many VC models used one-hot vectors as the speaker embedding due to its simplicity, but recent studies [41, 45-47] have used learned speaker embeddings (e.g., i-vector [33], d-vector [34]) to generalize to unseen speakers, which make it possible to perform the so-called zero-shot VC.

2.2. Foreign accent conversion

Early approaches to FAC [26, 52-54] usually involved building an articulatory synthesizer for an L2 speaker. The articulatory synthesizer was trained to map the speaker's articulatory trajectories (e.g., tongue and lip movements) into his or her acoustics features (e.g., Mel Cepstra) using GMMs [26], unit-selection models [52], and DNNs [53]. Once the synthesizer was built, it could be driven with articulatory trajectories from an L1 speaker to synthesize FAC speech. However, these approaches are impractical since collecting articulatory data is expensive and requires specialized equipment[2]. As an alternative, *acoustic* methods are more practical since they only require recording speech with a microphone. Previous acoustic methods can be grouped into two categories: frame-pairing methods [23, 24] and seq2seq methods [27, 28]. Frame-pairing methods first pair L1 and L2 speech frames based on their similarity, and

[2] Articulatory measurements can be performed via electromagnetic articulography [52], ultrasound imaging [55], palatography [56], and more recently real-time MRI [57]

then use a statistical model (e.g., a GMM) to convert from L1 frames to their corresponding L2 frames. Aryal and Gutierrez-Osuna [23] first proposed a technique to pair L1-L2 frames based on their acoustic similarity (in MFCC space), after applying vocal tract length normalization to reduce global differences between the L1 and L2 spectra. Following this, Zhao *et al.* [24] argued that the L1 and L2 frames should be paired based on their linguistic content, and consequently, they used Phonetic-PosteriorGram (PPG) similarity instead of MFCC similarity to pair acoustic frames. More recently, methods based on seq2seq models have been shown to significantly improve synthesis quality. In a previous study [27], Zhao *et al.* proposed a seq2seq PPG-to-Mel synthesizer for FAC. During training, the system learns a seq2seq model to convert PPGs to Mel-spectra extracted from utterances of an L2 speaker. During inference, the model is driven by PPGs extracted from a reference L1 utterance, which then produces FAC synthesis. In related work, Liu *et al.* [28] proposed a novel recognizer-synthesizer framework to remove the need for a reference L1 utterance. Their system trained a speaker recognizer, a multi-speaker TTS model, and an accent-sensitive ASR system. During inference, they feed L2 Mel-spectra to the ASR system with the corresponding accent, and then feed the output of the ASR system and the L2 speaker embedding to the multi-speaker TTS model to generate accent-converted utterances. These seq2seq model based FAC approaches can convert segmental and prosody features simultaneously, producing syntheses with higher speech naturalness and acoustic quality.

2.3. Few-shot and zero-shot learning

Few-shot learning [29] is the machine learning paradigm that trains the model using only limited samples with supervised information from a specific domain. Zero-shot learning [58], as an extreme case of few-shot learning, does not require any sample with supervised information from a specific domain during training. Previous few-shot learning studies solve the few-shot learning problems from three perspectives: data, model, and parameter searching, as discussed next.

In terms of data, augmentation techniques are used to increase the number of training samples. These techniques use prior knowledge to augment the samples from the specific domain, including manual augmentations within the training set (e.g., flipping, cropping, and rotation for images [59-61]; time warping, frequency masking, and time masking for speech spectrograms [62]) and augmentations using unlabeled data (e.g., using exemplar classifier to predict labels for unlabeled data [63]).

From the perspective of model, prior studies explore the use of knowledge sharing/transferring and designed different models to incorporate the shared knowledge. These include multi-task learning [61, 64, 65] and embedding learning [66-68]. Multi-task learning paradigm learns multiple tasks at the same time, which utilizes both task-generic and task-specific information. With multi-task learning, one can train a model that jointly learns the target task (which has limited samples) along with a few auxiliary tasks (which have sufficient samples). Embedding learning aims to learn a transformation function that can project samples to a lower-dimensional embedding vector. In this embedding space, samples that are similar regarding an attribute will be closer to each other, and vice versa (e.g., the speaker recognition model projects spectral features into an embedding space, where similar speakers are closer to each other). In few-shot learning, embedding learning can transfer the prior knowledge of the attribute to new samples, which helps the learning of the target task.

Lastly, in terms of parameter searching, the most commonly used techniques are model finetuning and adaptation [69, 70]. In computer vision, ImageNet pre-trained models [71] are often finetuned for different tasks (e.g., person re-identification, object detection, and image segmentation). Similarly, in natural language processing, pre-trained Bidirectional Encoder Representations from Transformers (BERT) are used to derive models for different tasks (e.g., language modeling, sentiment analysis, question answering). These model finetuning strategies can not only avoid the need of a large amount of data in model training but also significantly improve their performances.

In this dissertation, the models proposed in the first three objectives are with the few-shot learning paradigm. The sparse representation proposed in Objective 1 is a few-shot learning model since the sparse coding algorithm itself only requires minimal data. The speaker recognition models in Objective 2 and Objective 3 are in line with the embedding learning perspective of few-shot learning. Namely, I use the speaker embedding extracted from a separately trained to control the speaker identity of the seq2seq models.

2.4. Sparse coding and dictionary learning

Sparse coding algorithms [72] aim to learn a useful sparse representation of input data in an unsupervised fashion. Generally, they represent input feature vectors using a linear combination of a few atoms in a dictionary, which captures high-level patterns of input data. Formally, suppose we have input data vector $\mathbf{X} \in \mathbb{R}^D$ and a set of pre-selected dictionary $\mathbf{A} \in \mathbb{R}^{D\times N}$, we can represent $\mathbf{X}$ as:

$$\mathbf{X} \cong \mathbf{AW} \tag{2.1}$$

where $\mathbf{W} \in \mathbb{R}^N$ is a sparse non-negative weight vector (i.e., a sparse representation), N is the number of atoms in the dictionary, and D is the dimensionality of the vector. Given $\mathbf{X}$ and $\mathbf{A}$, $\mathbf{W}$ can be approximated via solving a standard sparse coding objective function (i.e., Lasso):

$$\mathbf{W} = \underset{\mathbf{W}}{\operatorname{argmin}}\, d(\mathbf{X}, \mathbf{AW}) + \alpha\|\mathbf{W}\|_1, \qquad s.t.\ \mathbf{W} \geq 0 \tag{2}$$

where $d(\cdot)$ is a distance metric, typically Euclidean distance. The L_1 norm term is included to enforce sparsity in $\mathbf{W}$, with α being a sparsity penalty.

Early sparse coding algorithms usually use a set of pre-defined atoms (e.g., samples from the signal) [73]. However, learning the dictionary rather than directly using these pre-defined bases has been shown to be more effective for signal reconstruction [74-76]. Accordingly, a number of algorithms have been proposed to learn these dictionaries, such as the Alternating Minimization fashioned algorithm [77], Proximal methods [78], and online dictionary learning algorithms [79]. Sparse coding and dictionary learning algorithms have proven to be successful in various computer vision and speech processing tasks such as face recognition [79, 80], image classification [77, 81], speech enhancement [82, 83], speech recognition [84], and source separation [85]. This model is used for Objective 1 in this research.

2.5. Seq2seq models

Seq2seq models aim to transform an input sequence into an output sequence that may have a different length, which was originally proposed by Sutskever et al. [86] for machine translation. The seq2seq model usually has an encoder-decoder architecture, as shown in Figure 2.1. The encoder learns a hidden representation sequence from an input sequence, and the decoder learns to autoregressively generate the output sequence given the hidden representation. To capture local contextual information and handle length mismatches between the input and output sequences, an attention mechanism is added between the encoder and the decoder.

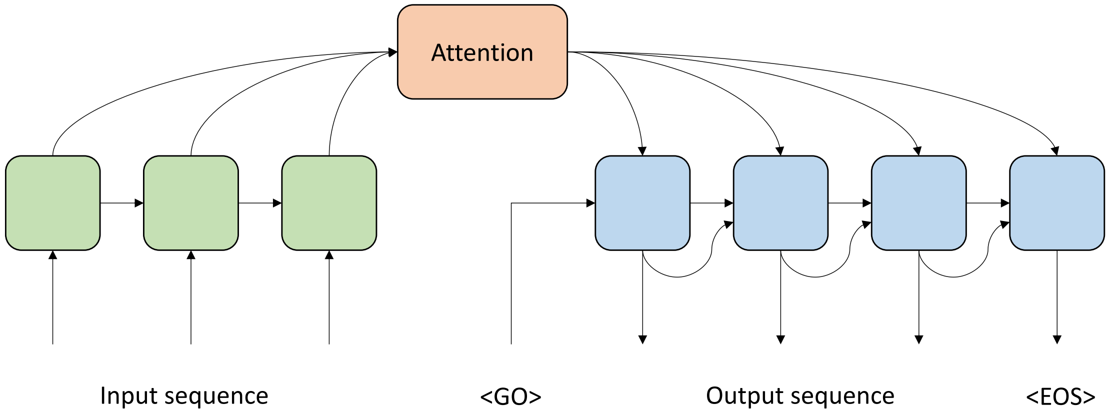


Figure 2.1: Illustration of seq2seq model. Green module indicates the encoder, and Blue module indicates the decoder. <GO> represents an initialized starting frame, and <EOS> represents the end of sentence frame.

In recent years, there has been growing interest in applying seq2seq model to speech synthesis. Wang et al. [87] first proposed a seq2seq based TTS synthesizer (Tacotron), which significantly improved the acoustic quality of the syntheses over previous methods. Following this, Shen et al. [88] proposed Tacotron2, which further improved the acoustic quality of Tacotron by using a novel model architecture and a WaveNet vocoder. Jia et al. [89] extended Tacotron2 to multi-speaker TTS by conditioning a speaker embedding on the decoder. Seq2seq model has also been applied to voice conversion [17, 48, 49] and foreign accent conversion [27, 28], which significantly improved the performance on these tasks compared to conventional approaches. This type of model is used for Objective 2 and Objective 3 in this dissertation.

2.6. Speaker recognition

Speaker recognition systems aim to automatically determine the identity of a speaker from the speech signal. There are two common tasks in speaker recognition: speaker identification and speaker verification. Speaker identification aims to determine who the speaker is and to which group s/he belongs. In contrast, speaker verification aims to verify if the voice of a speaker matches the speaker's claimed identity. A speaker recognition system consists of two main components: embedding learning and metric learning. Much of the work has focused on learning better speaker embeddings. Earlier systems used spectral features (e.g., MFCCs) as the speaker embedding [90, 91], but these have been superseded by systems based on identity vectors, or "i-vectors" for short [92]. An i-vector takes speech from a speaker (e.g., MFCCs) and uses it to adapt a speaker-independent Gaussian Mixture Model (GMM), referred to as a Universal Background Model (UBM). The means of the adapted GMM are then concatenated to form a supervector, which is then reduced in dimensionality using joint factor analysis. More recently systems use "d-vectors" as the speaker embedding [93]. The d-vector is computed as the final hidden layer of a Deep Neural Network (DNN) trained to classify speakers from frame-level acoustic features. Research efforts have also been devoted to finding better metrics to identify or verify speakers. Most of conventional speaker recognition systems used cosine distance between speaker embeddings as the similarity metric, but more advanced techniques such as Probabilistic Linear Discriminant Analysis (PLDA) [94] and its variants, heavy-tailed PLDA [95] and Gauss-PLDA [96] have also been used. In this dissertation work, I used the speaker embedding produced by speaker recognition models to represent speaker identity in Objective 2 and Objective 3.

2.7. Language and accent recognition

A complementary problem to speaker recognition is language recognition [97]. In language recognition, the goal is to automatically determine the language being used in a given speech segment from an unknown speaker. Language recognition techniques can be roughly grouped into two main categories: token-based (a.k.a., phonotactic) and spectral-based. In the token-based approach, a bank of phone recognizers is used to convert each speech utterance into a string of discrete units/tokens. These tokens are then used to classify the underlying language/dialect. In the spectral-based approach, a spoken utterance is represented as a sequence of cepstral feature vectors. Most modern language recognition systems are based on i-vectors extracted from these feature vectors. The standard approach first uses an UBM to extract i-vectors from the given speech segment. Once extracted, i-vectors are commonly classified using cosine scoring, Gaussian backend, neural network, or logistic regression [98].

The task of accent/dialect recognition [99, 100] is relatively unexplored compared to language recognition. This is in part due to the lack of common benchmark datasets, but also that accent/dialect recognition is often treated as a special case of language recognition, so researchers tend to concentrate on the more general problem of language recognition. However, accent recognition is a more difficult problem because the dialects that belong to the same language family are very similar, and, in the case of non-native accents, there exists large variability from speaker to speaker. Accent/dialect recognition is useful for a variety of problems. For example, in ASR it would enable the recognizer to adapt its pronunciation, acoustic, and language models appropriately [101, 102]. Dialect recognition is also useful for identifying a speaker's regional origin and ethnicity and would be helpful in forensic speaker profiling [103]. I used the accent embedding produced by accent recognition models to represent the accent pattern in Objective 3 of this dissertation.

2.8. Golden speaker in pronunciation training

Many second-language (L2) speakers of English emigrate to the United States to work in influential positions within higher education, medicine, and technology-related fields [104-107]. Despite having sound knowledge of English grammar and vocabulary, as well as good reading and listening abilities, their intelligibility can be impaired because of non-native pronunciation [108]. Native-like pronunciation performance becomes more challenging with increasing age. Unfortunately, this problem cannot be easily ignored, since the kinds of highly-skilled fields that attract L2 speakers of English also typically require advanced skills in oral communication, including highly intelligible pronunciation. Because such professionals come from a wide variety of first language (L1) backgrounds, and because their English pronunciation reflects the influence of these L1s, their pronunciation needs are highly individualized. The need for instruction might be best met by offering organized pronunciation classes. However, such classes remain relatively infrequent [109], and even if offered they cannot sufficiently meet the extensive and varied needs of learners. A classroom setting does not allow for significant amounts of one-on-one instruction, which is critical since the pronunciation difficulties of L2 learners vary so widely.

An alternative to organized classes is computer-assisted pronunciation training (CAPT) tools. CAPT is beneficial because it allows L2 learners to work on pronunciation skills at their own pace outside the traditional language classroom. CAPT has many advantages: it can provide diagnoses of difficulties, on-demand modeling of phonological features in the L2, explanations, and as many repetitions as desired by the learner. It can also provide practice with exercises in the same order or mixed up, as well as provide a virtual partner for discourse-level practice, reliable feedback, and the potential of a trained virtual teacher. Unfortunately, most CAPT materials make use of a single model speaker (typically native), which is not necessarily an ideal model for all learners because of the difficulty of imagining oneself speaking with a particular voice (e.g., [110]). In fact, prior research indicates that L2 learners are more likely to succeed when they imitate a speaker with a voice similar to their own, a so-called “golden speaker" [111]. This shortcoming is challenging; to date, no guidelines exist that help identify an ideal golden speaker for each L2 learner.

Prior studies [9, 112-115] also corroborate the effectiveness of golden speaker in CAPT, which report improvements in pronunciation (and intelligibility) if L2 learners are trained on their own utterances resynthesized to match the prosody of a native speaker. This focus on suprasegmentals has been, in part, motivated by technology: modifying the prosody of an L2 utterance is straightforward, e.g., through pitch-synchronous overlap-add (PSOLA) [116]. Although prosody is clearly important, previous studies have shown that segmental errors also have significantly impact on the intelligibility of L2 speech, in particular for segments with high functional load [117-119]. Furthermore, the two types of cues are interdependent in English: it is difficult to change stress or rhythmic features without a corresponding change to segmental features such as vowel quality [120]. It may even be said that the changes in suprasegmental and segmental features are often redundant features for listeners, each reinforcing the other's impact [121]. Thus, there is a need for speech-processing methods that allow both types of accent conversion (segmental and suprasegmental) to be performed. Meanwhile, the algorithms proposed in the third objective of this dissertation performs accent conversion considering both segmental and suprasegmental cues. Therefore, it would be meaningful to explore the effectiveness of using the proposed accent conversion algorithm to produce the golden speaker voices in pronunciation training.

3. FEW-SHOT VC BASED ON SPARSE REPRESENTATION: LEARNING STRUCTURED SPARSE REPRESENTATIONS FOR VOICE CONVERSION[*]

3.1. Overview

Sparse-coding techniques for voice conversion assume that an utterance can be decomposed into a sparse code that only carries linguistic content, and a dictionary of atoms that captures the speakers' characteristics. However, conventional dictionary-construction and sparse-coding algorithms rarely meet this assumption. The result is that the sparse code is no longer speaker-independent, which leads to lower voice-conversion performance. In this paper, we propose a Cluster-Structured Sparse Representation (CSSR) algorithm that improves the speaker-independence of the representations. CSSR consists of two complementary components: a Cluster-Structured Dictionary Learning module that groups atoms in the dictionary into clusters, and a Cluster-Selective Objective Function that encourages each speech frame to be represented by atoms from a small number of clusters. We conducted three experiments on the CMU ARCTIC corpus to evaluate the proposed method. In a first ablation study, results show that each of the

[*] © 2019 IEEE. Reprinted, with permission, from S. Ding, G. Zhao, C. Liberatore, and R. Gutierrez-Osuna, "Learning Structured Sparse Representations for Voice Conversion," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 343-354, 2019. In reference to IEEE copyrighted material, which is used with permission in this dissertation, the IEEE does not endorse any of Texas A&M University's products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink. This reprint contains necessary modifications to include suggestions from the dissertation committee.

two CSSR components enhance speaker independence, and that combining both components leads to further improvements. In a second experiment, we find that CSSR uses increasingly larger dictionaries more efficiently than phoneme-based representations by allowing finer-grained decompositions of speech sounds. In a third experiment, results from objective and subjective measurements show that CSSR outperforms prior voice-conversion methods, improving both acoustic quality and voice identity of the synthesized speech. Finally, we show that the CSSR captures latent (i.e., phonetic) information in the speech signal.

3.2. Introduction

Voice Conversion (VC) aims to transform the speech of a source speaker to sound as if a target speaker had produced it. VC finds use in a number of applications, such as personalized text-to-speech synthesis [1], pronunciation training [4], and speaker spoofing [122]. Various approaches have been proposed to perform VC. Statistical parametric methods based on Gaussian Mixture Models (GMM) [10, 11] and Deep Neural Networks (DNN) [12, 13, 35, 36, 39] are widely used and can achieve convincing results. A promising alternative to GMMs and DNNs are methods based on sparse representations [30-32]. A typical method based on sparse representations consists of a dictionary construction step (to encode the speaker's identity) and a sparse coding step (to encode the content of an utterance). During training, a dictionary consisting of pairs of source and target frames is constructed from a parallel corpus of time-aligned utterances. At runtime, the sparse representation of a source spectrum is computed with respect to the source dictionary, and then the target spectrum is approximated by multiplying the source sparse representation with the target's dictionary. Sparse representation methods have several advantages: they require much smaller training corpora [31] and are more robust to noisy speech than GMMs [30]. As a result, sparse representation methods are particularly appealing in applications where collecting a large corpus is impractical or background noise is inevitable (e.g., pronunciation training [21, 23]).

Sparse representation methods assume that the dictionary captures the speaker identity (i.e., how a speaker produces the various phonetic units), and that the sparse representation is speaker-independent and captures only the linguistic content. In practice, however, satisfying this assumption is difficult. First, the atoms in the dictionary do not fully capture speaker identity, since to do so the dictionary must capture all the phonetic units (e.g., tri-phones), which is not feasible for small corpora. Second, the sparse representation is not speaker independent (even if the dictionary contains all the phonetic units), since the standard sparse coding objective (i.e., Lasso [123]) ignores the phonetic structure of the dictionary. Namely, the Lasso minimizes the Mean-Square-Error using as few atoms as possible (the effect of the L_1 constraint) regardless of their phonetic content, so the sparse representations of the same utterance from different speakers tend to be different. These two factors are compounded, making the sparse representations less speaker-independent. As a result, the similarity between source and target sparse representations decreases, ultimately degrading the sound quality of the VC syntheses.

To address these problems, we propose a novel Cluster-Structured Sparse Representation (CSSR) for spectral transformation in VC. CSSR consists of two components, a Cluster-Structured Dictionary Learning algorithm (CSDL) and a Cluster-

Selective Objective Function (CSOF)[3]. The training and runtime processes are as shown in Figure 3.1 and Figure 3.2. During training, and given a time-aligned corpus, CSDL uses a hard-decision Expectation Maximization algorithm to learn a family of “structured” sub-dictionaries, where atoms (i.e., pairs of source-target acoustic frames) within each sub-dictionary (or cluster) are acoustically similar. At runtime, and given the structured source dictionary that was learned, we compute a structured sparse code for the source utterance by optimizing the CSOF, which uses the $L_{2,1}$ norm [124] to promote group sparsity and therefore tends to represent each speech frame using atoms from as few clusters as possible. Finally, we multiply the source’s structured sparse code with the target’s structured dictionary to generate the voice-converted utterance.

`We conducted three experiments on the CMU ARCTIC corpus [125] to evaluate the proposed method: an ablation study to examine the effectiveness of each component in the CSSR algorithm, an experiment to characterize the performance of CSSR as a

[3] Initial findings from this work were presented at the 19th Annual Conference of the International Speech Communication Association (Interspeech 2018) [17, 18]. The earlier conference papers examined the above two sub-problems individually and presented preliminary results, respectively. In this manuscript, we consider the two sub-problems jointly and propose a state-of-the-art sparse representation-based VC framework. We also describe our methods in full detail and significantly expand the validation experiments and analysis of results.

function of the number of atoms in the dictionary, and a set of objective and subjective studies to compare the proposed method against baselines from previous studies. The results of the ablation study show that both CSDL and CSOF can reduce the difference between source and target sparse representations and improve VC performance, and that combining both components leads to further performance improvements. In addition, results from the second experiment show that CSSR uses increasingly larger dictionaries more efficiently than phoneme-based representations by allowing finer-grained decompositions of speech sounds. Lastly, results from the objective and subjective studies show that CSSR significantly improves both acoustic quality and voice identity when compared to the baseline systems. In our final analyses, we provide a phonetic interpretation of CSSR using the ground-truth phonemes labels of the speech corpus.

The rest of the chapter is organized as follows. Section 3.3 reviews mainstream methods for VC, structured dictionary learning and sparse coding, how previous VC methods improve the speaker independence of sparse representations, and the relation of the proposed method to previous work. Section 3.4 describes the proposed method, including the overall VC framework, CSDL, and CSOF. Section 3.5 describes the experimental setup, including the corpus and the details in our implementation. Section 3.6 shows results for two sets of experiments. Finally, we conclude the paper with a thorough discussion of the results, a phonetic interpretation of the method, and future directions of work.

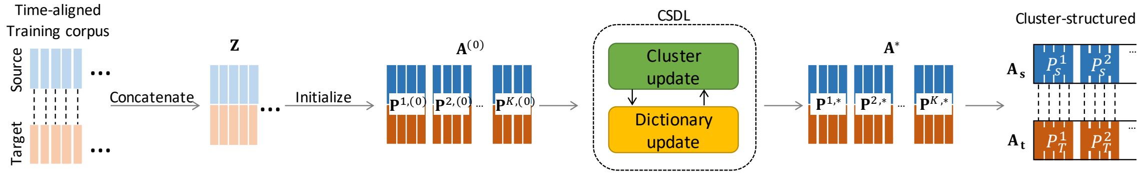


Figure 3.1: Training phase of CSSR. A source and a target utterances from training corpus are first time-aligned using dynamic time warping. The time-aligned frames are then concatenated, and the structured dictionaries are randomly initialized with the concatenated frames as $\mathbf{A}^{(0)}$. Then, CSDL performs two steps (cluster update and dictionary update) iteratively until convergence. The optimal structured dictionaries, $\mathbf{A}^{*}$, are then split into a source dictionary $\mathbf{A}_s$ and a target dictionary $\mathbf{A}_t$.

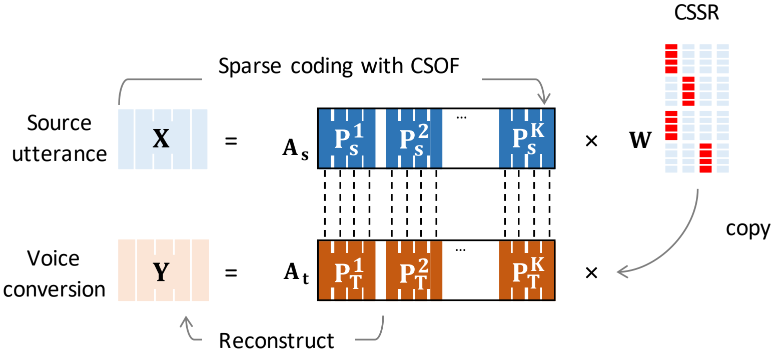


Figure 3.2: Testing phase of CSSR. The CSSR *W* for the source utterance *X* is computed relative to the source structured dictionary $\mathbf{A}_S$. The converted utterance is then generated by multiplying the CSSR W with the target structured dictionary $\mathbf{A}_T$.

3.3. Literature review

3.3.1. Voice conversion algorithms

Statistical parametric models such as Gaussian Mixture Models and Deep Neural Networks are among the most common algorithms for VC. GMM-based methods [1, 10] learn the joint distribution of source and target short-time spectra and then estimate the target spectral features through least-squares regression. However, the basic GMM-based method suffers from over-smoothing issues [11, 126] on the generated feature sequences. To address this problem, Toda et al. [11] proposed to use maximum likelihood parameter generation (MLPG) as a post-processing step for GMM-based methods. Furthermore, global variance (GV) is often combined with MLPG to increase the quality of the synthesized speech [11].

By contrast, DNN-based methods map the source spectral features directly into the target space through various network structures such as restricted Boltzmann machines [12], auto-encoders [35], feed-forward neural networks [13], and recurrent neural networks [36]. More recently, Phonetic Posteriorgrams [19, 45] from acoustic models, generative models including Generative Adversarial Networks [48, 127] and Variational Auto-Encoders [39, 42, 51] have been shown to enhance VC performance. These methods can solve more generalized VC problems such as many-to-many VC and non-parallel VC, but they require relatively large corpora. Other statistical models such as partial least squares [128] and Hidden Markov Models [129] have also shown success in VC tasks.

Methods based on non-parametric sparse representations have received much attention in recent years. Unlike statistical parametric methods, sparse representation methods require much smaller training corpora and are more robust to noisy speech. Takashima et al. [30] first applied sparse representations to perform VC in noisy environments. Following this work, subsequent studies focused on improving either the dictionary construction or the sparse coding process. Wu et al. [31] improved the original sparse representation by using both high-resolution and low-resolution features to capture spectral details and enforce temporal continuity. Zhao and Gutierrez-Osuna [130] proposed different strategies to construct more compact and effective dictionaries, while Fu et al. [131] used a dictionary learning algorithm to improve the effectiveness of the dictionary. Aihara et al. [32, 132], Sisman et al. [133], and Liberatore et al. [38] incorporated phonetic information in both dictionary construction and sparse coding, which enhanced the speaker independence of the sparse representations. Other innovations have dramatically improved the quality of sparse representation-based VC. Wu et al. [134] and Liberatore et al. [135] showed that warping the source residual and adding it to the estimated target spectra can also significantly improve the VC syntheses quality. Wu et al. [136] generalized MLPG and GV into sparse representation methods via an approximation algorithm, which also improved the quality of the converted speech.

3.3.2. Structured sparse coding and dictionary learning

Signals such as images and speech are highly correlated and always have internal structures. However, the standard sparse coding objective functions (i.e., Lasso) do not consider any prior information about the internal structure of the data. To take such information into account, various structured-sparse representations have been proposed. Yuan et al. [137] first proposed the Group Lasso based on distinct groups (e.g. variables of different categories) and provided two algorithms to solve the Group Lasso. Group-sparse representations have also been generalized to include trees and graph structures [138-141]. Accordingly, a number of algorithms have been proposed to learn dictionaries with group structures, such as the Alternating Minimization fashioned algorithm [77], Proximal methods [78], and online dictionary learning algorithms [79]. Given the internal structures of the data, these structured sparse representations are more flexible and accurate than conventional sparse representations. The structured sparse representations have proven to be successful in various computer vision and speech processing tasks such as face recognition [79, 80], image classification [77, 81], speech enhancement [82, 83], speech recognition [84], and source separation [85].

3.3.3. Improving the speaker independence of the sparse representations in VC

Several previous studies have proposed solutions to improve the speaker independence of the sparse representation. Aihara et al. first examined this problem and provided different solutions [32, 132, 142]. In [32], they used phoneme information to regularize the sparse representation and attempt to make it speaker independent. Namely, they categorized the atoms into sub-dictionaries according to their phoneme labels and then selected different sub-dictionaries to represent the speech frames. In [142], they proposed an activity-mapping non-negative matrix factorization algorithm to introduce mappings between the source and target sparse representations. To further reduce the computational complexity while enforcing speaker independence, they proposed a parallel dictionary learning algorithm [132] with a graph-embedded discriminative constraint. Sisman et al. [133] followed [32] in building phoneme-categorized dictionary but selected sub-dictionaries using phoneme labels at runtime, which also improved the speaker independence of the sparse representations. In related work, Liberatore et al. [38] used the centroids of each phoneme as atoms and constructed a more compact dictionary. The more compact dictionary prevented the source and target sparse representations from becoming too different, which implicitly improved the speaker independence.

3.3.4. Relation to prior work

Our proposed method differs from prior studies in several respects. First, CSDL learns the dictionaries directly from the data, without any supervised information (e.g., phoneme labels [32, 132, 133], etc.) It avoids the use of forced-alignment or automatic speech recognition to generate the labels, thus reducing computation. Second, CSDL is based on “hard-decision Expectation Maximization” algorithms [143-147] commonly used for learning models that depend on unobserved latent variables, which is different from previous dictionary learning algorithms [77-79] and previous dictionary learning VC methods [131, 132]. Finally, we use a CSOF to implicitly encourage the sparse coding algorithm to represent a speech frame using a compact set of atoms from a few clusters, rather than using a sub-dictionary selection procedure [32] or phoneme labels [133] at runtime. Additionally, CSDL and CSOF are complementary: CSDL learns a cluster-structured dictionary, and CSOF enforces the group-sparsity on the structured dictionary. The resulting structured sparse representation captures the internal structure of speech signals, which makes the representation more speaker independent. As a result, the VC performance is significantly improved.

3.4. Method

In the following sub-sections, we first introduce the entire VC framework based on the CSSR. Then, we provide a detailed derivation of the two components of the CSSR: the CSDL and the CSOF.

3.4.1. Voice conversion framework

First, we describe the conventional sparse representation method used in VC. During training, a source dictionary $\mathbf{A_s} \in \mathbb{R}^{D\times N}$ and a target dictionary $\mathbf{A_t} \in \mathbb{R}^{D\times N}$ are learned from time-aligned parallel utterances, where N is the number of atoms in each dictionary, and each atom is a D-dimensional vector. Note that the requirement of parallel utterances can be relaxed by using alignment algorithm such as those in [16, 23]. At runtime, an L-frame source utterance $\mathbf{X} \in \mathbb{R}^{D\times L}$ is represented as,

$$\mathbf{X} \cong \mathbf{A_s W} \tag{3.1}$$

where $\mathbf{W} \in \mathbb{R}^{N\times L}$ is a sparse non-negative weight matrix (i.e., a sparse representation). Given $\mathbf{X}$ and $\mathbf{A_s}$, $\mathbf{W}$ can be approximated via solving standard sparse coding objective (i.e., Lasso):

$$\mathbf{W} = \underset{\mathbf{W}}{\operatorname{argmin}}\, d(\mathbf{X}, \mathbf{A_s W}) + \alpha \|\mathbf{W}\|_1, \qquad s.t.\ \mathbf{W} \geq 0 \tag{3.2}$$

where $d(\cdot)$ is a distance metric, typically the KL-divergence or the Euclidean distance. The L_1 norm term is often included to enforce sparsity in $\mathbf{W}$, with α being a sparsity penalty. Given $\mathbf{A_t}$ and $\mathbf{W}$, a target utterance $\widehat{\mathbf{Y}} \in \mathbb{R}^{D\times L}$ can be generated as:

$$\widehat{\mathbf{Y}} = \mathbf{A_t W} \tag{3.3}$$

Voice conversion using CSSR. Compared to the conventional sparse representation used for VC, CSSR further considers that the speech signal has an internal structure (i.e., phonetic). Assume that the spectral frames of a speaker can be divided into K clusters. During training, we use the CSDL algorithm (described in section 3.4.2) to learn the structured dictionaries $\mathbf{A_s}$ and $\mathbf{A_t}$, each containing K sub-dictionaries:

$$\mathbf{A_s} = [\mathbf{P}_\mathbf{s}^1, \mathbf{P}_\mathbf{s}^2, \cdots, \mathbf{P}_\mathbf{s}^K] \tag{3.4}$$

$$\mathbf{A_t} = [\mathbf{P}_\mathbf{t}^1, \mathbf{P}_\mathbf{t}^2, \cdots, \mathbf{P}_\mathbf{t}^K] \tag{3.5}$$

where $\mathbf{P}_\mathbf{s}^i \in \mathbb{R}^{D\times M}$ and $\mathbf{P}_\mathbf{t}^i \in \mathbb{R}^{D\times M}$denote the source and the target sub-dictionaries corresponding to the i-th cluster, respectively, and and $i \in \{1, 2, \cdots, K\}$. M is the number of atoms in a sub-dictionary.

At runtime, once the structured dictionaries have been learned, we generate the CSSR $\mathbf{W}$ by jointly minimizing the objective function in eq. (3.2) and CSOF $\Psi(\mathbf{W})$:

$$\mathbf{W} = \underset{\mathbf{W}}{\operatorname{argmin}}\, d(\mathbf{X}, \mathbf{A_s W}) + \alpha\|\mathbf{W}\|_1 + \beta\Psi(\mathbf{W}), \qquad s.t.\ \mathbf{W} \geq 0 \tag{3.6}$$

where β is a penalty term for $\Psi(\mathbf{W})$, which is based on the $L_{2,1}$ norm [124]; see section 3.4.3 for details. CSOF *implicitly* encourages the sparse coding algorithm to represent a speech frame using atoms from as few clusters as possible, which as we will later show to encode phonetic information. With the target dictionary $\mathbf{A_t}$ and the computed CSSR $\mathbf{W}$, we then use eq. (3.3) to estimate the target spectrum.

3.4.2. Cluster-structured dictionary learning

Let $\mathbf{X} \in \mathbb{R}^{D\times L}$ and $\mathbf{Y} \in \mathbb{R}^{D\times L}$ denote the time-aligned source and target training utterances. Following Fu et al. [131], we concatenate the time-aligned source and target training utterances as $\mathbf{Z} = [\mathbf{X}, \mathbf{Y}]^T$. Our goal is to learn a concatenated dictionary $\mathbf{A} = [\mathbf{A_s}, \mathbf{A_t}]^T$, where $\mathbf{A_s}$ and $\mathbf{A_t}$ consist of sub-dictionaries, as defined in eqs. (3.4-3.5). For notational simplicity, we define the concatenated sub-dictionary as $\mathbf{P}^i = \left[\mathbf{P}_\mathbf{s}^i, \mathbf{P}_\mathbf{t}^i\,\right]^T$, and $\mathbf{A} = [\mathbf{P}^1, \mathbf{P}^2, \dots, \mathbf{P}^K]$. We solve this dictionary-learning problem through an iterative algorithm. At each iteration, we perform two steps: a cluster update and a dictionary update. Details of each step are provided in following sub-sections. The overall algorithm is summarized in Algorithm 3.1.

Algorithm 3.1: CSDL algorithm

```
Inputs: concatenated training utterances $\mathbf{Z}$, the number of clusters $K$
Outputs: learned structured dictionary $\mathbf{A}^{*} = [\mathbf{P}^{1,*}, \mathbf{P}^{2,*}, \dots, \mathbf{P}^{K,*}]$
Initialization: randomly assign a latent cluster label to each training
frame and divide the training frames to $K$ clusters according to the
latent cluster labels, as in eq. (3.10). Then initialize the dictionary
$\mathbf{A}^{(0)} = \left[\mathbf{P}^{1,(0)}, \mathbf{P}^{2,(0)}, \dots, \mathbf{P}^{K,(0)}\right]$ by solving eq. (3.11).
Repeat until convergence:
  Cluster update:
  1. compute $\mathbf{w}_l^{i,(t)}$ by solving eq. (3.8)
  2. compute $r_l^{i,(t)}$ as in eq. (3.7)
  3. assign each training frame $\mathbf{z}_l$ a latent cluster label $p_l^{(t)}$ as in (3.9)
  4. divide the training data into $K$ clusters as in (3.10).
  Dictionary update:
  5. update each sub-dictionary $\mathbf{P}^{i,(t+1)}$ in $\mathbf{A}^{(t+1)}$ by solving eq. (3.11).
Return $\mathbf{A}^{*} = [\mathbf{P}^{1,*}, \mathbf{P}^{2,*}, \dots, \mathbf{P}^{K,*}]$
```

3.4.2.1. Cluster update

We denote the concatenated dictionary and sub-dictionary at the t-th iteration as $\mathbf{A}^{(t)}$ and $\mathbf{P}^{i,(t)}$. In the cluster update step, all the sub-dictionaries $\mathbf{P}^{i,(t)}$ are fixed. For each frame $\mathbf{z}_l$ in $\mathbf{Z}$, we assign $\mathbf{z}_l$ to the cluster $\mathbf{P}^{i,(t)}$ whose sub-dictionary represents $\mathbf{z}_l$ with the lowest residual error. Formally, we denote the residual of $\mathbf{z}_l$ respect to $\mathbf{P}^{i,(t)}$ as,

$$r_l^{i,(t)} = \left\|\mathbf{z}_l - \mathbf{P}^{i,(t)}\mathbf{w}_l^{i,(t)}\right\|_2^2 \tag{3.7}$$

where $\mathbf{w}_l^{i,(t)}$ are the coefficients of the sparse representation. We compute $\mathbf{w}_l^{i,(t)}$ as,

$$\mathbf{w}_l^{i,(t)} = \operatorname*{argmin}_{\mathbf{w}} \left\|\mathbf{z}_l - \mathbf{P}^{i,(t)}\mathbf{w}\right\|_2^2 + \lambda\|\mathbf{w}\|_1 \tag{3.8}$$

which we solve using the Least Angle Regression (LARS) [148] algorithm, and λ is the sparsity penalty. Once the residuals are updated, we can assign $\mathbf{z}_l$ a latent cluster label $p_l^{(t)}$ as,

$$p_l^{(t)} = \underset{i}{\operatorname{argmin}}\, r_l^{i,(t)} \tag{3.9}$$

Then, we divide $\mathbf{Z}$ into K clusters based on their labels $p_l^{(t)}$ as,

$$\mathbf{Z}^{i,(t)} = \left\{\mathbb{I}\left(p_l^{(t)} = i\right)\mathbf{z}_l\right\}, \qquad l = 1, 2, \dots, L \tag{3.10}$$

where $\mathbf{Z}^{i,(t)}$ denotes all the speech frames in the i-th cluster, and $\mathbb{I}(\cdot)$ is the indicator function.

3.4.2.2. Dictionary update

In the dictionary update step, we fix the clusters and update the sub-dictionaries. For each $\mathbf{Z}^{i,(t)}$ (all the speech frames in the i-th cluster), we wish to find a sub-dictionary $\mathbf{P}^{i,(t+1)}$ that can represent it sparsely with minimum residual. In other words, for each sub-dictionary $\mathbf{P}^{i,(t+1)}$ we solve the problem:

$$\mathbf{P}^{i,(t+1)} = \underset{\mathbf{P}^i}{\operatorname{argmin}} \left\|\mathbf{Z}^{i,(t)} - \mathbf{P}^i\mathbf{w}\right\|_2^2 + \lambda\|\mathbf{w}\|_1 \tag{3.11}$$

which we solve using the online dictionary-learning algorithm proposed by Mairal et al. [149].

3.4.3. Cluster-selective objective function

The proposed objective function (CSOF) is a generalization of the Phoneme-Selective Objective Function (PSOF) we proposed in previous work [150]. PSOF promotes that each speech frame is represented with atoms from a small number of phonemes, which is achieved by enforcing group sparsity on the groups defined by phoneme labels (one group per phoneme). However, PSOF is limited by the fact that phonemes are often too coarse to capture detailed information in speech (e.g., allophones). To address this issue, CSOF allows the number of clusters to increase, e.g.

as the amount of training data increases. CSOF enforces group sparsity on the groups defined in the cluster-structured dictionary learned from CSDL. In practice, the most common mathematical tool to enforce group sparsity is the $L_{2,1}$ norm. Therefore, we formulate the CSOF $\Psi(\mathbf{W})$ as,

$$\Psi(\mathbf{W}) = \sum_{j=1}^{L} \sum_{k=1}^{K} \sqrt{\sum_{i=1, i \in \mathbf{P}_s^k}^{M} w_{ij}^2} \tag{3.12}$$

where w_{ij} denotes the (i, j)-th element of the weight matrix **W**, K denotes the number of sub-dictionaries (i.e., clusters), $\mathbf{P}_s^k$ represents the k-th sub-dictionary in the source dictionary, L is the number of frames in the utterance and M is the number of atoms in a sub-dictionary (see Section 3.4.1). By minimizing CSOF, we force the weights within a sub-dictionary to be activated or suppressed at the same time, and therefore implicitly encourage the sparse coding algorithm to represent a spectrum frame with atoms from as few clusters as possible.

3.5. Experimental setup

3.5.1. Corpus

We used four English speakers from the CMU ARCTIC [125] corpus: BDL (male), RMS (male), SLT (female), and CLB (female). For each speaker, we selected three sets of utterances: 20 utterances for training (about 1.5 minutes), 10 utterances for validation, and 50 utterances for testing4. Four VC pairs were considered for the experiments: M-M (BDL to RMS), M-F (RMS to SLT), F-F (SLT to CLB), and F-M (CLB to BDL). In what follows, all the results are averaged over these four VC pairs.

3.5.2. Implementation details

We used the WORLD vocoder [151] (D4C edition [152]) to extract a 513-dimensional spectral envelope, fundamental frequency (F0) and aperiodicity for each utterance with a 5ms window shift. We computed the 25-dimensional Mel-Frequency Cepstrum (MFCC) from the WORLD spectral envelope (removing MFCC0, which is the energy) and used the MFCCs as the acoustic feature in dictionary learning and voice conversion. Source and target utterances were time-aligned using dynamic time warping [153].

In the proposed method, we set the number of atoms in each sub-dictionary to 100. In the first and the third experiments, we set the number of clusters (sub-dictionaries) K to 40, i.e., the number of phonemes in CMU ARCTIC (except for silence). In the second experiment, we explored different number of clusters (sub-dictionaries), as will be described in Section 3.6.2. For silent frames, we used a voice activity detector to find them and directly copy silent frames from source to target. We

[4] A small number of training utterances was used to mimic a low-resource setting. Utterances for each set were selected using a maximum entropy criterion to ensure good phonetic balance.

used the SPAMS sparse coding toolbox [75, 149] to solve for eqs. (3.6), (3.8) and (3.11). We set α, β, and λ to 0.001, 0.05, and 0.01, respectively, based on preliminary experiments [150, 154]. CSDL will stop when no more than 5% of training frames are re-assigned from one iteration to the next.

Following Toda et al. [11], we convert the pitch trajectory (F_0) of the source speech to match the pitch range of the target speaker using log mean variance normalization. We estimate the converted spectral envelope from the converted MFCC, and finally synthesize the converted speech using the WORLD vocoder with the converted spectral envelope, converted F0 and source aperiodicity.

3.6. Results

We conducted three experiments to evaluate CSSR. The first experiment was an ablation study that examined the effectiveness of each CSSR component in reducing differences between source and target sparse representations and improving VC performance. In the second experiment, we explored the influence of different number of clusters in CSSR. In the final experiment, we evaluated the VC performance of CSSR and compared it against baselines from previous studies.

3.6.1. Ablation study

To understand how much each CSSR component contributes to reducing the sparse representations difference and improving VC performance, we conducted an ablation study that evaluates the contribution of each method: the dictionary learning algorithm and the sparse coding cost function. To do so, we compared five different system configurations; also see Table 3.1:

- **Random Dictionary Learning (RDL) + Lasso**: a baseline system following the conventional VC framework based on sparse representations [30], which constructs dictionaries from randomly selected speech frames in training, and optimizes the Lasso (eq. (3.2)) at runtime.

- **Phoneme Structured Dictionary Learning (PSDL) + Lasso**: a system that constructs the structured dictionary using phoneme labels during training (as in [32, 133, 150]) and optimizes the Lasso at runtime.
- **CSDL + Lasso**: a system that uses the CSDL algorithm to learn a cluster structured dictionary in training, and optimizes the Lasso at runtime.
- **PSDL + CSOF**: a system that constructs the structured dictionaries using phoneme labels during training and optimizes the joint cost function in eq. (3.6) at runtime.
- **CSDL + CSOF (CSSR)**: the *proposed* method: CSDL and CSOF combined.

Table 3.1. The five system configurations used in the ablation study

		Dictionary construction technique		
		Random	**Phoneme**	**CSDL**
Objective function	**MSE+L_1 (Lasso)**	RDL+Lasso	PSDL+Lasso	CSDL+Lasso
	MSE+L_1+ $L_{2,1}$ (CSOF)	N/A[5]	PSDL+CSOF	CSDL+CSOF

[5] We do not consider the combination RDL+CSOF since CSOF requires a structured dictionary, which cannot be randomly selected.

RDL+Lasso, PSDL+Lasso, and CSDL+Lasso share the same sparse coding cost function (MSE + L_1 norm) but differ in the dictionaries: random vs. derived from phoneme labels vs. learned via CSDL. This allows us to assess the relative merits of each dictionary construction technique. PSDL+Lasso and PSDL+CSOF share the same dictionary but differ in the sparse coding cost functions. This allows us to compare the two cost functions side by side. Finally, by comparing CSSR (i.e., CSDL+CSOF) against CSDL+Lasso and PSDL+CSOF we can evaluate the benefit of combining the two proposed algorithms.

We used two metrics to evaluate the five systems: the distance between the source and target sparse representations, which measures whether the representations are speaker dependent, and the Mel-Cepstral Distortion between the synthesized speech and the ground-truth target speech:

- ***Sparse Representation Distance***. As discussed by Aihara et al. [32, 132], the loss of speaker independence decreases the similarity between source and target sparse representations. Accordingly, we compute the difference between source and target sparse representations of time-aligned parallel utterances as,

$$D(W_S, W_T) = \frac{1}{T} \|\mathbf{W_s} - \mathbf{W_t}\|_F \tag{3.13}$$

  where $\mathbf{W_s} \in \mathbb{R}^{N\times T}$ and $\mathbf{W_t} \in \mathbb{R}^{N\times T}$ are the source and target sparse representations, T is the number of frames, and $\|\cdot\|_F$ denotes the Frobenius norm. The lower this distance is, the more similar the source and target sparse representations are, and so the sparse representation tends to be more speaker independent.

- ***Mel-Cepstral Distortion (MCD)***. We also measured the MCD of the voice-converted speech and its time-aligned target speech to examine the effect of sparse representation dissimilarity on synthesis quality. MCD is the most common objective measurement in VC systems, and is defined as,

$$\mathrm{MCD[dB]} = \frac{10}{\ln 10}\sqrt{2\sum_{d=1}^{24}(y_d - \hat{y}_d)^2} \quad (3.14)$$

where $\hat{y}_d$ and y_d are the d-th Mel-Cepstral coefficient of the converted speech and the time-aligned ground-truth target speech, respectively. Lower MCD indicates that the converted speech is closer to the time-aligned target speech.

Results for the sparse representation distance are shown in Figure 3.3 (a). From the results, we found that the sparse representation distance for CSSR (CSDL +CSOF) (0.37) is lower than that of the baselines: CSSR achieves 16.0% relative improvement over PSDL+CSOF (0.44), 68.4% relative improvement over CSDL+Lasso (1.17), 69.4% relative improvement over PSDL+Lasso (1.21), and 70.2% relative improvement over RDL+Lasso (1.24). These results indicate that CSSR systematically increases the similarity between the source and target sparse representations. Additionally, our results show that the system using CSDL (CSDL+Lasso) and the system using CSOF (PSDL+CSOF) outperform their corresponding baselines (RDL+Lasso and PSDL+Lasso), respectively. These results suggest that both CSDL and CSOF are essential in reducing the representation distance. Moreover, we found that CSSR outperforms both CSDL+Lasso and PSDL+CSOF, which indicates that combining CSDL and CSOF lead to further reductions in representation distance. Finally, we also

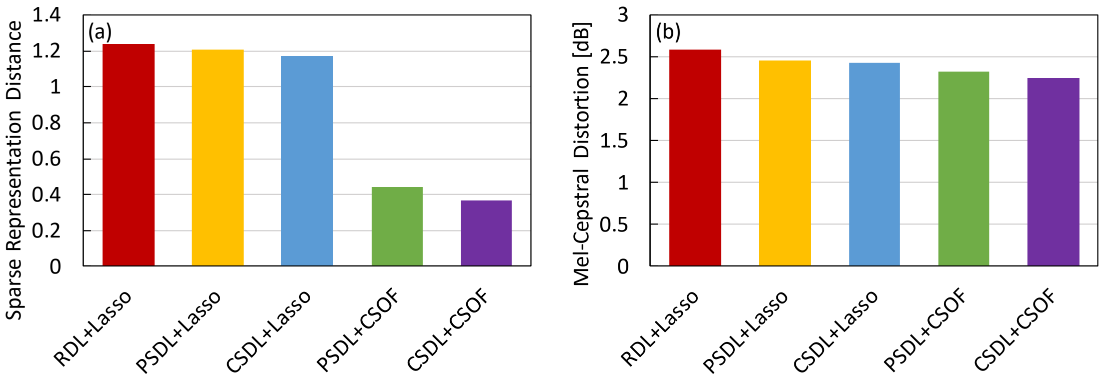


Figure 3.3: (a) Sparse representation distance of all the systems in the ablation study. As defined in eq. (3.13), lower distance means higher similarity between the source and target sparse representations (i.e., improved speaker indencence). (b) Average MCD of all the systems in the ablation study. Lower MCD generally leads to better VC performance.

found that the sparse coding cost function (CSOF) is more effective than the dictionary construction algorithm (CSDL) in reducing representation distance, and hence in improving speaker independence. A possible explanation for this result is that in CSDL+Lasso, the objective function (Lasso) ignores the phonetic structure of the dictionary and minimizes the Mean-Square-Error using as few atoms as possible regardless of their phonetic content.

Results for the Mel-Cepstral Distortion are shown in Figure 3.3 (b). CSSR systematically achieves lower MCD (2.25) than all the baseline systems: a 3.0% relative improvement over PSDL+CSOF (2.32), 7.4% relative improvement over CSDL+Lasso (2.43), 8.5% relative improvement over PSDL+Lasso (2.46), and 13.1% relative improvement over RDL+Lasso (2.59). These results suggest that using CSDL and CSOF individually can improve the voice-conversion syntheses, but that combining the two modules leads to further improvements. Although CSDL only achieves modest reductions in representation distance, it does significantly decrease the MCD. This result shows that the deliberately learned atoms can reduce misalignments and better capture the structure of speech (see Section 3.6.2 below), which also considerably enhances the voice-conversion syntheses.

3.6.2. Effect of dictionary size

In a second experiment, we characterized the performance of CSSR as a function of the number of atoms in the dictionary. Namely, we fixed the number of atoms in each sub-dictionary (cluster) to $M = 100$ while varying the number of sub-dictionaries $K = \{10, 20, 30, \dots, 100\}$, so the total number of atoms in the dictionary was $N = \{1000, 2000, 3000, \dots, 10000\}$. For comparison purposes, we used PSDL+CSOF as a baseline. Because the number of sub-dictionaries in PSDL+CSOF is fixed to 40 (defined by phoneme labels in CMU ARCTIC, except for silence), we increased the number of atoms in each sub-dictionary so the total number of atoms was equal among the two systems.

Results are shown in Figure 3.4 in terms of the average MCD of the two systems as a function of the total number of atoms. In both cases, the MCD decreases with increasing dictionary size. The MCD of the baseline system is systematically higher, and reaches a plateau of 2.31 after 3,000 atoms. In contrast, the MCD of the proposed system continues to decrease past that point, stabilizing at 2.18 with 80 sub-dictionaries or more. These results show that CSSR uses a given dictionary size more effectively by allowing a more fine-grained representation of the data (i.e., more sub-dictionaries) as the number of atoms in the dictionary increase. In other words, for a sufficiently large dictionary size, it is more effective to increase the number of sub-dictionaries (by fixing

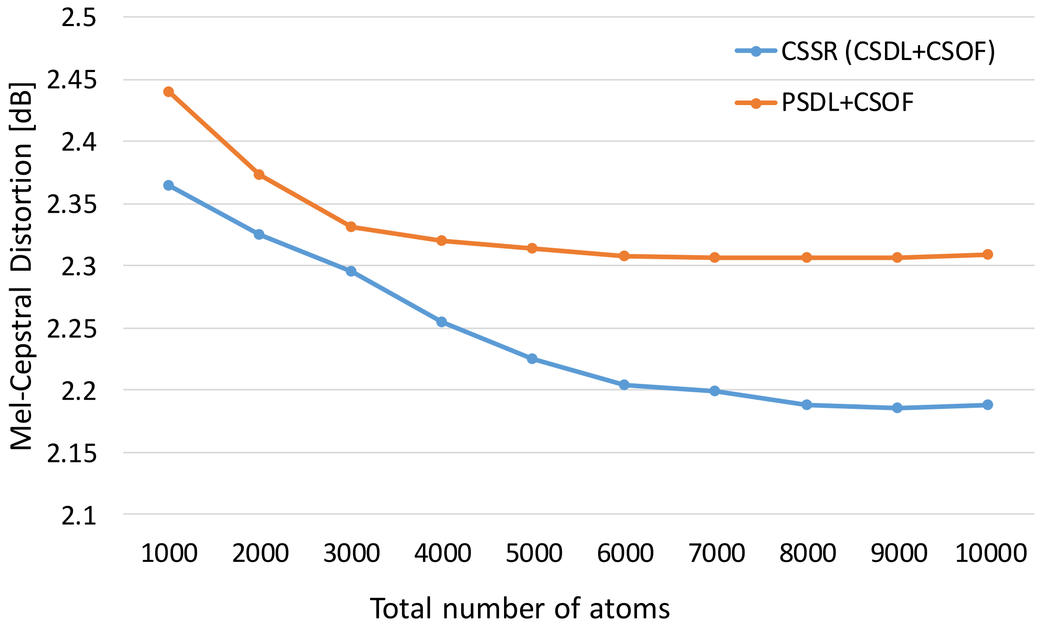


Figure 3.4: Average MCD of CSSR and PSDL+CSOF with different number of atoms in total. In CSSR, we fixed the number of atoms in each sub-dictionary to 100, varying the number of sub-dictionaries from 10 to 100. In PSDL+CSOF, we fixed the number of sub-dictionaries to 40 (the number of phonemes in CMU ARCTIC, except for silence), varying the number of atoms in each sub-dictionary from 25 to 250. Lower MCD generally leads to better VC performance.

the number of atoms per cluster) than to increase the number of atoms per sub-dictionary (by fixing the number of sub-dictionaries).

3.6.3. Voice conversion performance

In a third experiment, we evaluated the voice-conversion performance of the CSSR using objective and subjective measures, and compared it against three existing systems:

- **System 1:** The method we proposed in [150], which constructs the structured dictionaries using phoneme labels during training and jointly optimizes the standard cost function along with the $L_{2,1}$ norm (eq. (3.6)) at runtime.
- **System 2**: The method we proposed in [154], which learns the structured dictionary in

the joint source-target space without supervision (i.e., without phoneme labels) during training and selects the most likely sub-dictionary [32] using the standard cost function (eq. (3.2)) at runtime.

- **Baseline** [11]: A GMM-based VC method that models the joint distribution of source and target speech frames.

By comparing the proposed method (CSSR) against the two previous systems [150, 154], we aim to determine if the two algorithms are complementary. We did not include other sparse representation-based baseline methods (e.g., [31, 32]) in the comparison, since our two previous systems [150, 154] had outperformed them. We also did not include neural network baselines since they require relatively large training corpus (e.g., [36] used 593 utterances, or about 42 mins), whereas our training corpus consists of 20 utterances (or about 1.5 minutes). Instead, we used GMM-based method, which is one of the most common methods in this low-resource setting. For all three sparse representation-based methods (CSSR, System 1 and System 2), we used 40 sub-dictionaries and 100 atoms for each sub-dictionary, following the configurations from [150, 154]. For the GMM, we used 40 mixture components, the same as the proposed method to ensure a fair comparison. We did not use Maximum Likelihood Parameter Generation (MLPG) and Global Variance (GV) in any system to make the results comparable to those presented in [150, 154], but these techniques can be further incorporated to enhance the voice-conversion synthesis. Audio samples are available at https://shaojinding.github.io/samples/cssr/cssr_demo

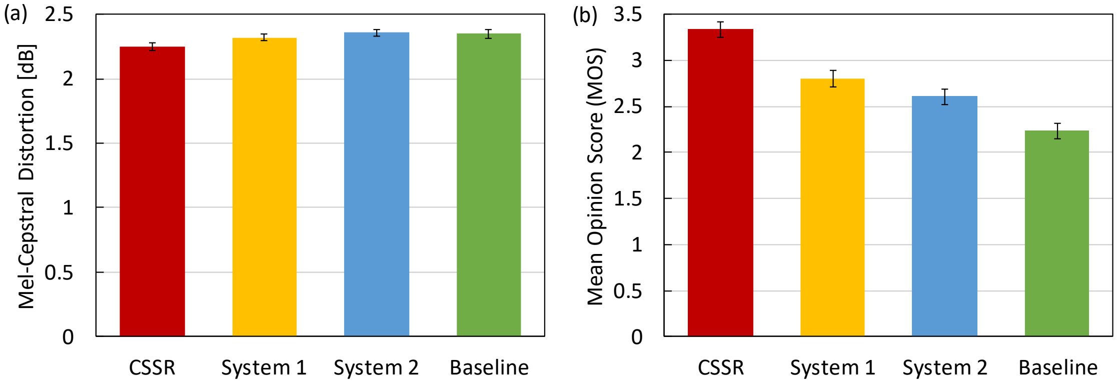


Figure 3.5: (a) Average MCD of the proposed method (CSSR) and three existing systems (System 1, System 2, and Baseline). Lower MCD generally leads to better VC performance. (b) Mean Opinion Scores (MOS) of CSSR and three baselines (System 1, System 2, and Baseline). MOS ranges from 0 to 5, with larger MOS indicating higher acoustic quality. The error bars show 95% confidence intervals.

3.6.3.1. Objective evaluation

First, we compared the four systems by computing the MCD between the converted speech and the time-aligned target speech. Figure 3.5 (a) summarizes the results. CSSR achieved the lowest MCD (2.25) and outperformed all three existing systems (System 1: 2.32, 3.0% relative improvement, single-tail t-test, $p \ll 0.001$; System 2: 2.36, 4.7% relative improvement, single-tail t-test, $p \ll 0.001$; Baseline: 2.35, 4.3% relative improvement, single-tail t-test, $p \ll 0.001$).

3.6.3.2. Subjective evaluation

In a final step, we conducted listening tests on Amazon Mechanical Turk to provide a subjective evaluation of the four systems. We measured acoustic quality with a 5-point Mean Opinion Score (MOS) test and speaker identity with a Voice Similarity Score (VSS) test ranging from -7 (definitely different speakers) to +7 (definitely the same speaker) [155].

Mean Opinion Score. Twenty-seven participants rated 92 utterances from the four systems: 20 utterances per system, 5 utterances per speaker pair plus 12 calibration utterances to detect if participants were cheating and remove them if they did [156]. We exclude ratings of the calibration utterances from the data analysis. Figure 3.5 (b) shows the Mean Opinion Scores of the four methods with 95% confidence intervals. The proposed method (CSSR) obtains a 3.34 MOS, which is higher than that of the other three systems with statistical significance: System 1 (2.80 MOS; 19.3% relative improvement; single-tail t-test, $p \ll 0.001$), System 2 (2.61 MOS; 28.0% relative improvement; single-tail t-test, $p \ll 0.001$), and GMM (2.23 MOS; 49.8% relative improvement; single-tail t-test, $p \ll 0.001$). These results show that combining the proposed dictionary construction algorithm (CSDL) and the proposed sparse coding cost function (CSOF) improves acoustic quality more than applying each technique individually. Additionally, System 1 and System 2 achieve statistically significant improvement over the Baseline (System 1: 25.6% relative improvement, single-tail t-test, $p \ll 0.001$; System 2: 17.0% relative improvement, single-tail t-test, $p \ll 0.001$), which corresponds to the results in [150, 154].

Voice Similarity Score. Twenty-five participants rated 140 utterance pairs: 32 pairs (16 VC-SRC and 16 VC-TGT pairs) for each system and 8 pairs (4 VC-SRC and 4 VC-TGT pairs) for each speaker pair; 12 calibration utterances. A VC-SRC pair consists of a voice-converted (VC) utterance and an utterance randomly selected from the source speaker (SRC), and a VC-TGT pair consists of a voice-converted (VC) utterance and an utterance randomly selected from the target speaker (TGT). We used the utterance that is randomly selected from the source/target speaker to avoid the interference of linguistic content and prosody. For each utterance pair, participants were required to decide whether the two utterances were from the same speaker and then rate their confidence in the decision on a 7-point scale. Following [155], VSS is computed by collapsing the above two fields into a 14-point scale. As shown in Table 3.2, participants were “quite confident” that (1) CSSR utterances and source (SRC) utterances were produced by different speakers (VSS: -5.90); and that (2) CSSR utterances and target (TGT) utterances were produced by the same speaker (VSS: 4.44). When analyzing VC-SRC pairs, we found no statistically significant differences in VSS between CSSR and the other three systems (System 1: p = 0.22; System 2: p = 0.36; Baseline: p = 0.48; two-tail t-test). When comparing VC-TGT pairs, we found no statistically significant differences in VSS between CSSR and the other three systems (System 1: p = 0.27; System 2: p = 0.22; Baseline: p = 0.20; two-tail t-test). Thus, these results indicate that the four methods can produce speech that is different from the source speaker and the same as the target speaker equally well. In Table 3.2, we also presented the VSS of the intra-gender pairs (M-M and F-F) and that of the inter-gender pairs (M-F and F-M). In all cases, we found no statistically significant difference between CSSR and the other three systems. Additionally, we found that the VC-SRC VSS of intra-gender pairs are slightly lower than that of inter-gender pairs. A possible reason is that the pitch ranges of the speakers in intra-gender pairs are closer to each other than those in inter-gender pairs. For inter-gender pairs, pitch (F0) conversion makes the voice-conversion more distinguishable from the source utterances. Moreover, we found that the VC-TGT VSS of inter-gender pairs are lower than that of intra-gender pairs, due to the fact that inter-gender voice conversion is more challenging than intra-gender voice conversion.

Table 3.2: Voice identity results of the proposed method (CSSR) and the three reference systems (System 1, System 2, and Baseline).Voice Similarity Score ranges from -7 (definitely different speakers) to +7 (definitely the same speaker). VC-SRC: VSS between VC and the source speaker; VC-TGT: VSS between VC and the target speaker. All the results are shown as average ± 95% confidence intervals.

System	All pairs		Intra-gender		Inter-gender	
	VC-SRC	VC-TGT	VC-SRC	VC-TGT	VC-SRC	VC-TGT
Baseline	-5.89±0.08	3.92±0.18	-4.93±0.14	4.68±0.14	-6.85±0.04	3.16±0.26
System 1	-6.11±0.07	4.07±0.17	-5.43±0.12	4.95±0.16	-6.78±0.04	3.18±0.23
System 2	-5.80±0.09	4.00±0.18	-4.84±0.16	4.72±0.17	-6.76±0.06	3.24±0.22
CSSR	-5.90±0.09	4.44±0.17	-5.04±0.15	5.32±0.16	-6.76±0.07	3.56±0.24

3.6.4. Phonetic interpretation of CSSR

In a final analysis, we seek to provide a phonetic interpretation of CSSR. In this analysis, we used $\mathrm{K} = 40$ sub-dictionaries and $\mathrm{M} = 100$ atoms for each sub-dictionary. We first analyze the learned cluster-structured dictionaries by exploring the relationship between ground-truth phoneme labels and the learned clusters. In “hard-decision” algorithms, clusters commonly represent latent variables; phonetic clusters can be thought of as latent variables in CSDL. Accordingly, we assigned each speech frame in the training set to the cluster that minimized its residual (eq. (3.7)). In parallel, we used forced-alignment to assign a phoneme labels to each frame, and computed how each phoneme was distributed among the clusters. Then, we matched each phoneme to the cluster that most frequently represented it. The confusion matrix of ground-truth phonemes[6] vs. matched clusters is shown in Figure 3.6. The dark diagonal elements indicate that each cluster is preferentially associated with a single phoneme label. Confusions do occur but are usually restricted to be within the same manner of articulation. For example, the sub-dictionary for cluster “35” represents all the nasals. Likewise, clusters “40”, “31”, “33”, “1”, “34”, “18” are all used for stops. Both “15”, “28” and “11” can represent /EY/, /IH/, /IY/ well, which are all front vowels. In addition, confusions also appear on phonemes that often co-occur, which can be caused by inaccurate forced alignments. For example, “9” is good at representing /ER/ and /R/, which usually co-occur in words ending with “er”. These results indicate that the proposed algorithm can learn the latent (i.e., phonetic) structure of speech and does it so without supervision. The learned latent structures are not restricted to phonemes but emerge directly from the data. Such structures can more accurately capture variability in pronunciations, which can further improve the similarity between source and target sparse representations than methods based on phoneme labels [32, 133, 150].

[6] We used Arpabet to represent the phonemes.

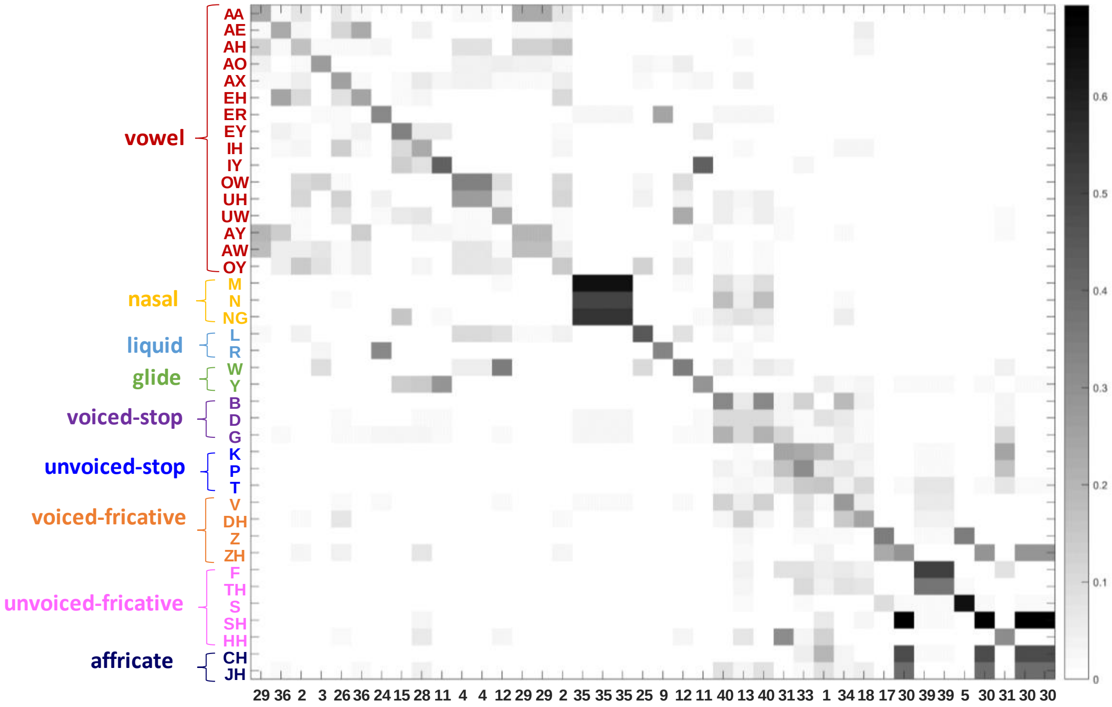


Figure 3.6: Confusion matrix between forced-aligned phoneme labels and the matched clusters. Y-axis values are phonemes (sorted by the manner of articulation), and X-axis values are the cluster IDs.

Next, we visualize the CSSR representation of an utterance to show that it is also phonetically meaningful. We used a similar approach as above to associate the learned clusters with phoneme labels, except each cluster was matched to the phoneme whose frames occurred most frequently in that cluster; this ensured that each cluster was matched with at least one phoneme. Figure 3.7a shows the Structured Sparse Representation of the word “never” from speaker BDL; for clarity, we only show the sub-dictionaries that were activated. As Figure 3.7a shows, the associated phoneme labels of the activated sub-dictionaries correspond to the ground-truth phoneme labels of the word, indicating that the cluster-structured sparse code is phonetically meaningful. Mismatches occur but are mostly in transitions and are restricted to adjacent clusters. For example, in the transition between /EH/ and /V/, atoms from clusters “9”, “24”, “27”,

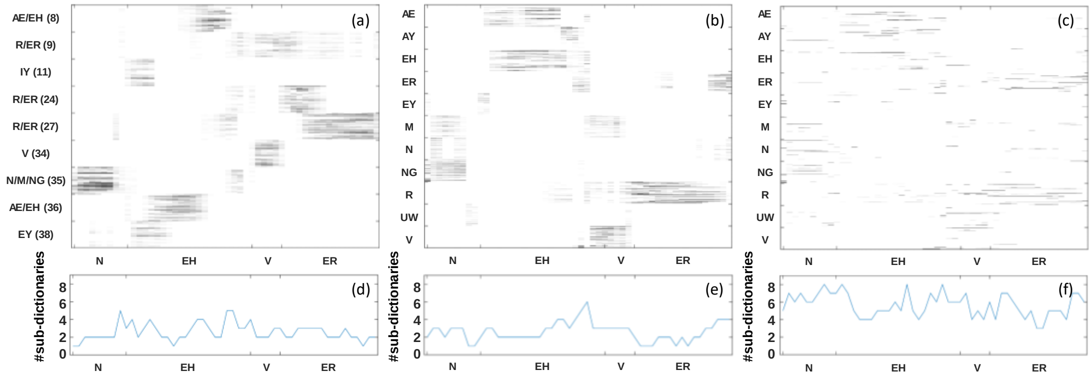


Figure 3.7: Visualization of sparse representations for the word 'never'. (a) CSSR, (b) PSDL+CSOF, (c) PSDL+Lasso. The x-axis denotes the transcription of the word, and the y-axis shows the cluster labels (denoted by numbers) of the sub-dictionaries and the associated phoneme labels. (d), (e), and (f): number of sub-dictionaries that were used in the sparse representations.

and "35" are activated; the speech frames of /ER/ are represented by atoms from clusters "9", "24", and "27", whose associated phoneme labels are all /ER/ and /R/.

Lastly, we compared the representation that emerges from CSSR (Figure 3.7a) with those of PSDL+CSOF (Figure 3.7b) and PSDL+Lasso (Figure 3.7c). To ensure a fair comparison, we set the sparsity penalty of the three systems to 0.05. As shown in Figure 3.7c, when using the Lasso cost function, a speech frame is represented by atoms from arbitrary phoneme labels, and this reduces the interpretability of the representation. Compare this to Figure 3.7a-b, where activation tends to occur on a few clusters/phonemes, as a result of adding the CSOF term to the Lasso. Figure 3.7d-f offers a complementary view of by showing the number of sub-dictionaries activated at each frame of the utterance. CSSR and PSD+CSOF usually activate fewer sub-dictionaries (~2) than PSDL+Lasso (~6 sub-dictionaries).

3.7. Discussions

In previous work [150, 154], we showed that CSDL and CSOF alone could improve voice-conversion performance relative to other sparse representation methods in the literature. This paper corroborates our earlier results and, more importantly, shows that jointly combining CSDL and CSOF can provide further improvements in voice-conversion performance.

In a first ablation study, we evaluated each CSSR component (CSDL and CSOF) by its ability to increase the speaker independence of the representation and reduce the MCD between the synthesized speech and the ground-truth target speech. Our results showed that both CSDL and CSOF are essential in reducing the sparse representation distance and the MCD, corresponding to the results in our previous work. Moreover, we found that combining both (CSSR) leads to further reductions in sparse representation distance and MCD.

In a second experiment, we compared the performance of CSSR against that of our previous system [150] as the number of atoms in the dictionary increases. CSSR increases the number of clusters (sub-dictionaries) in the representation (while keeping the number of atoms per cluster constant) whereas our previous system increases the number of atoms in each cluster (by maintaining the number of clusters constant). Our results show that CSSR is the more effective of the two approaches, as measured by the MCD between the converted speech and the ground truth. Thus, CSSR improves upon our previous work [150] by allowing more fine-grained speech information than phonemes.

In our study, we also evaluated the voice-conversion performance of CSSR through both objective and subjective measurements. We compared CSSR against the two systems from our previous work [150, 154] and against a GMM [11] baseline. In the objective evaluation, results showed that CSSR significantly improved the MCD over the three reference systems. In the subjective evaluation, CSSR was rated to have the highest acoustic quality (in agreement with results from the objective evaluation) and was rated to have the same similarity to the voice identity of the target speaker as the other systems. Additionally, we found that the comparisons between System 1 and Baseline as well as that between System 2 and Baseline are corresponding to the results presented in [150, 154].

In the final analysis, we provided a phonetic interpretation for CSSR by analyzing the cluster-structured dictionary and the CSSR representation. In terms of the cluster-structured dictionary, we visualized he confusion matrix of ground-truth phonemes vs. matched clusters in the cluster-structured dictionary. The results showed that the CSDL can learn the phonetic structure of speech without supervision. In terms of the CSSR representation, we visualized the CSSR representation of the word “never” from speaker BDL From the results, we found that the CSSR is phonetically meaningful. Additionally, when comparing it with PSDL+CSOF and PSDL+Lasso. CSSR and PSDL+CSOF usually activate fewer sub-dictionaries than PSDL+Lasso, which demonstrated the effect of CSOF.

3.8. Conclusions

In this paper, we proposed a Cluster-Structured Sparse Representation (CSSR) for spectral transformation in voice conversion. CSSR consists of two inter-connected components: CSDL and CSOF. CSDL learns a structured dictionary from training utterances, and CSOF produces a structured sparse code at runtime. We conducted three experiments to evaluate CSSR. We first conducted an ablation study to examine the effectiveness of each component in CSSR. Then, we conducted an experiment to characterize the performance of CSSR as a function of the number of atoms in the dictionary. Lastly, we conducted both objective and subjective experiments to evaluate the performance of CSSR and compared it with previous methods. The ablation study showed that both CSDL and CSOF promote the sparse representation to be speaker independent and improve VC performance, and that combining the two components leads to further performance improvements. In addition, results from the second experiment show that CSSR uses increasingly larger dictionaries more efficiently than phoneme-based representations by allowing finer-grained decompositions of speech sounds. Finally, results of objective and subjective studies show that CSSR significantly improves both acoustic quality and voice identity over the previous two systems.

4. ZERO-SHOT VC BASED ON SEQ2SEQ MODEL: IMPROVING THE SPEAKER IDENTITY OF NON-PARALLEL MANY-TO-MANY VOICE CONVERSION WITH ADVERSARIAL SPEAKER RECOGNITION*

4.1. Overview

Phonetic Posteriorgrams (PPGs) have received much attention for non-parallel many-to-many Voice Conversion (VC), and have been shown to achieve state-of-the-art performance. These methods implicitly assume that PPGs are speaker-independent and contain only linguistic information in an utterance. In practice, however, PPGs carry speaker individuality cues, such as accent, intonation, and speaking rate. As a result, these cues can leak into the voice conversion, making it sound similar to the source speaker. To address this issue, we propose an adversarial learning approach that can remove speaker-dependent information in VC models based on a PPG2speech synthesizer. During training, the encoder output of a PPG2speech synthesizer is fed to a classifier trained to identify the corresponding speaker, while the encoder is trained to spoof the classifier. As a result, a more speaker-independent representation is learned. The proposed method is advantageous as it does not require pre-training the speaker classifier, and the adversarial speaker classifier is jointly trained with the PPG2speech

* © 2020 ISCA. Reprinted, with permission, from S. Ding, G. Zhao, and R. Gutierrez-Osuna, " Improving the speaker identity of non-parallel many-to-many voice conversion with Adversarial Speaker Recognition," in *Interspeech*, 2019, pp. 776-780. DOI: 10.21437/Interspeech.2020-1033. This reprint contains necessary modifications to include suggestions from the dissertation committee.

synthesizer end-to-end. We conduct objective and subjective experiments on the CSTR VCTK Corpus under standard and zero-shot VC conditions. Results show that the proposed method significantly improves the speaker identity of VC syntheses when compared with a baseline system trained without adversarial learning.

4.2. Introduction

Voice conversion (VC) aims to convert utterances from a source speaker to make it sound as if a target speaker had produced it. Conventional VC approaches [11-13, 30, 157] usually require training a model for each speaker pair using parallel corpora. Alternative approaches have emerged in recent years that do not require parallel corpora and can build a universal model for all pairs of speakers [19, 39-41, 45-47, 49]. Among these, the Phonetic-PosteriorGram-to-speech (PPG2speech) synthesizer [19, 45-47] has been shown to be effective for non-parallel many-to-many VC. The PPG2speech synthesizer is a sequence-to-sequence (seq2seq) model that transforms PPGs to speech features (e.g., Mel-spectrogram). The PPG2speech synthesizer has an encoder-decoder structure. During training, the encoder learns a speaker-independent hidden representation from input PPGs, and the decoder learns to generate the speech features given the hidden representation and the corresponding speaker embedding (e.g., i-vector [33], d-vector [34]). During inference, the PPG of a source speaker and the speaker embedding of a   target speaker is used to produce VC syntheses.

The PPG2speech synthesizer assumes that the input PPG represents the pronunciation of speech sounds in a speaker normalized space, which is speaker-independent and contains only linguistic information. In practice, however, PPGs still carry speaker identity information such as accent, intonation, and speaking rate [17] that can leak into the voice conversions.

In this work, we address this problem using adversarial learning. Namely, we propose a new training procedure that includes an *adversarial speaker classifier* jointly trained with the PPG2speech synthesizer. During training, the encoder output is fed into the adversarial speaker classifier, and the classifier is optimized to identify the corresponding speaker. At the same time, the encoder is optimized to fool the adversarial speaker classifier. As a result, the encoder outputs become more speaker-independent. The adversarial speaker classifier does not need to be pre-trained. Instead, it is jointly trained with the synthesizer end-to-end, and the minimax optimization in adversarial learning is achieved by back-propagation.

To evaluate the proposed adversarial learning system, we applied it to a state-of-the-art non-parallel many-to-many PPG2speech synthesizer based on Tacotron2 [88]. Then, we tested its effectiveness against the same PPG2speech synthesizer trained without adversarial learning. Using the CSTR VCTK Corpus [158], we conducted both objective and subjective experiments under two testing conditions: *standard* (test speakers were known during training) and *zero-shot* (test speakers were unseen during training). Results show that the proposed method can significantly improve the perceived speaker identity of the VC syntheses in both testing conditions.

The remainder of the chapter is organized as follows. Section 4.3 reviews prior VC approaches, the use of PPG in VC, and the relation of the proposed method to previous studies. Section 4.4 introduces the proposed approach, including the architecture of baseline PPG2speech synthesizer model and the proposed adversarial training scheme. Section 4.5 describes the experimental setup, including the corpus, acoustic model, the speaker recognition model, the neural vocoder, and the PPG2speech synthesizer. Section 4.6 shows results for two sets of experiments. Finally, we conclude the paper in Section 4.7.

4.3. Literature review

Conventional VC frameworks (e.g., based on GMMs [11], sparse representations [30, 157], and DNNs [12, 13]) require time-aligned parallel corpora in training. However, the size of parallel corpora is usually limited (e.g., ~1 hour per speaker in the widely used CMU ARCTIC corpus [125]), and collecting parallel corpora can be laborious and expensive. To overcome this limitation, several non-parallel VC approaches have been proposed, such as the INCA algorithm [37], and various DNN architectures [14-18]. These methods avoid the use of parallel corpora, but they still require training a separate model for each pair of source-target speakers. To address this problem, serveral studies have proposed non-parallel many-to-many VC approaches based on Variational Autoencoders (VAE) [39-42] and the PPG2speech synthesizer [45-47]. One-hot vectors are typically used as speaker embedding, due to its simplicity; several studies [41, 45-47] also explored the use of learned speaker embeddings (e.g., i-vector [33], d-vector [34]) to generalize to unseen speakers (i.e., zero-shot VC).

PPGs have gained much recent attention for VC. Sun *et al.* [19] first proposed to use PPGs for one-to-one VC. In this work, they extracted PPGs from source speech using an acoustic model, and then trained a DNN to produce the converted speech from source PPGs. Miyoshi *et al.* [17] extended the previous PPG-based method with a sequence-to-sequence model that converted the context posterior probabilities, which improved the speaker identity of the converted speech. Zhou *et al.* [20] adopted bilingual PPG for cross-lingual voice conversion. Liu *et al.* [45], Lu *et al.* [47], and Mohammadi *et al.* [46] extended the one-to-one PPG-based VC framework for many-to-many VC by conditioning on a speaker embedding.

Two previous studies [49, 159] explored the use of adversarial learning to disentangle linguistic and speaker representations in VC. Huang *et al.* [159] used a pre-trained speaker classifier in a VAE to reduce speaker information from the linguistic representations. Zhang *et al.* [49] achieved the same purpose using AEs by explicitly enforcing the distribution of the hidden representation from each speaker to be identical. **Our proposed method differs from these prior approaches in several aspects**. First, our adversarial learning algorithm has two advantages. Huang *et al.* [159] pre-trained the classifier and froze its weights during the training of the VC model. In contrast, our proposed method does not require the pre-training of the adversarial speaker classifier. Zhang *et al.* [49] used an explicit loss function for adversarial learning. In contrast, the speaker-independent hidden representation in our proposed method is implicitly learned through the minimax optimization. Second, these previous approaches have only been evaluated for *standard* conditions. In contrast, our study considers both *standard* and *zero-shot* conditions, the latter being appealing for real-world applications since it requires little data from the target speaker.

4.4. Methods

As illustrated in Figure 4.1, our proposed VC system consists of four modules (highlighted in blue): a speaker-independent acoustic model to extract PPGs, a speaker recognition model to extract d-vectors as the speaker embeddings, a PPG2speech

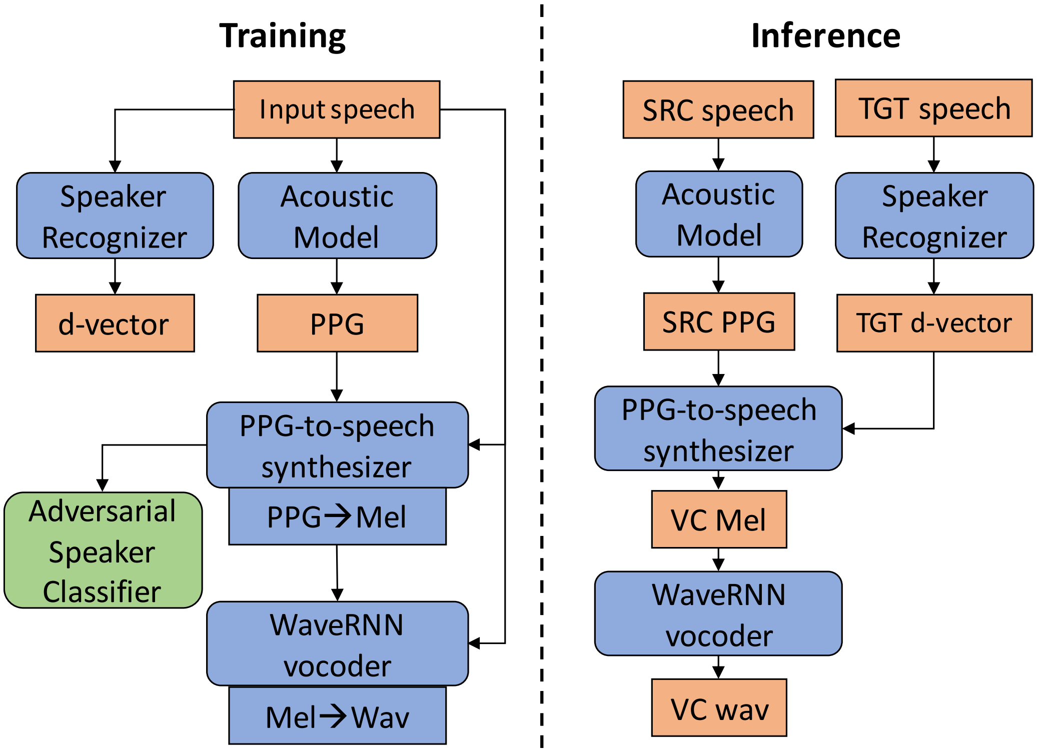


Figure 4.1: Overall workflow of the proposed non-parallel many-to-many VC system.

synthesizer to convert PPG to Mel-spectrograms, and a final neural vocoder to generate a speech waveform from the Mel-spectrogram. First, we introduce a state-of-the-art PPG2speech synthesizer based on Tacotron2 [88] as a baseline system. Then, we describe the proposed adversarial learning approach.

4.4.1. Baseline method: PPG2speech synthesizer

Our system is based on the text-to-speech Tacotron2 model, which uses a seq2seq model to convert a text embedding sequence to a Mel-spectrogram. Tacotron2 has an encoder-decoder architecture. The overall framework of our PPG2speech synthesizer is shown in Figure 4.2. Given a non-parallel corpus containing multiple speakers, the inputs to the network are pairs of PPGs ($\boldsymbol{x} \in \mathbb{R}^{T \times D}$) and the corresponding speaker embeddings ($\boldsymbol{s} \in \mathbb{R}^{M}$), where T is the length of the sequence, D is the dimensionality of the PPG, and M is the dimensionality of speaker embedding. During training, a PPG sequence $\boldsymbol{x}$ is first fed to the encoder $\boldsymbol{E}$,

$$\boldsymbol{z} = \boldsymbol{E}(\boldsymbol{x}; \boldsymbol{\theta_e}) \tag{4.1}$$

where $\boldsymbol{z}$ is the resulting hidden representation and $\boldsymbol{\theta_e}$ are the encoder parameters. Then, the hidden representation $\boldsymbol{z}$ and the speaker embedding $\boldsymbol{s}$ are concatenated and fed to an autoregressive attention-decoder network (along with the post-net) $\boldsymbol{D}$ with parameters $\boldsymbol{\theta_d}$, to produce the Mel-spectrogram $\widehat{\boldsymbol{y}}_{Mel}$,

$$\widehat{\boldsymbol{y}}_{Mel} = \boldsymbol{D}([\boldsymbol{z}, \boldsymbol{s}]; \boldsymbol{\theta_d}) \tag{4.2}$$

The speaker embedding we used here is a d-vector [160], which can be applied to either inset or unseen speakers. At the same time, the network also predicts if the generating process should stop, i.e., a stop token $\widehat{\boldsymbol{y}}_{\boldsymbol{stop}}$. The model is optimized by minimizing the loss:

$$\begin{aligned} L_{VC}(\boldsymbol{\theta_e}, \boldsymbol{\theta_d}) = \alpha \| \widehat{\boldsymbol{y}}_{Mel} - \boldsymbol{y}_{Mel} \|_2^2 \\ + \beta \mathrm{CE}\left(\widehat{\boldsymbol{y}}_{stop}, \boldsymbol{y}_{stop}\right) \end{aligned} \tag{4.3}$$

where $\boldsymbol{y}_{Mel}$ is the ground-truth Mel-spectrogram; $\boldsymbol{y}_{stop}$ is the ground truth stop token values; $\mathrm{CE}(\cdot)$ is the cross-entropy loss; α, β are the weights for each term to control the relative importance. The detailed architecture of the encoder and decoder will be described in the following subsections, and the hyperparameters of each module are shown in Table 4.1.

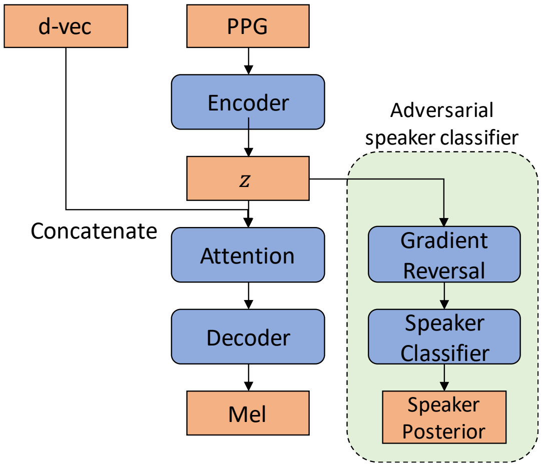


Figure 4.2: PPG2speech synthesizer with adversarial speaker classifier. *z* denotes the hidden representation produced by the encoder. The adversarial speaker classifier is only used during training.

Table 4.1: Hyper-parameters of the proposed seq2seq FAC model.

Block	***Component***	***Parameters***
Inputs	*PPG*	40-dim
	Speaker d-vector	256-dim
Encoder	*Convolution layers*	Three convolution layers Convolution kernel size: 5×1 Stride: 1×1 Output-dim: 256
Attention	*Attention layer*	Attention-dim: 128 Attention convolution filters: 32 Attention kernel size: 31
Decoder	*PreNet*	Two fully-connected layers each has 256 ReLU units, 0.5 dropout probability Output-dim: 256
	LSTM	Two LSTM layers 1,024 cells in each direction 0.1 dropout probability Output-dim: 512
	PostNet	Five 1-D convolution layers Convolution kernel size: 5 Output-dim: 80

4.4.1.1. Encoder

The encoder network in original Tacotron is consists of three convolution layers and one Bidirectional Long Short Term Memory (LSTM) layer, which takes a text embedding as the input and produces a hidden representation. In our case (voice conversion), the inputs of the PPG2speech synthesizer are PPGs instead of text embeddings. The PPG sequences are usually significantly longer than text embedding sequences. To capture the high-level phonetic and contextual information in an input PPG sequence, we replace the LSTM layer in the encoder with two pyramidal-LSTM (pLSTM) layers [161]. Each pLSTM reduces the time resolution by a factor of two, and therefore our encoder produces four times shorter hidden representation sequences compared with the input sequences.

Taking a PPG sequence $\boldsymbol{x}$ as the input, the convolution layers first captures the local temporal context information. Each of these layers has 32 kernels with a shape of 3×3 in time×frequency and a stride of 1×1, followed by ReLU activations and batch normalization [162]. The pLSTM layers capture high-level frequency-wise feature and long-term temporal context information, each of which has 256 cells in each direction, followed by ReLU activations and batch normalization. The resulting hidden representation sequence $\boldsymbol{z}$ is four times shorter than the input PPG sequence with a dimensionality of 512.

4.4.1.2. Decoder

The decoder is an autoregressive recurrent network with a location-sensitive attention mechanism [163], as shown in Figure 4.3: The diagram of the decoder network.Figure 4.3. The decoder consumes the concatenation of the hidden representation $\boldsymbol{z}$ and speaker embedding $\boldsymbol{s}$, generating an 80-dimensional Mel-spectrogram $\boldsymbol{y_{Mel}}$ as an estimation of the target speech. During each time step t, the estimated Mel-spectrogram from the previous step $\boldsymbol{y}_{Mel}^{t-1}$ is first input to a two-layer pre-net of 256 neurons and produces a query vector:

$$\boldsymbol{q}^t = \mathrm{PreNet}\;\; \boldsymbol{y}_{Mel}^{t-1} \tag{4.4}$$

Following this, the location-sensitive attention mechanism produces a context vector $\boldsymbol{c}^t$ based on the query vector $\boldsymbol{q}^t$, the concatenation between the hidden representation and speaker embedding $[\boldsymbol{z}, \boldsymbol{s}]$, and the attention context vector from the previous time step $\boldsymbol{c}^{t-1}$:

$$\boldsymbol{c}^t = \mathrm{Attention}\;\; \boldsymbol{q}^t, [\boldsymbol{z}, \boldsymbol{s}], \boldsymbol{c}^{t-1} \tag{4.5}$$

Then, the query vector $\boldsymbol{q}^t$ is concatenated with the attention context vector $\boldsymbol{c}^t$ and passed to two-layer of LSTMs with 256 cells, the output of which is concatenated with the attention context vector $\boldsymbol{c}^t$ again and fed to a linear layer, predicting the 80-dimensional Mel-spectrogram:

$$\boldsymbol{y}_{pre}^t = \mathrm{Linear}\;\; \mathrm{LSTM}\;\; \boldsymbol{q}^t, \boldsymbol{c}^t \;\;, \boldsymbol{c}^t \tag{4.6}$$

At the same time, another linear layer predicts a stop token $\boldsymbol{y}_{stop}$ to determine if the decoding process should stop during inference:

$$\boldsymbol{y}_{stop} = \mathrm{Linear}\;\; \mathrm{LSTM}\;\; \boldsymbol{q}^t, \boldsymbol{c}^t \;\;, \boldsymbol{c}^t \tag{4.7}$$

Finally, the Mel-spectrogram is input to a post-net with five convolution layers, which predicts the residual and improves the synthesis by adding the residual. Each of the convolution layers has 512 kernels with 5×1 shape and 1×1 stride, followed by *tanh*

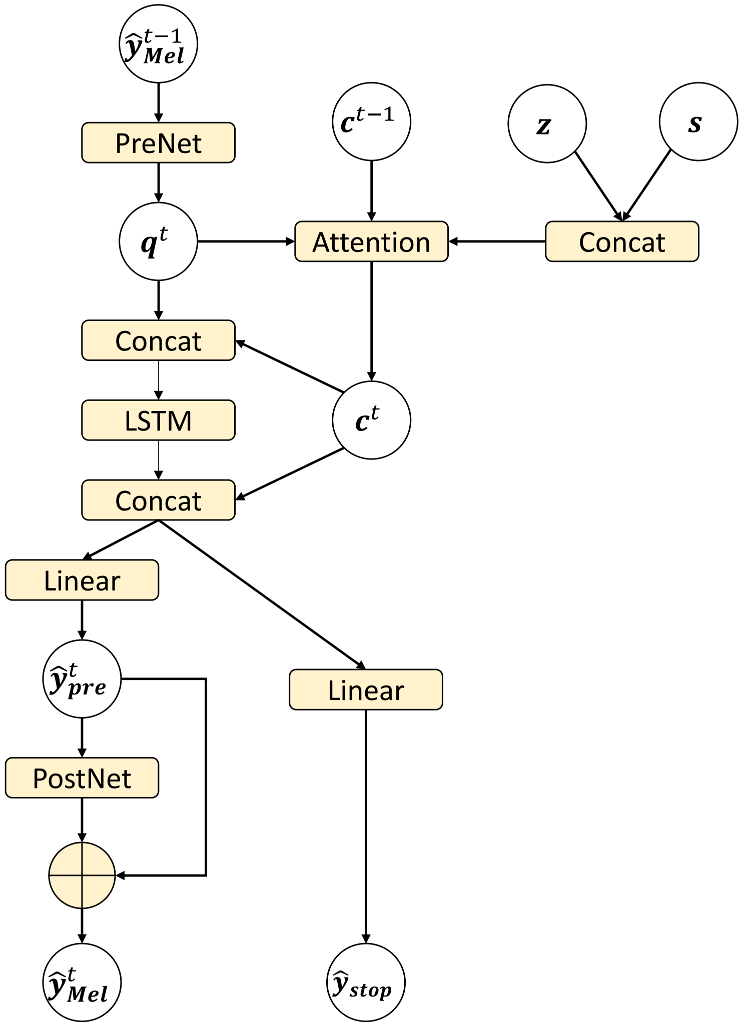


Figure 4.3: The diagram of the decoder network.

activation and batch normalization, which is added back to the original prediction to form the final prediction:

$$\boldsymbol{y}_{Mel}^t = \boldsymbol{y}_{pre}^t + \mathrm{PostNet}(\boldsymbol{y}_{pre}^t) \qquad 4.8$$

4.4.2. Proposed method: Adversarial speaker classifier

As we have noted, the PPG2speech synthesizer ignores the fact that PPGs carry speaker individualities such as accent, intonation, and speaking rate. As a result, the converted speech can still resemble the source speaker. The proposed adversarial speaker classifier, shown in Figure 4.2, is designed to address this issue. The classifier $\boldsymbol{C}$ takes the encoder output $\boldsymbol{z}$ as input and generates a probability for each speaker:

$$\boldsymbol{p} = \boldsymbol{C}\ \ \boldsymbol{z};\boldsymbol{\theta_c}\ \ = \boldsymbol{C}\ \ \boldsymbol{E}\ \ \boldsymbol{x};\boldsymbol{\theta_e}\ \ ;\boldsymbol{\theta_c} \tag*{4.4}$$

where $\boldsymbol{\theta_c}$ denote model parameters. The encoder $\boldsymbol{E}$ and adversarial speaker classifier $\boldsymbol{C}$ are jointly trained with an adversarial loss:

$$L_{ADV}\ \ \boldsymbol{\theta_e},\boldsymbol{\theta_c}\ \ = -\sum_{k=1}^{K}\mathbb{I}(y_{speaker} == k)\log p_k \tag*{4.5}$$

where $\mathbb{I}(\cdot)$ is the indicator function, K is the number of speakers, $y_{speaker}$ is the speaker who produced $\boldsymbol{x}$, and p_k is the probability of speaker k. During training, parameters $\boldsymbol{\theta_c}$ are optimized to minimize the adversarial loss to better identify the corresponding speaker, whereas parameters $\boldsymbol{\theta_e}$ are optimized to maximize the adversarial loss (i.e., to fool the classifier.) This minimax competition will finally converge when the output of the encoder is sufficiently speaker-independent such that the classifier is not able to identify the speaker.

The VC model is trained jointly with the adversarial speaker classifier in a multi-task learning fashion,

$$L\ \ \boldsymbol{\theta_e},\boldsymbol{\theta_d},\boldsymbol{\theta_c}\ \ = L_{VC}\ \ \boldsymbol{\theta}_e,\boldsymbol{\theta_d}\ \ - \lambda L_{ADV}\ \ \boldsymbol{\theta_e},\boldsymbol{\theta_c} \tag*{4.6}$$

where λ control the relative importance of L_{AVD}. Parameters $\boldsymbol{\theta_e},\boldsymbol{\theta_d},\boldsymbol{\theta_c}$ are optimized such that,

$$\boldsymbol{\theta_e},\boldsymbol{\theta_d} = \operatorname{argmin} L\ \ \boldsymbol{\theta_e},\boldsymbol{\theta_d},\boldsymbol{\theta_c} \tag*{4.7}$$

$$\boldsymbol{\theta_c} = \operatorname{argmax} L\ \ \boldsymbol{\theta_e},\boldsymbol{\theta_d},\boldsymbol{\theta_c} \tag*{4.8}$$

and they can be updated though back-propagation using stochastic gradient descent (SGD) as,

$$\boldsymbol{\theta_e} \leftarrow \boldsymbol{\theta_e} - \mu \left(\frac{\partial L_{VC}}{\partial \boldsymbol{\theta_e}} - \lambda \frac{\partial L_{ADV}}{\partial \boldsymbol{\theta_e}} \right) \tag{4.9}$$

$$\boldsymbol{\theta_d} \leftarrow \boldsymbol{\theta_d} - \mu \left(\frac{\partial L_{VC}}{\partial \boldsymbol{\theta_d}} \right) \tag{4.10}$$

$$\boldsymbol{\theta_c} \leftarrow \boldsymbol{\theta_c} - \mu \left(\frac{\partial L_{ADV}}{\partial \boldsymbol{\theta_c}} \right) \tag{4.11}$$

where μ is the learning rate. The negative coefficient $-\lambda$ in eq. (3.9) reversed the gradient back-propagated from the adversarial speaker classifier. The gradient reversal maximizes L_{AVD} for $\boldsymbol{\theta_e}$ and makes the encoder spoof the classifier, which is key to the optimization. In practice, we use the gradient reversal layer introduced in [164, 165]. During forward-propagation, it operates as an identity transform, and during back-propagation it multiplies the gradient by $-\lambda$.

4.5. Experimental setup

4.5.1. Acoustic model, speaker recognition model, and neural vocoder

We used a fully-connected DNN [166] as the acoustic model, which outputs 5,816 senones. We used the implementation in Kaldi [167] and trained the acoustic model on the Librispeech corpus [168]. We implemented the speaker recognition model proposed in [160] to produce a 256 dimensional d-vectors and trained it on the VoxCeleb2 dataset [169]. We used a universal WaveRNN [170] as the neural vocoder for all the testing speakers. The vocoder was trained on the VCTK training set (see below). Both the speaker recognition model and the neural vocoder were implemented in PyTorch [171].

4.5.2. PPG2speech synthesizer

We trained and evaluated the proposed VC system on the CSTR VCTK Corpus [158], which contains utterances from 109 English speakers with several accents (e.g., British, American, Scottish, Irish, Indian). For each speaker, there are on average 300 utterances, a subset of which have the same linguistic contents across all speakers. In our experiments, we divided the corpus into three subsets: a training set, a standard (test speakers were seen in training) test set, and a zero-shot set (test speakers were unseen in training) test set. The training set consists of 105 speakers. Among these speakers, we selected four speakers for standard testing (p227, p228, p240, p256). We used the first 20 utterances of these speakers as the standard test set, and excluded them from the training set. The zero -shot test set consists of the first 20 utterances of 4 speakers (p225, p226, p229, p232) that did not appear during training. All the test speakers had a British accent. For the standard test set, we considered four VC directions: p227 to p228 (M-F), p228 to p240 (F-F), p240 to p256 (F-M), and p256 to p227 (M-M). For the zero -shot test set, we also considered four VC directions: p225 to p226 (F-M), p226 to p232 (M-M), p232 to p229 (M-F), and p229 to p225 (F-F).

For each utterance, we down-sampled the waveform from 48kHz to 16kHz to match the sampling rate of other modules, and then extracted an 80-dim Mel-spectrogram with a 50ms window and 12.5ms shift. Following the same frame shift, we extracted the PPG (collapsed into a 40-dim mono-phone PPG from the 5,816-dim senone PPG) and the d-vector (256-dim) for each utterance using the acoustic model and speaker recognition model, respectively.

We implemented the VC models using TensorFlow[7] [172] and trained on a single NVIDIA V100 GPU. Hyperparameters α, β were set to $1.0,\ 0.005$ empirically. Following [164], we gradually changed λ in adversarial speaker classifier from 0 to 1 during the training process as:

$$\lambda_p = \frac{2}{1 + \exp\ -10 \cdot p} - 1 \qquad 4.12$$

where p is the percentage of the training process. We used a batch size of 64 and an Adam Optimizer with a learning rate of 10^{-4}. The model converged after 60,000 steps, and the entire training time was around 30 hours.

4.6. Experiments

We conducted both objective and subjective experiments under standard (the test speakers were seen) and zero-shot conditions (the test speakers were unseen). For objective evaluation, we used the Mel-Cepstral Distortion (MCD) [173] between VC and the ground-truth target utterances, which is defined as,

$$\mathrm{MCD[dB]} = \frac{10}{\ln 10} \sqrt{2 \sum_{t=1}^{T} \sum_{d=1}^{24} (y_d^t - \hat{y}_d^t)^2} \qquad (4.13)$$

where $\hat{y}_d^t$ and y_d^t are the d-th Mel-Cepstral coefficient of the t-th frame of VC speech and

[7] Audio samples and source code are available at https://github.com/shaojinding/Adversarial-Many-to-Many-VC.

the time-aligned ground-truth target speech, respectively. Since computing MCD requires the ground-truth target speech, we selected a subset of 19 utterances that have the same linguistic content. For subjective evaluation, we conducted two listening tests on Amazon Mechanical Turk. In the first test, we asked listeners to rate the acoustic quality using a 5-point (1-bad, 5-excellent) Mean Opinion Score (MOS). In the second test, we asked listeners to rate the similarity between pairs of utterances using a Voice Similarity Score (VSS; -7-definitely different speakers, +7 definitely the same speaker) [155]. All participants were required to pass a pre-test that asked them to identify different regional accents in the United States. Additionally, in each listening test, we used 12 calibration utterances to detect if participants were cheating. We excluded ratings of the calibration utterances from the data analysis.

4.6.1. Standard testing

For standard testing, we compared the proposed adversarial-learning approach (denoted as Proposed) against the baseline PPG2speech system in Section 4.4.1 (PPG2speech). We did not compare it to other zero-shot methods, as our PPG2speech baseline shares the same spirit as previous methods but with an advanced network structure. To ensure a fair comparison, we kept the encoder and decoder architectures identical to the proposed approach.

Results from the objective and subjective evaluations are summarized in Table 4.2. In objective evaluations, the proposed method achieved a statistically significant lower MCD (8.37) than the baseline (8.47, $p = 0.01$), suggesting the synthesis from the proposed approach is closer to the target speech.

For the VSS test, 17 participants rated 108 utterance pairs: 32 pairs (16 VC-SRC pairs, 16 VC-TGT pairs) for each of the three systems, and 12 calibration utterances[8]. Each pair consisted of a VC utterance and an utterance randomly selected from either the source (i.e., VC-SRC pair) or the target speaker (VC-TGT pair). For each utterance pair, participants were required to decide whether the two utterances were from the same speaker and then rate their confidence in the decision on a 7-point scale. The VSS was computed by collapsing the above two fields into a 14-point scale: -7 (definitely different speakers) to +7 (definitely the same speaker). As shown in the in Table 4.2, the proposed approach received a VSS rating of -6.20 on VC-SRC pairs, and 5.02 on VC-TGT pairs, which indicated that listeners were confident that VC syntheses and source speech were produced by different speakers, and that syntheses and target speech were produced by the same speaker, respectively. These scores were significantly better than those for the baseline: -5.62 VC-SRC, 4.25 VC-TGT; $p \ll 0.001$ in both cases, indicating that the synthesis from the proposed approach has an identity that is much closer to the target speech than the baseline.

[8] A VC-SRC pair consists of a VC utterance and an utterance randomly selected from the source speaker (SRC), whereas a VC-TGT pair consists of a VC utterance and an utterance randomly selected from the target speaker (TGT).

Table 4.2: Objective (MCD, lower the better) and subjective (MOS and VSS, higher the better) results under *standard* setting. All the results are shown with 95% confidence interval.

	MCD	VSS		MOS
		VC-SRC	VC-TGT	
PPG2speech	8.47±0.07	-5.62±0.09	4.25±0.12	3.77±0.06
Proposed	**8.37±0.07**	**-6.20±0.06**	**5.02±0.10**	**3.86±0.05**

In MOS test, 19 participants rated 72 utterances from the three VC systems: 20 utterances per system, and 12 calibration utterances. These utterances shared the same linguistic content across all the systems to ensure a fair comparison. For each utterance, participants were required to rate its acoustic quality from 1-bad to 5-excellent. As shown in in Table 4.2, participants rate the proposed approach to have a 3.86 MOS, which is significantly higher than the baseline (3.77, $p = 0.03$), boosting the synthesis acoustic quality. In summary, the proposed system shows significant improvements in terms of all three objective and subjective measurements compared to the baseline approach, which p[roves the effectiveness of adversarial training scheme.

4.6.2. Zero-shot testing

In the second set of experiments, we evaluated our approach under zero-shot condition. The zero-shot condition has attracted more attention from real-world applications in recent years since it requires little data from each target speaker, which saves the users' time and improves their experiences. For zero-shot testing, we also compared the proposed approach against the PPG2speech baseline.

Results from the objective and subjective evaluation tests are shown in Table 4.3. In the MCD test, the proposed method (9.31) marginally outperforms the PPG2speech

Table 4.3: Objective (MCD, lower the better) and subjective (MOS and VSS, higher the better) results under *zero-shot* setting. All the results are shown with 95% confidence interval.

	MCD	VSS		MOS
		VC-SRC	VC-TGT	
PPG2speech	9.38±0.09	-5.53±0.11	4.17±0.21	3.61±0.06
Proposed	**9.31±0.08**	**-6.12±0.10**	**4.80±0.20**	**3.77±0.06**

baseline (9.38, $p = 0.4$). In the VSS test, 18 participants rated 76 utterance pairs: 32 pairs (16 VC-SRC pairs and 16 VC-TGT pairs) for each of the two systems, and 12 calibration utterances. Each pair consisted of a VC utterance and an utterance randomly selected from either the source (i.e., VC-SRC pair) or the target speaker (VC-TGT pair). As shown in Table 4.3, participants were quite confident that the syntheses from the proposed method and the source speech were produced by different speakers (-6.12 VC-SRC), and that the syntheses and the target speech were produced by the same speaker (4.80 VC-TGT). This result also surpasses the PPG2speech baseline (-5.53 VC-SRC, 4.17 VC-TGT; $p \ll 0.001$ in all cases) with statistical significance, suggesting a superior voice identity of the proposed approach.

In the MOS test, 19 participants rated 52 utterances from the two VC systems: 20 utterances per system, and 12 calibration utterances. As shown in Table 4.3, participants rated the proposed approach to have a 3.77 MOS, which is significantly higher than the ratings of the baseline (3.61, $p \ll 0.001$).

To summarize, the proposed approach significantly outperforms the baseline regarding all of the measurements, similar to our observations under standard testing condition. When comparing the performance of the proposed approach under the two conditions, we found that the performance under zero-shot condition is still not as good as that under standard condition. A possible explanation is that zero-shot condition challenging setting since the test speakers are unseen during training, yet the transferability of the speaker embedding is still limited to new speakers.

4.7. Conclusions and future work

We have proposed an adversarial learning approach to improve speaker identity in non-parallel many-to-many voice conversion. During training, the encoder output is consumed by an adversarial speaker classifier, which is optimized to identify the corresponding speaker. At the same time, the encoder is optimized to fool the adversarial speaker classifier, and therefore, it can produce more speaker-independent linguistic representations. We conducted both objective and subjective experiments under standard and zero-shot conditions. Results indicate that the proposed method consistently improves the speaker identity and acoustic quality of VC syntheses over the baseline under both conditions.

Currently, there is still a gap between the performances under the two conditions. As a result, one potential future work is to improve the performance under zero-shot condition, which is probably due to the limited transferability of the speaker embedding. To improve the transferability of the speaker embedding, we can increase the number of training speakers, so that the model can better capture the speaker space. At the same time, overfitting is another issue that can reduce the transferability to unseen speakers of the model. To address this issue, we can explore the use of relative techniques such as dropouts [174] and batch normalization [162] in speaker recognition models.

5. ZERO-SHOT FAC BASED ON SEQ2SEQ MODEL: FOREIGN ACCENT CONVERSION TO ARBITRARY NON-NATIVE SPEAKERS USING ZERO-SHOT LEARNING [*]

5.1. Overview

Foreign accent conversion (FAC) aims to create a new voice that has the *voice identity* of a given second-language (L2) speaker but with a native (L1) *accent*. Previous FAC approaches usually require training a separate model for each L2 speaker and, more importantly, generally require considerable speech data from each L2 speaker for training. To address these limitations, we propose an approach that can generate accent-converted speech for arbitrary L2 speakers unseen during training. In the proposed approach, we first train a speaker-independent acoustic model on L1 corpora to extract bottleneck-features that represent the linguistic content of utterances. Then, we develop a speaker encoder and an accent encoder to generate embedding vectors for the desired voice identity (L2 speaker's) and accent (L1 accent), respectively. Lastly, we use a sequence-to-sequence model to transform L1 bottleneck-features to Mel-spectrograms, conditioned on the L2 speaker embedding and the L1 accent embedding. We conducted experiments on the L2-ARCTIC corpus under two testing conditions: the *standard* FAC

[*] This chapter is being submitted to the Computer Speech and Language.

setting where test L2 speakers were *seen* during training, and a *zero-shot* FAC setting where test L2 speakers were *unseen* during training. The proposed system achieves over 27% relative improvement in accentedness ratings compared to two state-of-the-art FAC systems in the standard FAC setting. More importantly, our results show that the proposed approach generalizes to the zero-shot FAC setting with no performance loss. Therefore, in practical use scenarios (e.g., computer-assisted pronunciation training software), our proposed approach can effectively avoid the need to adapt or retrain the model, which significantly reduces computations and the users' waiting time.

5.2. Introduction

Foreign accent conversion (FAC) [4] aims to create a new voice that has the *voice identity* of a given L2 speaker and the *accent (or pronunciation patterns)* of an L1 speaker. In pronunciation training, FAC can serve as a "golden speaker" for the L2 speaker to practice with: their own voice, but with a native accent [4-7]. FAC also finds applications in movie dubbing [175], personalized text-to-speech (TTS) synthesis [176, 177], and improving speech recognition performance [178]. A variety of techniques have been proposed to perform FAC, including voice morphing [4, 21, 22], frame pairing [23, 24], articulatory synthesis [25, 26], and sequence-to-sequence (seq2seq) modeling [27, 28]. However, previous FAC approaches have two major limitations. First, they operate in a one-to-one fashion, which requires training a separate model for each pair of L1 and L2 speakers. Second, they need a considerable amount of speech data (~1,000 utterances) for each L2 speaker. Thus, when using these conventional FAC methods in real-world applications such as pronunciation training, L2 learners need to record a large number of utterances and then wait for a dedicated model to be trained, which can be tedious and demotivating.

To address this issue, we propose a zero-shot learning [58] approach to FAC that can synthesize speech for arbitrary L2 speakers who were unseen during training**.** Our system consists of four independent models: (1) a speaker-independent acoustic model that captures the linguistic content of an L1 utterance as a sequence of *bottleneck feature vectors*, (2) a speaker encoder that captures the voice identity of the L2 speaker, denoted as a *speaker embedding*, (3) an accent encoder that captures the desired L1 pronunciation patterns, denoted as an *accent embedding*, and (4) a sequence to sequence (seq2seq) model that generates a Mel-spectrogram from the sequence of bottleneck features, conditioned on the desired speaker and accent embeddings. These components can be trained independently, at which point the system can generate accent conversions to arbitrary L2 speakers given a few seconds of audio (i.e., enough speech to compute a speaker embedding), without the need to have any model re-training or adaptation process.

To our knowledge, ours is the first work to apply zero-shot learning for the task of FAC. Though zero-shot learning has been used for voice conversion [11, 39] and voice cloning [89, 179], previous studies [19, 39, 43, 89, 179] have focused exclusively on manipulating voice identity, ignoring the speaker's accent, which holds important cues to speaker recognition [180] and speech perception [181-184]. Incorporating accent into the conversion process requires changes to the conventional encoder-decoder structure of seq2seq models for voice conversion. Our encoder takes a sequence of L1 bottleneck feature vectors as the input, and produces a hidden representation sequence. In a conventional voice conversion system [39-42, 45-47, 185], this hidden representation sequence is then concatenated with the speaker embedding of the target speaker. In our case, however, the system also concatenates the accent embedding, which is treated as an additional independent and controllable factor during synthesis. The combined bottleneck/speaker/accent embedding is consumed by a decoder coupled with a location-sensitive attention mechanism [163]. During each decoding step, the decoder autoregressively predicts a Mel-spectrogram frame based on the output from the previous decoding step and a context vector produced by the attention mechanism. Finally, the output Mel-spectrogram is converted back into a waveform through either the Griffin-Lim algorithm [186] or a separately trained vocoder (e.g., WaveNet [187], WaveRNN [170]).

We thoroughly evaluated the proposed approach on the L2-ARCTIC corpus [188]. First, we visualized the speaker and accent embedding distributions for the accent-converted speech and natural speech, and the results show that our FAC syntheses can successfully capture the L2 voice identity along with an L1 accent. Second, we conducted a series of listening tests under two different settings: (1) a *standard* FAC setting, where the test L2 speakers were available during training, and (2) a *zero-shot* FAC setting, which assumes that the test L2 speakers were not available during training. Our results show that the proposed system achieves 27% relative improvement in accentedness while retaining the acoustic quality and voice identity, compared to two state-of-the-art FAC systems in standard FAC settings,. In addition, the proposed system is proven to have no performance loss when we test it under zero-shot FAC setting.

The chapter is organized as follows. Section 5.3 reviews prior approaches to foreign accent conversion, many-to-many voice conversion, and sequence-to-sequence models. Section 5.4 describes the proposed foreign accent conversion method in detail. Section 5.5 provides the experimental setup, including the corpora and implementation details. Section 5.6 presents an analysis based on visualizations and two sets of subjective evaluations of the proposed method. We discuss the implications of the results in Section 5.7. Lastly, we conclude the findings of this work and point out potential future directions in Section 5.8.

5.3. Related work

5.3.1. Foreign accent conversion

The problem of foreign accent conversion was initially formulated by Felps et al. [4] as the means to provide implicit feedback in computer assisted pronunciation training. Early approaches to FAC [26, 52-54] involved building an articulatory synthesizer for the L2 speaker. The articulatory synthesizer was trained to map the speaker's articulatory trajectories (e.g., tongue and lip movements) into his or her acoustics features (e.g., Mel Cepstra) using GMMs [26], unit-selection models [52], and DNNs [53]. Once the synthesizer was built, it could be driven with articulatory trajectories from an L1 speaker to synthesize FAC speech. However, these approaches were impractical for pronunciation training since collecting articulatory data is expensive and requires specialized equipment[9]. As a result, later work on FAC has focused on *acoustic* methods, since they only require recording speech with a microphone. Previous acoustic methods can be grouped into two categories: frame-pairing methods [23, 24] and seq2seq methods [27, 28]. Frame-pairing methods first pair L1 and L2 speech frames based on their similarity, and then use a statistical model (e.g., a GMM) to convert from L1 frames to their corresponding L2 frames. Aryal and Gutierrez-Osuna [23] first proposed a technique to pair L1-L2 frames based on their acoustic similarity (in MFCC space), after applying vocal tract length normalization to reduce global differences between the L1 and L2 spectra. Following this, Zhao *et al.* [24] argued that the L1 and L2 frames should be paired based on their linguistic content, and consequently, they used Phonetic-PosteriorGram (PPG) similarity instead of MFCC similarity to pair acoustic frames. More recently, methods based on seq2seq models have been shown to significantly improve synthesis quality. In a previous study [27], we proposed a seq2seq PPG-to-Mel synthesizer for FAC. During training, the system learns a seq2seq model to convert PPGs to Mel-spectra extracted from utterances of an L2 speaker. During inference, the model is driven by PPGs extracted from a reference L1 utterance, which then produces FAC synthesis. In related work, Liu *et al.* [28] proposed a novel

[9] Articulatory measurements can be performed via electromagnetic articulography [52], ultrasound imaging [55], palatography [56], and more recently real-time MRI [57]

recognizer-synthesizer framework to remove the need for a reference L1 utterance. Their system trained a speaker recognizer, a multi-speaker text-to-speech (TTS) model, and an accent-sensitive automatic speech recognition (ASR) system. During inference, they feed L2 Mel-spectra to the ASR system with the corresponding accent, and then feed the output of the ASR system and the L2 speaker embedding to the multi-speaker TTS model to generate accent-converted utterances. These seq2seq model based FAC approaches can convert segmental and prosody features simultaneously, producing syntheses with higher speech naturalness and acoustic quality.

5.3.2. Many-to-many voice conversion

Foreign accent conversion is related to the more general problem of voice conversion (VC) [189, 190], which aims to synthesize a voice that has the linguistic content of an utterance from a source speaker and the voice identity of a target speaker. Traditional VC approaches use GMMs [11, 37], sparse representations [30, 157], and DNNs [12-20] to transform the spectra from a source speaker to that of the target speaker. These methods require training a separate model for each pair of source-target speakers. More recently, several studies have proposed many-to-many VC approaches based on Variational Autoencoders (VAE) [39-44] and the PPG-to-speech synthesizer [45-49]. Hsu *et al.* [39, 50] first proposed to use a VAE for many-to-many VC. Their VAE consisted of an encoder and a decoder. During training, the encoder learns a speaker-independent latent embedding from input speech signals, and the decoder reconstructs the input speech signals given the latent embedding and the corresponding speaker embedding. During inference, the speaker embedding is replaced with that of a target speaker to produce a VC synthesis. A number of subsequent studies have been conducted to improve performance through various techniques, such as auxiliary classifiers [43], WaveNet vocoder adaption [44], and discrete latent spaces [40, 51]. Other studies [45-47] have used a PPG-to-speech synthesizer approach to perform many-to-many VC. The PPG-to-speech synthesizer is a neural network that takes PPGs as an input, and predict spectra conditioned on the speaker embedding of the target speaker. Early many-to-many VC models used one-hot vectors as the speaker embedding due to its simplicity, but recent studies [41, 45-47] have used learned speaker embeddings (e.g., i-vector [33], d-vector [34]) to generalize to unseen speakers, which make it possible to perform VC in a zero-shot fashion.

5.3.3. Seq2seq models

Our seq2seq model was originally proposed by Sutskever *et al.* [86] for machine translation. The seq2seq model usually has an encoder-decoder architecture. The encoder learns a hidden representation sequence from an input sequence, and the decoder learns to autoregressively generate the output sequence given the hidden representation. To capture local contextual information and handle length mismatches between the input and output sequences, an attention mechanism is added between the encoder and the decoder. In recent years, there has been growing interest in applying seq2seq model to speech synthesis. Wang *et al.* [87] first proposed a seq2seq based TTS synthesizer (Tacotron), which significantly improved the acoustic quality of the syntheses over previous methods. Following this, Shen *et al.* [88] proposed Tacotron2, which further improved the acoustic quality of Tacotron by using a novel model architecture and a WaveNet vocoder. Jia *et al.* [89] extended Tacotron2 to multi-speaker TTS by conditioning a speaker embedding on the decoder. Seq2seq model has also been applied to voice conversion [17, 48, 49] and foreign accent conversion [27, 28], which significantly improved the performance on these tasks compared to conventional approaches.

5.4. Methods

The proposed FAC system consists of four modules: (1) a speaker-independent acoustic model that generates a linguistic representation of an utterance, (2) a speaker encoder that captures the voice identity of the desired speaker, (3) an accent encoder that captures the desired accent, and (4) a seq2seq model that consumes the previous three representations to synthesize Mel-spectrogram for an arbitrary L2 speaker.

The workflow for training our system is shown in Figure 5.1. The acoustic model, speaker encoder, and accent encoder are trained separately, and then are used as feature extractors for the seq2seq model. The seq2seq model is trained on a parallel corpus with multiple L1 and L2 speakers, capturing the voice characteristics of different speakers and accents. In what follows, we define a *"source"* speaker to be a selected canonical L1 speaker, and a *"target"* speaker to be any L1/L2 speaker. To train the seq2seq model, we pair the source speaker with each target speaker. Then, for each pair of speakers, we feed source utterances to the speaker-independent acoustic model to extract bottleneck features (BNFs), which we assume to capture only the linguistic content. Next, we feed an utterance from the target speaker to the speaker encoder and the accent encoder, which extract their speaker embedding and accent embedding,

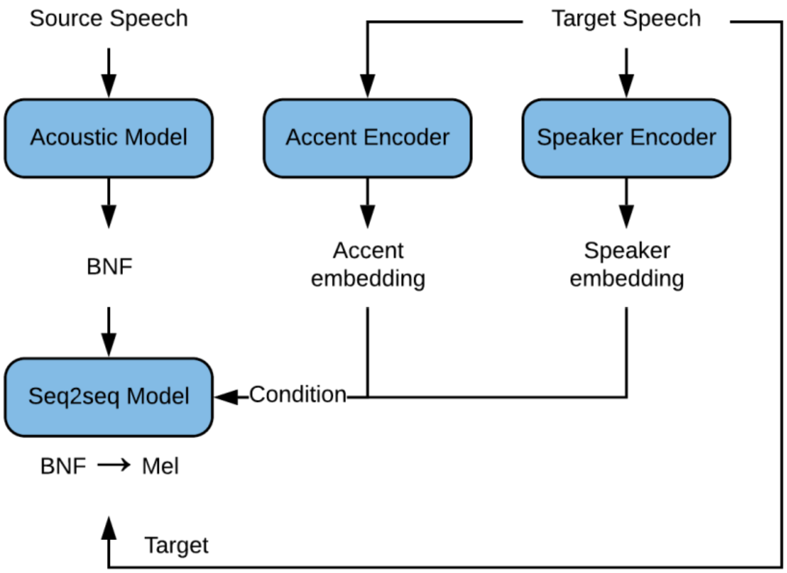


Figure 5.1: Overall training workflow of the proposed FAC approach. Source: a selected canonical L1 speaker, Target: any L1/L2 speaker, BNF: bottleneck feature. Each of the modules is trained independently.

respectively. Finally, we train the seq2seq model to convert the source BNFs to the target Mel-spectrogram, conditioning on the target speaker's speaker and accent embeddings.

The workflow during inference is illustrated in Figure 5.2. The system requires a source utterance from an L1 speaker and an utterance from the L2 speaker. First, we extract the BNFs and an accent embedding from the L1 utterance, and a speaker embedding from the L2 utterance. The L1 BNFs and L1 accent embedding encode the linguistic content and native accent, respectively, where the L2 speaker embedding encodes the voice identity of the L2 speaker. Then, we pass the L1 BNFs, L1 accent embedding, and L2 speaker embedding to the seq2seq model, which generates the accent-converted Mel-spectrogram. Finally, we train the seq2seq model to convert the

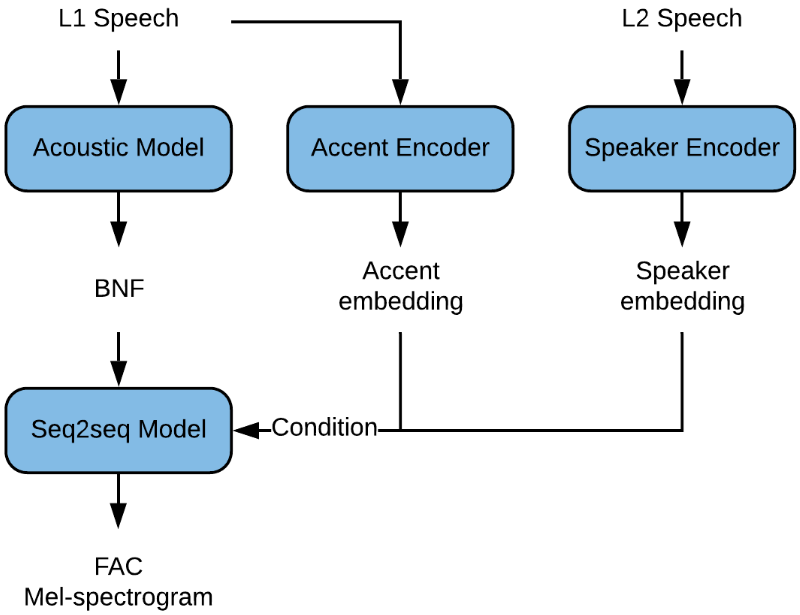


Figure 5.2: Overall inference workflow of the proposed FAC approach. L1: native, L2: non-native, BNF: bottleneck feature.

source BNFs to the target Mel-spectrogram, conditioning on the target speaker's speaker and accent embeddings.

5.4.1. Speaker-independent acoustic model

To capture the linguistic content of an utterance, we use the output of the last hidden layer of a speaker-independent acoustic model (AM) as BNFs, rather than the output of the final layer of the AM, which represent the PPGs (i.e., the probabilities of each senone/tri-phone). BNFs contain similar linguistic information as PPGs but have much lower dimensionality (e.g., Senone-PPG: 6,024 dimensions; BNF: 256 dimensions), which avoids the need to perform dimensionality reduction in the seq2seq model.

Our AM is based on a Factorized Time Delayed Neural Network (TDNN-F) [191, 192], a feed-forward neural network acting as a sequential classifier. Given an input acoustic feature vector (i.e., 40-dimensional MFCCs), the TDNN-F produces the probabilities of the vector belonging to each senone/triphone (6,024 senones). The TDNN-F takes time-delayed input frames as side inputs to its hidden layers to model long-term temporal dependencies, concatenated with a 100-dimensional i-vector [92] of the corresponding speaker[10]. Additionally, the TDNN-F uses factorized layers with semi-orthogonal constraints as hidden layers and dilated connections between hidden layers, which are more efficient during training and inference than recurrent layers due to their feed-forward nature [191]. The TDNN-F model is composed of five hidden layers. Each of the first four hidden layers has 1,280 neurons, followed by ReLU activation and batch normalization [162], whereas the last hidden layer has 256 neurons, corresponding to the dimensionality of the BNFs. We train the model through a supervised 6,024-way senone classification task. To promote that the AM produces speaker-independent BNFs, we train the model on speech data from several thousands of speakers (Librispeech corpus [168], 2,484 native English speakers; see Section 5.5.1).

[10] As noted by Peddinti et al. [192], this allows the model to capture both speaker and environment specific information, which is useful for neural network adaption

5.4.2. Speaker and accent encoders

We compute the voice identity and accent using two separate encoders. The speaker encoder is built upon a speaker recognition model trained to determine the identity of a speaker from an input utterance, whereas the accent encoder is based on an accent recognition model trained to recognize accent/dialect patterns (e.g., pronunciation and prosody). For this work, we use a convolutional neural network (CNN) based on ResNet-34 [193] for both the speaker encoder and the accent encoder. We use the same CNN architecture for both models, so we only describe the detailed workflow for speaker encoder; the training and inference workflows of the accent encoder can be derived similarly.

The architecture of the speaker encoder is shown in Figure 5.3. The model takes 300×257 in time×frequency magnitude spectrogram segments as inputs. The inputs are first fed to a convolution layer containing 64 7×7 kernels with 2×2 stride, followed by a 2×2 max-pooling layer. These layers decrease the spatial resolution of the feature maps, reducing model complexity and improving training speed. On top of them, there are 16 convolution residual blocks, which extract more abstract features. Each convolution block consists of two convolution layers with 3×3 kernels. The first convolution layer in each block has 2×2 stride to further decrease the spatial resolution of the feature maps. More importantly, each block has a skip connection as an alternative path to avoid gradient vanishing in a very deep model. The 16 convolution blocks have different numbers of kernels, as highlighted in different colors in Figure 5.3 (Purple: 64 kernels; Green: 128 kernels; Orange: 256 kernels; Blue: 512 kernels). Next, there is an

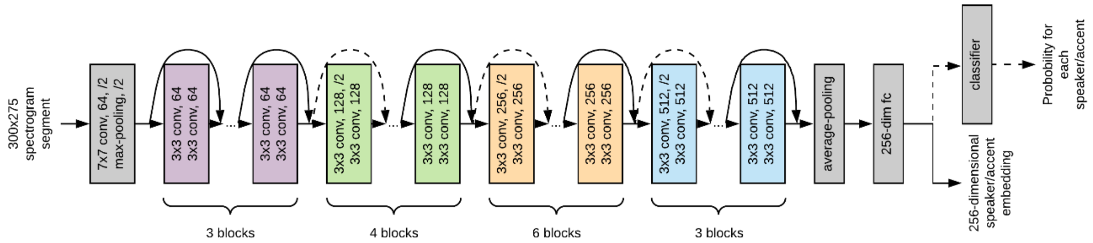


Figure 5.3: Speaker/accent encoder model architecture. The model is based on ResNet-34 [193]. Each convolution block is illustrated as the kernel size and channel numbers. "/2" means the layer divides the spatial resolution by 2.

average pooling layer that produces a 512-dimensional vector, followed by a 256-dimensional fully-connected layer. All the layers are followed by ReLU activations and batch-normalization.

The model is trained through a supervised speaker-classification task. During training, a classifier on top of the 256-dimensional fully-connected layer produces the probabilities that the segment belongs to each speaker. The network is then optimized by minimizing the cross-entropy loss between the prediction and the target speaker label. During inference, we discard the final classifier layer and directly use the 256-dimensional bottleneck feature as the segment-wise speaker embedding. To obtain utterance-level speaker embeddings for a speaker that does not appear during training, we divide each test utterance into 300-frame segments with a 150-frame overlap using a sliding window, and then we compute the average of these segment-wise embeddings as the utterance-level speaker embedding (i.e., d-vectors [34]).

5.4.3. Seq2seq foreign accent conversion model

Our proposed seq2seq model is inspired by the text-to-speech Tacotron2 model [88]. As shown in Figure 5.4, the seq2seq model has an encoder-decoder architecture.

During training, inputs to the network consist of triplets of (1) a sequence of BNFs from the source (L1) speaker $\boldsymbol{x} \in \mathbb{R}^{T_i \times D_{BNF}}$, (2) a speaker embedding of the target (L1 or L2) speaker $\boldsymbol{s} \in \mathbb{R}^{D_{speaker}}$ extracted from the speaker encoder, and (3) the accent embedding of the target speaker $\boldsymbol{a} \in \mathbb{R}^{D_{accent}}$ extracted from the accent encoder. T_i is the length of the sequence $\boldsymbol{x}$. D_{BNF} is the dimensionality of the BNFs (e.g., 256 in this work). $D_{speaker}$ and D_{sccent} are the dimensionalities of the speaker embedding ($\boldsymbol{s}$) and accent embedding ($\boldsymbol{a}$), respectively (both of them are 256 in this work). The ground-truth target of the model is a sequence of Mel-spectrogram frames $\boldsymbol{y} \in \mathbb{R}^{T_o \times D_{Mel}}$, where T_o is the length of the sequence and D_{Mel} is the number of Mel-filterbanks (e.g., 80 in this work). First, the encoder accepts a BNF sequence $\boldsymbol{x}$ and produces a hidden representation $\boldsymbol{h}$:

$$\boldsymbol{h} = \text{Encoder}\ \boldsymbol{x} \qquad 5.1$$

Then, to condition the decoder on the voice identity and the accent of the target speaker, we concatenate the target speaker's speaker embedding and accent embedding to the hidden representation:

$$\boldsymbol{h_{concat}} = [\boldsymbol{h}, \boldsymbol{s}, \boldsymbol{a}] \qquad 5.2$$

Finally, the decoder autoregressively predicts the Mel-spectrogram of the target speech using the attention context computed based on the concatenated hidden representation:

$$\boldsymbol{y}^t = \text{Decoder}\ \boldsymbol{y}^{t-1}, \boldsymbol{h_{concat}} \qquad 5.3$$

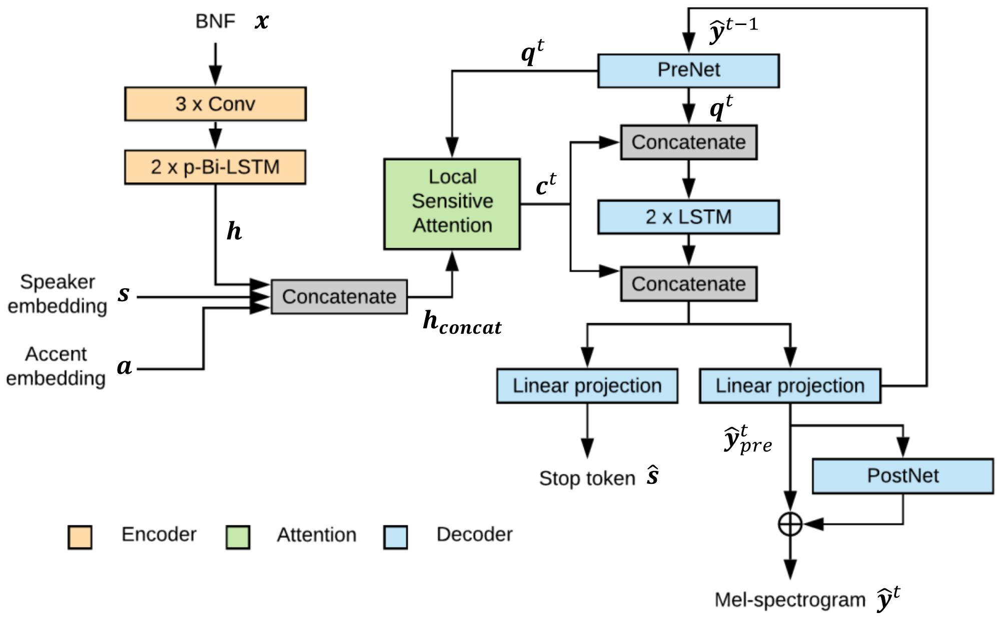

BNF $\boldsymbol{x}$
3 x Conv
2 x p-Bi-LSTM
$\boldsymbol{h}$
Speaker embedding $\boldsymbol{s}$
Accent embedding $\boldsymbol{a}$
Concatenate
$\boldsymbol{h}_{\boldsymbol{concat}}$
$\boldsymbol{q}^t$
Local Sensitive Attention
$\boldsymbol{c}^t$
$\hat{\boldsymbol{y}}^{t-1}$
PreNet
$\boldsymbol{q}^t$
Concatenate
2 x LSTM
Concatenate
Linear projection
Linear projection
Stop token $\hat{\boldsymbol{s}}$
$\hat{\boldsymbol{y}}^t_{pre}$
PostNet
Mel-spectrogram $\hat{\boldsymbol{y}}^t$
Encoder
Attention
Decoder

Figure 5.4: Proposed seq2seq FAC model.

During inference, the inputs to the network are triplets of (1) a sequence of BNFs from an L1 speaker $\boldsymbol{x}$, a speaker embedding of an L2 speaker $\boldsymbol{s}_{L2}$, and an accent embedding of an L1 speaker $\boldsymbol{a}_{L1}$. The network first produces a hidden representation $\boldsymbol{h}$, and then $\boldsymbol{h}$, $\boldsymbol{s}_{L2}$, and $\boldsymbol{a}_{L1}$ are concatenated and fed to the decoder to produce the predicted FAC Mel-spectrogram. We describe each component in the following subsections. The hyper-parameters of each component are summarized in Table 5.1.

Table 5.1: Hyper-parameters of the proposed seq2seq FAC model.

Block	***Component***	***Parameters***
Inputs	*BNF*	256-dim
	Speaker d-vector	256-dim
	Accent d-vector	256-dim

Table 5.1 Continued.

Block	*Component*	*Parameters*
Encoder	*Convolution layers*	Three convolution layers Convolution kernel size: 5×1 Stride: 1×1 Output-dim: 256
	p-Bi-LSTM layer	Two *p-Bi-LSTM layers* 256 cells in each direction Each layer reduces the time resolution by 2 Output-dim: 512
Attention	*Attention layer*	Attention-dim: 128 Attention convolution filters: 32 Attention kernel size: 31
Decoder	*PreNet*	Two fully-connected layers each has 256 ReLU units, 0.5 dropout probability Output-dim: 256
	LSTM	Two LSTM layers 1,024 cells in each direction 0.1 dropout probability Output-dim: 512
	PostNet	Five 1-D convolution layers Convolution kernel size: 5 Output-dim: 80

5.4.3.1. Encoder

The encoder converts a BNF sequence to a hidden representation sequence. The original text-to-speech Tacotron2 encoder contains three 1-dimensional convolution layers and one Bidirectional Long Short-Term Memory (Bi-LSTM) layer. However, in our case, the inputs of the seq2seq model are BNF sequences instead of text embeddings, which are usually significantly longer. To capture the high-level phonetic and contextual information in an input BNF sequence, we replace the LSTM layer in the encoder with two pyramidal Bidirectional LSTM (p-Bi-LSTM) layers [161]. Each p-Bi-LSTM reduces the time resolution by a factor of two, and therefore our encoder produces four times shorter hidden representation sequences compared with the input sequences. A convolution layer has 512 kernels with 5 × 1 shape in time×frequency and 1 × 1 stride, followed by ReLU activation and batch normalization. Each convolution kernel spans five BNF frames, which models the local context information. A p-Bi-LSTM layer has 256 cells in each direction, followed by ReLU activation and batch normalization, producing a 512-dimensional hidden representation sequence.

5.4.3.2. Decoder

The decoder is an autoregressive recurrent neural network coupled with a local sensitive attention mechanism. We use the same decoder architecture as in Tacotron2. The decoder accepts the concatenated hidden representation sequences as inputs, and produces an 80-dimensional Mel-spectrogram as the prediction of the L2 speech. During each decoding step, the predicted Mel-spectrogram frame from the previous step is first passed into a pre-net that has two 256-dimensional fully-connected layers with ReLU activations:

$$\boldsymbol{q}^t = \mathrm{PreNet}\ \boldsymbol{y}^{t-1} \tag*{5.4}$$

The pre-net acts as an information bottleneck, which is essential for learning attentions [88]. Next, the location-sensitive attention mechanism computes a 128-dimensional attention context vector $\boldsymbol{c}^t$ based on the pre-net output, the concatenated hidden representations, and the attention context from the previous step:

$$\boldsymbol{c}^t = \mathrm{Attention}\ \boldsymbol{q}^t, \boldsymbol{h_{concat}}, \boldsymbol{c}^{t-1} \tag*{5.5}$$

Following this, the pre-net output is concatenated with the context vector and fed to two unidirectional LSTM layers with 256 cells. Then, the output of the second LSTM layer is concatenated again with the context vector $\boldsymbol{c}^t$ and passed through an 80-unit linear layer to make a prediction of the 80-dimensional L2 Mel-spectrogram frame:

$$\boldsymbol{y}_{pre}^{t} = \mathrm{Linear}\ \mathrm{LSTM}\ \boldsymbol{q}^t, \boldsymbol{c}^t\ \ , \boldsymbol{c}^t \qquad 5.6$$

More importantly, the network also predicts if the generating process should stop at the current decoding step at the same time, i.e., a stop token $\boldsymbol{s} \in \mathbb{R}^T$. Finally, to incorporate the spectral residual and improve synthesis quality, the predicted Mel-spectrogram is passed through a post-net consisting of five convolution layers to predict the residual. Each of these layers has 512 kernels with 5×1 shape and 1×1 stride, followed by *tanh* activation and batch normalization. The residual is added back to the original prediction to form the final prediction:

$$\boldsymbol{y}_{post}^{t} = \boldsymbol{y}_{pre}^{t} + \mathrm{PostNet}(\boldsymbol{y}_{pre}^{t}) \qquad 5.7$$

The model is optimized by minimizing the L2 distance between the target Mel-spectrogram and the prediction before/after the post-net. We also jointly minimize an extra cross-entropy loss to learn the stop token for model inference.

$$\begin{aligned} L = &\|\boldsymbol{y}_{pre} - \boldsymbol{y}\|_2^2 + \ \|\boldsymbol{y}_{post} - \boldsymbol{y}\|_2^2 \\ &+ \lambda \mathrm{CrossEntropy}\ \boldsymbol{s}, \boldsymbol{s} \end{aligned} \qquad 5.8$$

where $\|\cdot\|_2^2$ is the Euclidean distance; $\boldsymbol{s}$ is the sequence of predicted stop tokens, and $\boldsymbol{s}$ is the sequence of target stop tokens; λ is the weight controlling the relative importance of the cross-entropy loss. Additionally, we use the teacher-forcing procedure during training by feeding in the correct output instead of the predicted output on the decoder side, which has been shown to improve the efficiency of the model training [194].

5.5. Experimental setup

5.5.1. Acoustic model

We trained the TDNN-F acoustic model using the Librispeech corpus [168], which consists of 960 hours of 16 kHz audiobook speech data produced by 2,484 native English speakers, the majority being American English. The training set consists of two "clean" subsets and a "noisy" subset. We used both sets in training to ensure that the BNF was speaker-independent. In addition, we used a subset (200 hours) of the training set to train the i-vector extractor. We implemented the training following the official "tdnn_1d" recipe of the TDNN-F model in Kaldi[11]. The trained model achieves 3.76% word error rate (WER) on Librispeech test-clean subset and 8.92% WER on the test-other subset.

5.5.2. Speaker encoder

We trained the speaker encoder using the VoxCeleb1 corpus [195], which contains 153,516 utterances of 16 kHz speech produced by 1,251 speakers. Specifically, we used the training set from the official identification split, which is comprised of

[11] https://github.com/kaldi-asr/kaldi/blob/master/egs/librispeech/s5/local/chain/tuning/run_tdnn_1d.sh

138,316 utterances (~300 hours) from these speakers. We extracted 257-dimensional magnitude spectrograms with a 25ms window and 10ms shift. We trained the model on a single NVIDIA Tesla V100 GPU with a batch size of 128. We used Adam Optimizer with an initial learning rate of 10^{-2}, which was annealed down to zero following a cosine schedule [196]. The trained model achieves 81.34% Top-1 accuracy and 94.49% Top-5 accuracy on the official VoxCeleb1 identification testing set.

5.5.3. Accent encoder

We trained the accent encoder using the Speech Accent Archive dataset [197], which consists of recordings of the “Please call Stella” paragraph [197] produced by speakers in 386 native and non-native English accents. For most of the accents, however, the number of speakers is limited, which may degrade the performance of the accent encoder. To address this issue, we selected a subset of accents where each accent has at least 30 speakers. The resulting subset we used during training has 18 accents, with an average of 107 speakers in each accent. The total length of the selected subset is around 16 hours. We randomly selected 90% utterances from each accent as the training set and used the remaining 10% utterances as the testing set. The audio waveforms in the original dataset have 8 kHz sampling rate. To match it with other modules, we resample them to 16 kHz. Other configurations were the same as that for speaker recognition. Our trained model achieves 79.36% Top-1 accuracy and 95.42% Top-5 accuracy on the testing set.

5.5.4. Seq2seq FAC model

To evaluate the proposed approach, we conducted experiments with the ARCTIC [125] and L2-ARCTIC corpora [188]. We used four native English speakers from ARCTIC (BDL, RMS, SLT, CLB) and all 24 non-native English speakers from L2-ARCTIC. For each speaker, we divided their utterances into three subsets: a training set of 1,032 utterances (~1 hour of speech), a validation set of 50 utterances, and a testing set of 50 utterances. During training, we set BDL as the source speaker and paired it with all 28 speakers, including himself. During inference, we used both BDL (male) and CLB (female; used as an unseen L1 speaker for the zero-shot FAC setting) as the native reference speakers, and we performed FAC on four L2 speakers whose first languages were different: NJS (Spanish, female), TXHC (Mandarin, male), YKWK (Korean, male), and ZHAA (Arabic, female).

The original L2-ARCTIC audio waveforms have a 44.1 kHz sampling rate, so we resampled them to 16 kHz to match the ARCTIC recordings. We extracted 80-dimensional Mel-spectrogram with a 25ms window and 10ms shift. Following the same frame shift, we extracted BNFs for each utterance using the acoustic model (Section 5.4.1). In addition, we extracted utterance-level speaker and accent d-vectors from the speaker encoder and accent encoder, respectively. We implemented the model using TensorFlow [172] and trained it on a single NVIDIA Tesla V100 GPU. The hyperparameter λ (eq. 8) in the loss function was set to 0.005 empirically. We set the batch size to 48, and we used an Adam Optimizer with an initial learning rate of 10^{-3}, which was then annealed down to 10^{-5} following exponential scheduling. The model converged after 200,000 steps, and the entire training time was around 100 hours. During model inference, we used a separately trained speaker-independent WaveRNN [170] vocoder to invert the Mel-spectrogram back to the time-domain waveform. We trained the WaveRNN model[12] on the Librispeech dataset. Audio samples from this work can be found at https://shaojinding.github.io/samples/fac-to-arbitrary-speaker/. We intend to open-source our code after this work has been peer-reviewed.

5.6. Results

Our experiments are comprised of a t-SNE [198] visualization and two sets of subjective evaluations. In the t-SNE visualization, we visualized the speaker and accent embedding distributions for the accent-converted speech and natural speech to qualitatively evaluate the voice identity and the accentedness of the FAC syntheses. In the first set of subjective evaluation, we tested the system when the test L1 and L2 speakers are seen during training (*standard* FAC setting) and compared it against two state-of-the-art FAC systems [27, 28]. We also tested whether our system could be used in the *reverse* direction, i.e., to impart a non-native accent to a native speaker's voice. In the second set of subjective evaluation, we explored the effectiveness of the proposed method when the test L1 and L2 speakers were unseen during training (*zero-shot* FAC

[12] We use the open-source implementation at https://github.com/fatchord/WaveRNN

setting). Also, we characterized the performance of the proposed method as a function of the number of available L2 test utterances during inference, which were used to extract the L2 speaker's voice identity footprint, and we compared it against a system that uses these utterances to finetune a pre-trained FAC system to provide more insights between the choices between zero-shot learning model and finetuning.

5.6.1. Visualization of speaker and accent embedding spaces

First, we visualized the speaker and accent embedding spaces to provide a qualitative and intuitive explanation of how our proposed system operates. For this purpose, we used t-distributed stochastic neighbor embedding (t-SNE) [198] to visualize the embeddings. We first visualized the speaker and accent embeddings of 20 FAC utterances for TXHC, a male Mandarin speaker. We used the system in the zero-shot condition when both L1 and L2 speakers were unseen (see Section 5.6.3.1) to generate the syntheses, since it acts as a performance lower-bound for all our systems, and it can also provide insights for zero-shot FAC. We also plotted the embeddings of natural speech from 10 L1 and L2 speakers as references (20 utterances for each speaker). Results are shown in Figure 5.5. We use colors and shapes to represent speaker and accent, respectively (e.g., we use blue for BDL speaker and diamond for L1 accent). In general, speaker embeddings of utterances from the same speaker form a cluster, and the boundary between different clusters are clear. Similarly, accent embeddings from speakers with the same accent form a cluster, verifying the correctness of our speaker and accent encoder. In terms of FAC syntheses, their speaker embeddings are distributed in the cluster of TXHC, and the accent embeddings lie in the clusters of L1 speakers

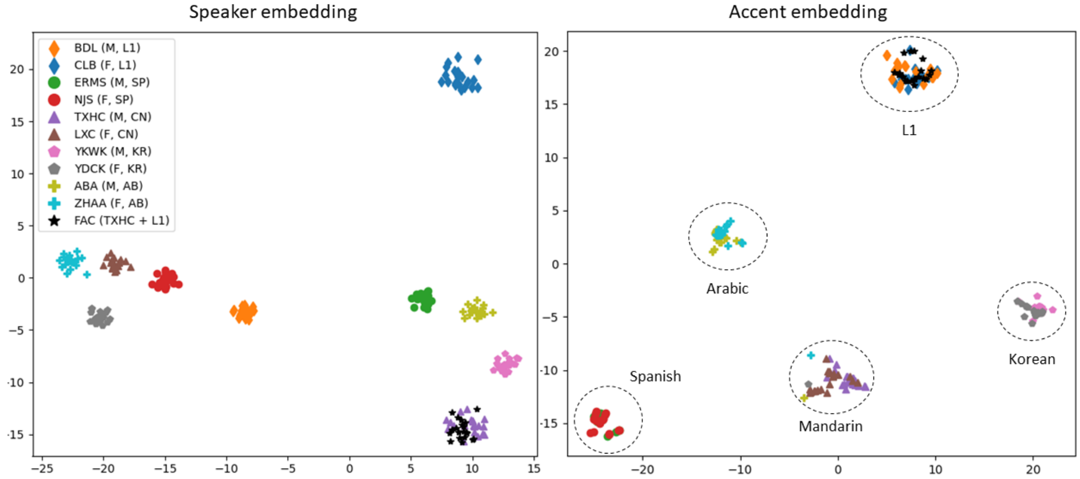


Figure 5.5: Speaker and accent embedding visualization of FAC syntheses for TXHC using t-SNE. Left: speaker embedding; right: accent embedding. Colors and shapes represent speaker and accent, respectively. Speakers in the legend are annotated with gender and accent. L1: native accent; SP: Spanish accent; CN: Mandarin accent; KR: Korean accent; AB: Arabic accent.

(BDL and CLB). These visualizations indicate that the FAC syntheses can successfully capture the voice identity of TXHC and an L1 accent.

We also conducted the same visualization on a “reverse” FAC task [24, 199], where the goal was to synthesize speech with the voice identity of a given L1 speaker but with an L2 accent. This is a straightforward process in our system, since we only need to change the inputs of the seq2seq model to use an L2 accent embedding and an L1 speaker embedding during inference. Here, we synthesize a voice that has the voice identity of CLB but with an L2 (Mandarin) accent. As shown in Figure 5.6, the speaker embeddings of reverse FAC syntheses lie in the cluster of CLB, whereas the accent embeddings lie in the cluster of Mandarin speakers (TXHC and LXC), indicating that the reverse FAC syntheses have a voice identity of CLB and a Mandarin accent.

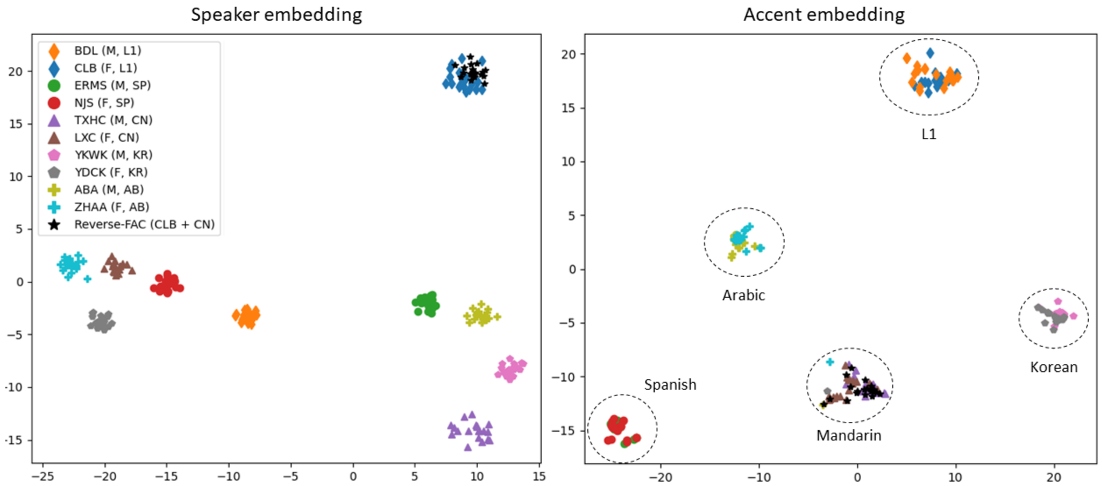


Figure 5.6: Speaker and accent embedding visualization of reverse FAC syntheses (CLB with a Mandarin accent) using t-SNE. Left: speaker embedding; right: accent embedding. Colors and shapes represent speaker and accent, respectively. Speakers in the legend are annotated with gender and accent. L1: native accent; SP: Spanish accent; CN: Mandarin accent; KR: Korean accent; AB: Arabic accent.

5.6.2. Subjective evaluations under standard FAC setting

In the first set of subjective evaluations, we evaluated the performance of the system under the standard FAC setting, i.e., the test L1 and L2 speakers were seen during training. During training, we used the union of the training sets of all 28 speakers. During inference, we used BDL as the L1 speaker, who then had been "seen" during training. First, we compared the proposed approach against two state-of-the-art FAC approaches:

- **Baseline1:** the system proposed by Zhao et al. [27], a one-to-one FAC approach based on seq2seq model. This baseline system trains a seq2seq PPG-to-speech synthesizer for each L2 speaker, and drives the synthesizer with PPGs extracted from an L1 speaker. As such, the baseline system requires training separate models for each L2 speaker.

- **Baseline2:** the system proposed by Liu et al. [28], a reference-free many-to-many FAC approach based on a novel recognizer-synthesizer architecture. The system is trained on 105 speakers from CSTR VCTK dataset [158]. Audio samples were produced by feeding the test utterances through their system, which is provided as a courtesy by Liu et al. Due to the implementation differences between the systems, we conducted two post-processing steps to ensure a fair comparison. First, as the accent conversion model of Liu et al. was trained on VCTK speakers, the stop-token predictions on L2-ARCTIC test utterances are not robust, occasionally resulting in a few seconds of white noise at the end of speech in accent conversion syntheses. To solve this issue, we manually removed the trailing white noises in these test utterances. Second, we resampled the syntheses of Liu system from 22.05 kHz to 16 kHz to make the sampling rate be consistent with other systems.

We conducted listening tests through Amazon Mechanical Turk[13] to rate three perceptual attributes of the synthesized speech:

[13] https://www.mturk.com

- **Accentedness**: The test asked participants to rate the degree of foreign accentedness of each utterance in a 9-point scale (1-no foreign accent; 9-very strong foreign accent), which is commonly used in the pronunciation literature [200]. Participants were told that the native accent in this task was General American.
- **Acoustic quality**: The test asked participants to rate the acoustic quality of each utterance through a standard 5-point Mean Opinion Score (MOS; 1-bad, 5-excellent).
- **Voice identity**: The test asked participants to rate the voice similarity between the FAC syntheses and the original L2 speech through a 14-point Voice Similarity Score (VSS) [155]. For each FAC-L2 utterances pair, participants were required to decide whether the two utterances were from the same speaker and then rate their confidence in the decision on a 7-point scale (1: not confident at all; 3: somewhat confident; 5: quite a bit confident; 7: extremely confident). The VSS was computed by collapsing the above two fields into a 14-point scale: -7 (definitely different speakers) to +7 (definitely the same speaker). To minimize the influence of accent, the two utterances had different linguistic content and were played in *reverse*, following [4].

Instructions were given in each test to help participants focus on the target speech attribute. For example, in the accentedness test, we asked participants to "*Try to ignore the audio quality (noise, distortions). Please focus only on the speaker's accent, for example, their pronunciation, rhythm, and fluency*". Test utterances were randomly selected from our test set, and the presentation order was counter-balanced. Additionally, in each listening test, we included five calibration utterances to detect if participants were cheating. We excluded ratings of the calibration utterances from the data analysis [201]. We recruited 18 participants for each listening test. All participants resided in the United States, and they passed a qualification test that asked them to identify different regional accents in the United States.

5.6.2.1. Comparison to the baseline system

Accentedness. Participants rated 20 utterances per system (5 utterances for each test L2 speaker). These utterances shared the same linguistic content across all the systems to ensure a fair comparison. Additionally, participants also rated the same set of sentences from the original L1 and L2 speakers as a reference. Results are shown in the second column of Table 5.2. Our proposed system received an average 3.39 accentedness score, which is significantly better (i.e., lower) than both baselines (Baseline1: 4.63, 27% relative improvement, $p \ll 0.001$; Baseline2: 6.25, 46% relative improvement, $p \ll 0.001$). The proposed system received significantly lower ratings of foreign accentedness than the original L2 utterance (7.11), although it still did not reach those of the original L1 utterance (1.06). These results suggest that our proposed seq2seq FAC model can effectively reduce foreign accentedness from the L2 speech.

Acoustic quality. As shown in the third column of Table 5.2, the proposed method achieved an MOS of 3.51, which is comparable to Baseline1 (3.47, $p > 0.05$) but significantly higher than Baseline2 (3.12, 13% relative improvement, $p \ll 0.001$). The original L1 speech received the highest MOS (4.90), followed by the original L2

Table 5.2: Accentedness (1-no foreign accent, 9-very strong foreign accent) results and acoustic quality (1-bad, 5-excellent) results under standard FAC setting. All the results are shown as average ± 95% confidence intervals.

	Accentedness	**Acoustic quality**
Original L2	7.11 ± 0.21	3.67 ± 0.28
Original L1	1.06 ± 0.12	4.90 ± 0.10
Baseline1	4.63 ± 0.10	3.47 ± 0.14
Baseline2	6.25 ± 0.39	3.12 ± 0.13
Proposed	3.39 ± 0.14	3.51 ± 0.15

speech (3.67). Note that the MOS ratings of the proposed system are closer to those of the original L2 speech than to the original L1 speech, possibly due to the raters confounding *acoustic quality* with *intelligibility*; we discuss this issue in Section 5.7. Thus, the proposed system achieves similar (or better) acoustic quality as the baseline systems, but unlike them does not require training a separate model for each new test L2 speaker.

Voice identity. Participants rated 20 pairs of utterances per system (5 pairs of utterances for each test L2 speaker). Each pair consisted of a FAC utterance and an utterance randomly selected from the L2 speaker. Voice identity results are shown in Table 5.3. The proposed system achieved a 5.05 VSS, indicating that the participants were "quite confident" that the FAC syntheses and the L2 speech were produced by the same speaker. These ratings are comparable to those of Baseline1 (5.05 VSS, $p > 0.5$) and significantly higher than those of Baseline2 (3.81 VSS, 33% relative improvement,

Table 5.3: Voice identity results. Voice Similarity Score ranges from -7 (definitely different speakers) to +7 (definitely the same speaker) under standard FAC setting. All the results are shown as average ± 95% confidence intervals.

	Voice Similarity Score
Baseline1	5.05 ± 0.28
Baseline2	3.81 ± 0.29
Proposed (All pairs)	5.05 ± 0.31
Proposed (Intra-gender)	5.29 ± 0.30
Proposed (Inter-gender)	4.80 ± 0.35

$p < 0.001$). It was worth noting that the L1 speaker in Baeline1 had the same gender as the L2 speaker, whereas the proposed system used the same L1 speaker for all L2 speakers. As a result, syntheses from the proposed system included both intra (same)-gender FAC pairs and inter (different)-gender FAC pairs, the latter being more challenging due to the differences in prosody and pitch range. Although the VSS on inter-gender pairs (4.80) was lower than that on intra-gender pairs (5.29) and Baseline1, the difference was not significant ($p = 0.14$). These results suggest that the proposed system can generate FAC syntheses that greatly resemble the voice identity of L2 speakers of any gender, using a canonical reference L1 speaker.

5.6.2.2. Performance on reverse FAC

To evaluate our system on the reverse FAC task, we synthesized testing utterances using the accent embeddings from NJS, TXHC, YKWK, and ZHAA, and the speaker embedding from BDL. Table 5.4 shows the accentedness, acoustic quality, and

Table 5.4: Accentedness (1-no foreign accent, 9-very strong foreign accent) results, acoustic quality (1-bad, 5-excellent) results, and voice identity results (-7-definitely different speakers, +7-definitely the same speaker) of reverse foreign accent conversion under standard condition. All the results are shown as average ± 95% confidence intervals.

	Accentedness	Acoustic quality	Voice identity
Original L2	7.11 ± 0.21	3.67 ± 0.28	-
Original L1	1.06 ± 0.12	4.90 ± 0.10	-
Proposed	5.58 ± 0.35	3.24 ± 0.17	4.91 ± 0.34

voice identity results of the reverse FAC evaluation. Our proposed system received a 5.58 rating of accentedness, much closer to that of the original L2 speech (7.11) than to the original L1 speech (1.06), indicating that our approach was able to impart an L2 accent to utterances from an L1 speaker. The proposed system also received a 3.24 MOS, significantly lower (p=0.02) than the MOS of the "direct" FAC syntheses (3.51), a result that is likely due to the correlation between acoustic quality and intelligibility – see Section 5.6.2.1. Finally, the proposed system received a 4.91 VSS, indicating that raters were "quite confident" that the reverse FAC syntheses and the L1 speech were produced by the same speaker; we found no significant differences between the voice identity ratings of reverse and direct FAC syntheses. Thus, we can conclude that the proposed approach can also operate in the reverse direction, generating non-native utterances with the voice identity of a native speaker.

5.6.3. Subjective evaluations under zero-shot FAC setting

In the second set of subjective evaluations, we evaluated the proposed system under the zero-shot FAC setting, where the L1 speaker and/or the L2 speaker were

Table 5.5: The four conditions in zero-shot FAC experiment

		L1 speaker	
		Seen	**Unseen**
L2 speaker	**Seen**	Condition SS	Condition US
	Unseen	Condition SU	Condition UU

unseen during training. The zero-shot FAC setting is appealing for real-world applications since it requires minimal data from the target speaker. First, we compared the performance of the proposed approach when using seen/unseen L1 or L2 speakers during inference. Then, we characterize the performance of the proposed method as a function of the number of available L2 utterances.

5.6.3.1. Comparing different conditions in zero-shot foreign accent conversion

We considered four different conditions in this experiment, as summarized in Table 5.5 In condition SS, the L1 speaker and the L2 speaker were both seen during training. Note this condition is the same as the system evaluated in Section 5.6.2.1, so it serves as a best-case scenario. In condition US, the L1 speaker was unseen during training, and the L2 speaker was seen during training. In condition SU, the L1 speaker was seen during training, and the L2 speaker was unseen during training. Finally, in condition UU, the L1 speaker and the L2 speaker were both unseen during training. Thus, condition UU was the most challenging of the four.

To ensure that the test speakers were unseen during training, we trained four models using different training sets. In Condition SS, we used the same model as in the standard FAC condition. In Condition US, we excluded CLB from the training set and used it as the test L1 speaker. In Condition SU, we excluded the four test L2 speakers

Table 5.6: Accentedness (1-no foreign accent, 9-very strong foreign accent) results, acoustic quality (1-bad, 5-excellent) results, and voice identity results (-7-definitely different speakers, +7-definitely the same speaker) under zero-shot FAC condition. All the results are shown as average ± 95% confidence intervals.

	Accentedness	**Acoustic quality**	**Voice identity**
Condition SS	3.39 ± 0.15	3.51 ± 0.14	5.15 ± 0.28
Condition US	3.33 ± 0.26	3.47 ± 0.13	4.99 ± 0.30
Condition SU	3.35 ± 0.25	3.50 ± 0.12	4.92 ± 0.28
Condition UU	3.30 ± 0.26	3.43 ± 0.12	4.59 ± 0.34

from the training set. In Condition UU, we excluded the four test L2 speakers and CLB from the training set, and we also used CLB as the test L1 speaker. For unseen L1/L2 speakers, we used the 50 utterances from the test set to generate the accent/speaker embedding. As before, we conducted three types of listening tests through Amazon Mechanical Turk to rate the accentedness, acoustic quality, and voice similarity of the synthesized speech. In addition, we kept the participants the same as those in the first experiment, so that the results are comparable between different experiments (e.g., participants in the accentedness test for the two experiments were the same).

Results from the accentedness, acoustic quality, and voice identity tests are shown in Table 5.6. We found no statistically significant differences between condition SS (best-case scenario) and the three more challenging conditions (US, SU, UU); $p > 0.5$ in all cases. This result suggests that the proposed system generalizes to unseen L1 or (and) L2 speakers during inference without any degradation in accentedness, acoustic quality, and voice identity.

5.6.3.2. Influence of the number of available L2 utterances

For practical FAC applications, it is important to understand the minimum amount of data needed from a target speaker. Requiring L2 learners to record a large amount of speech before they can hear their “golden speaker” voice can be tedious and demotivating. On the other hand, training the system with insufficient speech data might significantly degrade performance. To characterize the data requirements of the system under zero-shot FAC condition, we measured the performance of the UU codition (unseen L1 speaker and unseen L2 speaker) as a function of the number of available L2 utterances. We used the UU condition since it is the most flexible for real-world applications (e.g., computer-assisted pronunciation training), and also the most challenging, which provides a lower bound of the performance for the proposed approach. For the results shown in Table 5.7, condition UU used 50 test L2 utterances to produce the speaker embedding during inference. For this experiment, we reduced the number from 50 to 1 ($N = 50, 20, 10, 5, 1$) and re-evaluated system performance.

Results are shown in Table 5.7. Reducing the number of utterances from 50 to 1 has no impact on any of the three perceptual measures ($p > 0.5$ in all cases). These results indicate that as little as a single utterance (~3 seconds of speech) is sufficient to generate accent conversions for a new unseen L2 speaker, with no impact on performance. To some extent, this is to be expected since the test utterances are only used to compute the speaker embedding. Thus, we also examined whether the test utterances could instead be more beneficial if they were used to finetune a pre-trained

FAC system. Starting with a pre-trained UU model, we finetuned the model on each unseen L1-L2 speaker pair (i.e., CLB-NJS, CLB-TXHC, CLB-YKWK, CLB-ZHAA) with $N = 50, 20, 10, 5, 1$ test utterances, resulting in 20 finetuned models (4 speakers; 5 models with different number of training utterances for each speaker) for the unseen L2 speakers. Results are also shown in Table 5.7. We observe performance degradations in all three measurements when reducing the number of utterances from 50 to 1. When there are 50 test utterances, the finetuned system shows a marginal improvement compared to the proposed system (i.e., without finetuning), though the differences are not statistically significant (Accentedness: 3.03 vs. 3.30, $p = 0.03$; Acoustic quality: 3.54 vs. 3.43, $p = 0.18$; Voice identity: 4.97 vs. 4.59, $p = 0.25$). When decreasing the number from 50 to 20, the proposed system achieves comparable performance as the finetuned system ($p > 0.5$). Surprisingly, fine-tuning the systems with fewer than 20 utterances degrades performance as compared to the proposed system. At the extreme case (with only 1 utterance), the proposed system significantly outperforms the finetuned system in all three measurements (Accentedness: 4.72 vs. 3.31, $p < 0.001$; Acoustic quality: 3.24 vs. 3.43, $p = 0.01$; Voice identity: 3.73 vs. 4.57, $p < 0.001$). These results verified the robustness of our proposed approach in zero-shot condition, and they also provide insights regarding the choice between zero-shot learning models and finetuning models in FAC.

Table 5.7 : Accentedness (1-no foreign accent, 9-very strong foreign accent) results, acoustic quality (1-bad, 5-excellent) results, and voice identity results (-7-definitely different speakers, +7-definitely the same speaker) with different numbers of available L2 (non-native) utterances during inference. All the results are shown as average ± 95% confidence intervals.

#L2 utterances	**Accentedness**		**Acoustic quality**		**Voice identity**	
	Proposed	**Finetuned**	**Proposed**	**Finetuned**	**Proposed**	**Finetuned**
50	3.30 ± 0.26	3.03 ± 0.24	3.43 ± 0.12	3.54 ± 0.11	4.59 ± 0.34	4.97 ± 0.27
20	3.30 ± 0.22	3.47 ± 0.18	3.45 ± 0.11	3.48 ± 0.11	4.68 ± 0.30	4.65 ± 0.23
10	3.34 ± 0.26	3.84 ± 0.18	3.44 ± 0.12	3.46 ± 0.11	4.59 ± 0.29	4.06 ± 0.26
5	3.32 ± 0.23	4.58 ± 0.10	3.43 ± 0.11	3.38 ± 0.10	4.42 ± 0.33	3.49 ± 0.34
1	3.31 ± 0.25	4.72 ± 0.08	3.43 ± 0.12	3.24 ± 0.13	4.57 ± 0.29	3.73 ± 0.35

5.7. Discussion

We have proposed a many-to-many system that can convert utterances from a source speaker to appear as if someone else, and with a different accent, had produced it. We thoroughly evaluated the system through a visualization and a series of perceptual listening experiments. In the visualization, we used t-SNE to visualize the speaker and accent embedding of the FAC syntheses and reverse FAC syntheses. Both visualization results show that the proposed method can capture the desired voice identity and accent. Another interesting question that we would like to investigate here is whether the speaker embedding also carries accent cues, as these two aspects are closely related in the perception of speech [181-184]. If the speaker embedding is still entangled with accent information, the FAC syntheses tend to have an incorrect accent that is introduced by speaker embeddings. To examine it, we visualize speaker embeddings of natural utterances from 16 speakers with 4 accents, as shown in Figure 5.7. According to the

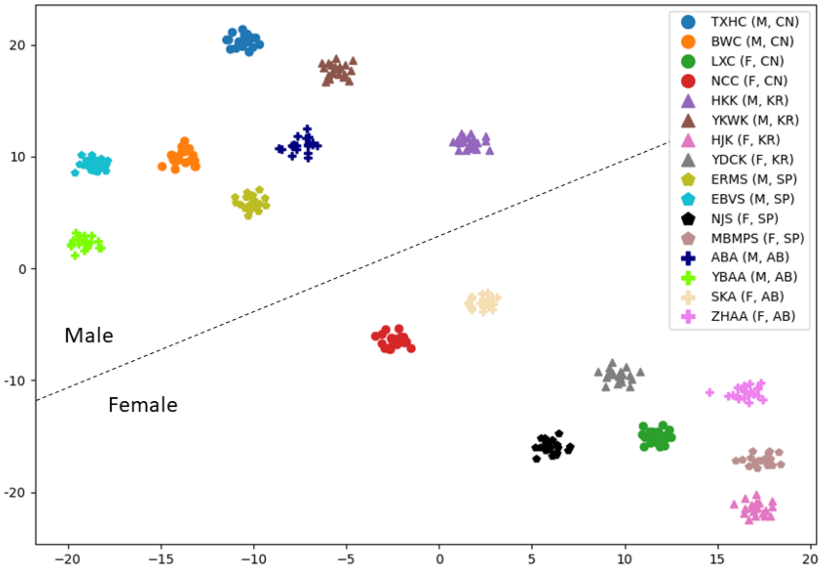


Figure 5.7: t-SNE visualization of the speaker embeddings from 16 speakers with 4 accents. Colors and shapes represent speaker and accent, respectively. . Speakers in the legend are annotated with gender and accent. L1: native accent; SP: Spanish accent; CN: Mandarin accent; KR: Korean accent; AB: Arabic accent.

figure, we cannot find any adjacency pattern between the speakers with the same accent. Instead, there is an obvious gap between the male speakers and female speakers. These observations suggest that speaker embedding mainly encodes information such as voice timbre and pitch, and there is no evidence for the entanglement between speaker and accent cues.

Next, we evaluated the proposed approach through three perceptual listening tests in standard FAC setting and compared it against two state-of-the-art FAC systems. Compared to the first baseline [27], the proposed system achieves significantly better (i.e., lower) ratings of foreign accentedness (3.39 vs. 4.63; $p < 0.001$), and similar acoustic quality (3.51 vs. 3.47; $p > 0.5$) and voice identity (5.05 vs. 5.05; $p > 0.5$) ratings, an important finding since the first baseline system builds a dedicated model for each pair of L1-L2 speakers, which one would expect would help capture voice identity more faithfully than a many-to-many system such as the one we have proposed. Compared to the second baseline [28], the proposed system achieves preferable ratings in all measurements (accentedness: 3.39 vs. 6.25; acoustic quality: 3.51 vs. 3.12; voice identity: 5.05 vs. 3.81; $p < 0.001$ in all cases). This comparison is necessary since the second baseline system is also a many-to-many system as ours, but instead of using L1 reference speech to encode the linguistic content and accent during inference, the baseline system uses L2 speech to encode the linguistic content and has no module to explicitly encode the accent. As suggested by the results, although the baseline system can avoid the need of L1 reference speech, the quality of their FAC syntheses is significantly inferior to ours. As a result, the linguistic content and accent carried by reference L1 speech are key to the integrity of FAC syntheses. Meanwhile, we believe that the use of L1 speech reference in FAC systems will not become a burden of CAPT, since a considerable number of L1 speech corpora are publicly available (e.g., Librispeech [168], VoxPopuli [202]). We also evaluated the proposed system on a reverse FAC task, where the goal was to impart a non-native accent to a native utterance. The results corroborates to our visualizations, suggesting that the reverse accent conversion can capture an L2 accent well while preserving the L1 speaker’s voice identity. Collectively, results from the direct and reverse FAC tasks indicate that the proposed system can disentangle linguistic content, voice identity, and accent in speech signals, instead of simply memorizing mappings between different speaker pairs.

The results of the perceptual listening tests in standard FAC setting lead to several additional observations. First, though the first baseline and the proposed system have similar architectures (i.e., both are based on a seq2seq model), the proposed system achieved significantly better (lower) ratings of accentedness. A possible explanation for this result is that the baseline system was trained using only utterances from an L2 speaker, i.e., at no point during training does it see L1 utterances. Thus, if the L2 speaker has systematic substitution or deletion errors (e.g., Mandarin speakers from certain areas systematically substitute /SH/ with /S/), the correct pronunciations will be missing in their utterances. When the model is driven by L1 PPGs during inference, it has to interpolate these missing pronunciations. The interpolation may not be accurate, and therefore, the syntheses could still retain considerable segmental errors. In contrast, the proposed system avoids this potential issue since it is trained using both L1 and L2 speech. Second, when evaluating voice identity, previous studies on FAC [23, 24, 27] generally use a reference L1 speaker that had the same gender as the target L2 speaker to avoid mismatches in pitch range. In contrast, our results show that our system can achieve similar voice identity ratings for intra-gender pairs and inter-gender pairs. Therefore, our system addresses these limitations and makes it possible to build a universal model for arbitrary L2 speakers using only a single reference L1 speaker. Finally, though the system achieves ratings of acoustic quality that are lower than those of the original L1 speech, the ratings are comparable to those of the original L2 speech. A possible explanation suggested by prior literature [4] is that the native listeners associate acoustic quality with intelligibility, and they may be influence by the intelligibility and provide lower ratings for non-native speech.

Lastly, we evaluated the proposed approach through the same perceptual listening tests in zero-shot FAC setting. First, we compared all four combinations in which the L1 speaker and L2 speakers were seen/unseen during training. Our results show that there are not significant differences among the four conditions in terms of accentedness, acoustic quality, or voice identity. Thus, the proposed system performs equally well under both standard FAC settings and any zero-shot FAC settings, which suggests that it can generalize to unseen L1 and L2 speakers without the need to re-train or finetune the model. Second, we characterized the performance of the system as a function of the number of available L2 utterances. Our results show that the accentedness, acoustic quality, and voice identity were not statistically significantly different when reducing the number of test utterances from 50 to 1, indicating that the system can retain its performance even using only one utterance (around three seconds of speech) from an arbitrary L2 speaker. Compared to previous FAC approaches, our approach avoids the need to collect a large corpus from testing L2 speakers (e.g., the users of computer-assisted pronunciation training software [203]), improving user experience and pronunciation training efficiency. Additionally, we also used these test utterances to finetune a pre-trained FAC system to see if finetuning would provide extra performance gains. Starting with 50 test utterances, we do observe marginal performance gain from the finetuning. However, when reducing the number of utterances from 50 to 1, the performance of the finetuned system starts to degrade and becomes inferior to the proposed system with 20 or fewer utterances. A possible reason to it is that the model overfits to a few utterances that were used for finetuning, and therefore the model loses generalizability to the test utterances, resulting in inferior performance.

5.8. Conclusion

In this paper, we propose a system that can generate accent conversion for any L2 speaker (seen or unseen). Our proposed approach is in contrast to most of existing FAC approaches, which require building a separate model for each L2 speaker. The proposed approach first uses separately trained models to extract L1 bottleneck features, L1 accent embeddings, and L2 speaker embeddings. Then it uses a seq2seq model to transform the L1 bottleneck features to accent-converted Mel-spectrogram, conditioned on an L1 accent embedding and L2 speaker embedding. Our results suggest that the proposed system can successfully transform L1 speech to match the voice identity of an L2 speaker while using a small amount of data from the L2 speaker.

6. APPLICATION OF VC/FAC ALGORITHMS: GOLDEN SPEAKER BUILDER–AN INTERACTIVE TOOL FOR PRONUNCIATION TRAINING[*]

6.1. Overview

The type of voice model used in Computer Assisted Pronunciation Instruction is a crucial factor in the quality of practice and the amount of uptake by language learners. As an example, prior research indicates that second-language learners are more likely to succeed when they imitate a speaker with a voice similar to their own, a so-called "golden speaker". This manuscript presents Golden Speaker Builder (GSB), a tool that allows learners to generate a personalized "golden-speaker" voice: one that mirrors their own voice but with a native accent. We describe the overall system design, including the web application with its user interface. Next, we present results from a user study in a language-instruction setting and collected their comments to GSB, which show that practising with GSB leads to improved fluency and comprehensibility. We suggest reasons for why learners improved as they did and recommendations for the next iteration of the training.

[*] © 2019 Elsevier. Reprinted, with permission, from S. Ding, C. Liberatore, S. Sonsaat, I. Lučić, A. Silpachai, G. Zhao, E. Chukharev-Hudilainen, J. Levis, and R. Gutierrez-Osuna, "Golden speaker builder–An interactive tool for pronunciation training," in *Speech Communication*, vol, 115, 2019, pp. 51-66. DOI: 10.1016/j.specom.2019.10.005. This reprint contains necessary modifications to include suggestions from the dissertation committee.

6.2. Introduction

Pronunciation teaching often includes practice with a teacher, who can guide learners individually and provide feedback in the correct manner and amount when necessary [204]. Yet this is often time- consuming and expensive when the educational institutions' benefits are taken into consideration. Additionally, this does not match up well with the way that teachers usually approach pronunciation teaching. Research shows that most teachers approach pronunciation teaching in an ad-hoc manner, that is, they address pronunciation issues mostly in presence of a salient error or an error causing a communication problem. This is mostly either because teachers do not have sufficient training [205] or self-confidence [206, 207] in pronunciation teaching. Another common belief among teachers is that pronunciation improvement will take care of itself with sufficient input and it does not require teaching in the way that other language skills do. This is a belief that was motivated by the principles of communicative language teaching which emphasized fluency over accuracy [208].

However, providing instruction and feedback on immediate production in pronunciation teaching is an essential pedagogical requirement for learners' improvement, even though it can demand extensive instructional interventions [209]. One solution to the lack of time and training of teachers is computer-assisted pronunciation training (CAPT) systems, which have been utilized to support learners to study autonomously and help teachers provide learners with individual feedback without using large amounts of time in class [210-213]. CAPT may also be motivating for many learners, both because of their interest in technology and because of learning preferences that make working with a computer program more comfortable than interacting with a real person. CAPT gives learners the chance to work on their pronunciation in a stress-free environment, at their own time and pace. For instance, pronunciation is a skill that may require extensive listening and repetition. Some learners may feel uncomfortable about asking for a repetition in class more than once, but with a CAPT program it is easier to make use of extensive repetition [214]. All said, CAPT offers great promise for individualized pronunciation instruction, more consistent practice, and greater comfort in learning [215].

With advancements in speech technologies such as automatic speech recognition (ASR) and speech synthesis, CAPT can also provide practice opportunities that a face-to-face class cannot. For example, the use of speech visualizations that adapt to each person's speech [216], the use of multiple voices in perceptual training [217-219], or the use of personalized voices [8] all provide learning opportunities that classroom pronunciation training cannot. The later idea (i.e., personalized voices), has resurfaced several times in the CAPT literature. It was first proposed nearly thirty years ago by Nagano and Ozawa [9]. In their pioneering study, Japanese learners were asked to practice with a model of their own voice that had been modified to match the prosody of a reference English speaker. Post-training utterances from these learners were rated as more native-like than those for a second group of learners who instead had practiced with the reference English voice. More than a decade later, Probst et al. [8] published a study in Speech Communication where L2 learners were asked to practice with a native speaker voice that had different characteristics. Participants who imitated a well-matched voice (i.e., one with characteristics similar to their own voice) improved more than those who imitated a poor match. This result led the authors to suggest that each learner has an ideal speaker voice to imitate, a so-called “Golden Speaker.” Nearly ten years later, and in an article also published in Speech Communication [4], we proposed that each learner’s Golden Speaker should be their own voice, resynthesized to have a native accent. Most notably, in that study we presented an accent-conversion technique that was able to correct not only the learner’s prosody (as Nagano and Ozawa had done) but also their segmental errors (i.e., phoneme substitutions, additions and deletions). Missing from our study, however, was a validation of the technique on pronunciation-training experiments. It has taken us nearly a decade to refine our initial accent-conversion technique to make it robust for deployment in the classroom. This is a clear next step. A decade since the first paper has shown that refining the accent-conversion technique for successful deployment in pronunciation training was more challenging than expected. The improvement we have seen in accent-conversion quality makes us optimistic for further successful deployment of the Golden Speaker algorithms.

This chapter describes a web application (Golden Speaker Builder; GSB) and the effectiveness of GSB being used in a language-instruction setting with a population of Korean L2 learners of English. The study was guided the research questions:

- **RQ1**: What is the effect of using the GSB on learners’ improvement of their comprehensibility and fluency?
- **RQ2**: What features of the GSB did learners find useful, and what did they find in need of improvement?

6.3. Review of the literature

6.3.1. Feedback in Second Language Pronunciation Acquisition

Feedback refers to “information learners receive in response to their communicative efforts” [220] (p. 210), which is a significant perspective in L2 learning. Prior studies have explored and emphasized the role of feedback in Second Language Acquisition (SLA). In [221], Swain and Lapkin suggest that output is equally important as input in SLA since output fosters deeper engagement with language than input alone. Similarly, Swain [222] highlighted the importance of output by stating “output may stimulate learners to move from the semantic, open-ended, strategic processing prevalent in comprehension to the complete grammatical processing needed for accurate production” (p. 99).

When it comes to the computer-assisted language learning (CALL) environment, the feedback forms are different from conventional oral classroom settings, due to the differences of the medium, as noted by Heift [223] (see Table 6.1). These types of feedback can also be provided to second language learners in CAPT applications, which are developed based on speech recognition/synthesis models. A first feedback type that may lead to an improvement in pronunciation is clarification. For example, speech recognition algorithms were used in CAPT to detect pronunciation errors in an utterance and thus provide a pronunciation score [224]. These scores may lead learners to repeat their performance until they get a satisfactory score, which shares the spirit of clarification in an oral classroom.

Table 6.1: Feedback types in the oral classroom and CALL environment [223] (p. 418)

Feedback type	Oral classroom	CALL
Explicit correction	You mean…	Correct answer
Recast	Teacher reformulation	Correct answer
Clarification	What do you mean?	Try again!
Meta-linguistic feedback	Explanation of error type	Explanation of error type
Elicitation	Ellipsis	Highlighting
Repetition	Intonation	Highlighting

Another effective type of feedback SLA is recast, a correct restatement of the mispronounced utterance. In CALL, it can be interpreted as the imitation of a correct utterance, mostly pronounced by a native speaker. A previous study [225] indicated that imitation exercise improves learners' perception [225], which proves their effectiveness in pronunciation improvement. However, questions about what voice a language learner should imitate, i.e., what factors lead to a '*golden speaker'*, led to new research directions in CALL. Probst et al. [8] advocated that it is beneficial for L2 learners to imitate a voice that is similar to their own voice in pronunciation training. In other words, the golden speaker voice would serve as a recast for the learner's production. Other research also shows that the choice of a golden speaker may depend on L2 learners' language background, proficiency, and learning stage. For instance, learners may go from slower to faster to find a comfortable utterance speed [7]. Additionally, Probst et al. [8] proposed that a CAPT application should provide learners multiple *golden speakers* to practice with; Wang and Lu [7] suggested that this means that these applications should provide learners chances to control voice features such as different speech rates and pitch formants based on their own preference, when synthesizing golden speaker voices.

6.3.2. Self-imitation in pronunciation training

A handful of studies have examined the possibility of modifying the learner's own voice and using it for pronunciation training [112-115, 226, 227]. In early work, Nagano and Ozawa [228] evaluated a prosodic-conversion method to teach English pronunciation to Japanese learners. One group of students was trained to mimic utterances from a reference English speaker, whereas a second group was trained to mimic utterances of their own voices, previously modified to match the prosody of the reference English speaker. Post-training utterances from the second group of students were rated as more native-like than those from the first group. More recently, Bissiri et al. [112, 113] used prosodic modification to teach German lexical stress to Italian speakers. Receiving feedback in the form of the learner's own voice (resynthesized to match the local speech rate, intonation, and intensity of a reference German speaker) was shown to be more effective than receiving feedback in the voice of the reference German speaker. Providing feedback in the learner's own voice also had a motivating effect, with several participants asking to continue the training, whereas participants in the control group showed no particular interest.

Pronunciation training with prosodic modifications of the learner's utterances has been shown to improve not only accentedness but also intelligibility. De Meo et al. [115] evaluated the effectiveness of two forms of training (imitation and self-imitation) to teach suprasegmental patterns of Italian to Chinese learners. Participants in the self-imitation condition heard their own voice, resynthesized to match the native model, whereas those in the imitation condition followed traditional imitation exercises. Native listeners were then asked to classify learners' post-training productions as belonging to one of four speech acts: requests, orders, granting, and threats. Classification performance was significantly higher for utterances from participants in the self-imitation group. Similar improvements in communicative effectiveness were obtained in a later study with Japanese learners of L2 Italian [114]. These studies show that (1) prosodic accent conversions are an effective tool to teach pronunciation to L2 learners, and (2) the effect is robust across several L1-L2 combinations. Incorporating segmental accent conversion–the next logical step in this new genre of technology–is the major contribution of our work.

6.3.2.1. Algorithms for segmental accent conversion

In contrast with the self-imitation literature, where no studies exist that incorporate segmental adjustments of the learner's own voice, the speech-processing literature offers a few studies on speech modification of segmental errors in non-native speech. These studies have shown that segmental modifications effectively reduce the perceived accent of an utterance than prosody modification alone, both within regional accents of the same language [229] and across languages [4].

In early work, Yan et al. [229] developed a method to transform vowels of three major regional English accents (British, Australian, and General American). The authors built a statistical model of vowel formant ratios from multiple speakers, and then extracted empirical rules to modify pitch patterns and vowel durations across the three accents. Using this model, the authors then adjusted formant frequencies, pitch patterns and vowel durations of an utterance to match a desired target accent. In an ABX test, 78% of Australian-to-British accent conversions were perceived as having a British accent, and 71% of the British-to-American accent conversions were perceived to have an American accent. In both cases, changing prosody alone (pitch and duration) led to noticeable changes in perceived accent, though not as significantly as formant modifications. The method hinged on being able to extract formant frequencies, so it cannot be easily extended to larger corpora because formant frequencies are ill-defined for unvoiced phones and cannot be tracked reliably even in voiced segments.

A few studies have attempted to blend L2 and L1 vocal tract spectra instead of completely replacing one with the other. In one such study, Huckvale and Yanagisawa [230] reported improvements in intelligibility for Japanese utterances produced by an English text-to-speech (TTS) after blending their spectral envelope with that of an utterance of the same sentence produced by a Japanese TTS. Felps et al. [4] proposed a suitable method for voiced and unvoiced phones. The authors split short-time spectra into a spectral envelope and flat glottal spectra. Then, they replaced the spectral envelope of an L2 utterance with a frequency-warped spectral envelope of a parallel L1 utterance and recombined it with the L2 glottal excitation. Listening tests showed a significant reduction in accent following segmental modification. More recently, Aryal et al. [21] presented a voice morphing strategy that can be used to generate a continuum of accent transformations between an L2 speaker and a native speaker. The approach decomposes the speech Cepstrum into spectral slope and spectral detail, then generates accent conversions by combining the spectral slope of the L2 speaker with a morph of the spectral detail of the native speaker. This morphing technique provides a tradeoff between reducing the accent and preserving the voice identity of the L2 learner, and it may serve as a behavioral shaping strategy in computer-assisted pronunciation training.

Accents originate from differences in articulation, which suggests that articulatory information may be useful in accent conversion. To explore this possibility, Felps et al. [52] used concatenative speech synthesis to replace mispronounced diphones in an L2 utterance with other L2 diphones whose articulatory configuration was similar to a reference native utterance. The approach reduced the perceived non-native accents by 20%, but performed poorly when tasked with finding phonemes that the L2 did not utter. To address this problem, Aryal and Gutierrez-Osuna [26] proposed a statistical parametric approach, which trains a GMM-based articulatory synthesizer for the L2 speaker, then drives it with articulatory data from a reference native utterance mapped to the L2 articulatory space via a Procrustes transform. In listening tests, the authors found that the method reduced the perceived non-native accents while preserving the voice quality of the L2 speaker. However, these methods require articulatory data, which is impractical for pronunciation training.

6.3.3. Comprehensibility and Fluency

In this paper, we assessed L2 learner's speech productions using comprehensibility and fluency, which are partially independent measures of speech understanding [200]. Comprehensibility refers to the amount of cognitive effort put forth by listeners in understanding speech [231]. Highly comprehensible speech is thus easy to understand, taking little extra effort. Comprehensibility is closely related to accentedness, but comprehensibility may be a better predictor of communicative success than accentedness in evaluating the success of pronunciation training [232]. In contrast with accentedness ratings, comprehensibility ratings correlate with a wide range of features beyond pronunciation, including prosodic skills, fluency features, features related to vocabulary and grammatical complexity, and discourse features related to the construction of oral texts [233-235].

Another feature we evaluated in this study is fluency. Fluency is not directly related to pronunciation accuracy, but is instead a measure of how automatically speech is produced. Fluency is connected to a wide variety of temporal features of speech (i.e., speech rate, the use of pauses, and repairs), the use of formulaic language [236], whether phrases are logically constructed [237], phonological features of speech [238], interactive characteristics of speech in conversation [239], perceived smoothness of speech by listeners [119], mean length of run (see [240]), and automaticity of speech production [241]. Fluency is not independent of accentedness and comprehensibility but is indirectly related to both. For example, comprehensibility ratings and fluency ratings correlate with perspectives in common [233]. In addition, speech rate is also predictive of fluency judgments [242, 243], and similar judgments of fluency may be given for speech at different rates. Listeners are sensitive to whether speech is fluent, and speech that is heard as too fast or too slow may also be heard as more accented or as less comprehensible [234].

6.3.4. Effects of instruction

Three recent studies were proving the instruction from either human teachers or in CAPT to improve the pronunciation of L2 learners. First, Saito [244] designed 15 pre-/post-test studies to explore if the instruction can lead to pronunciation improvement. The author concluded that explicit attention to pronunciation typically led to improvement, where improvement was more common in controlled tasks and less common in spontaneous speech. Second, Lee, Jang, and Plonsky [245] conducted a study through a meta-analysis of 86 studies to explore the success of pronunciation instruction. Their results suggested significant improvements, especially when the instruction was carried out over longer time periods, and the learners were receiving consistent feedback. Lastly, Thomson and Derwing [219, 246] analyzed most of the studies in Lee et al., [245], but focusing on what pronunciation training should be like. Their study examines improvements in comprehensibility, but most results that show improvements in global ratings privilege prosody rather than segmentals. In summary, all three studies suggest that interventions should be successful, and that explicit attention to pronunciation should lead to improvement. However, they do not consider implicit feedback, such as golden speaker voice. As a result, it would be meaningful to explore if this type of feedback is sufficient to lead to improvement in comprehensibility and fluency, which is the focus of our study.

6.4. System description

To answer the Research Questions presented earlier, we developed Golden Speaker Builder (GSB), an online interactive tool that allows L2 learners to build a personalized pronunciation model: their own voice producing native-accented speech (i.e. a “golden speaker”). To build their golden speaker, L2 learners follow three steps. In the first step, the learner records a keyword for each phone (e.g., for phoneme /ʒ/, the learner records the keyword “vision”) under the guidance of an instructor to ensure that the utterance has near-native production. After recording each keyword, the learner segments the phone using a graphical display of the waveform. In the second step, the learner records several sentences, which are used to estimate the learner’s pitch statistics. In a final step, the learner selects a native speaker as a source model, and GSB resynthesizes the native speaker’s sentences using the recorded phone segments and prosody statistics of learner. The process can be completed in less than thirty minutes and generates a Golden Speaker voice that produces intelligible speech with the voice quality of the L2 learner, and the prosody of the source native speaker normalized to the pitch range of the L2 learner.

The software architecture of GSB is shown in Figure 6.1. GSB consists of three components: a web application, a signal processing back-end, and a middleware to connect the signal processing back-end to the web application. The web application provides a graphical interface for the learner, responds to the learner’s requests, and

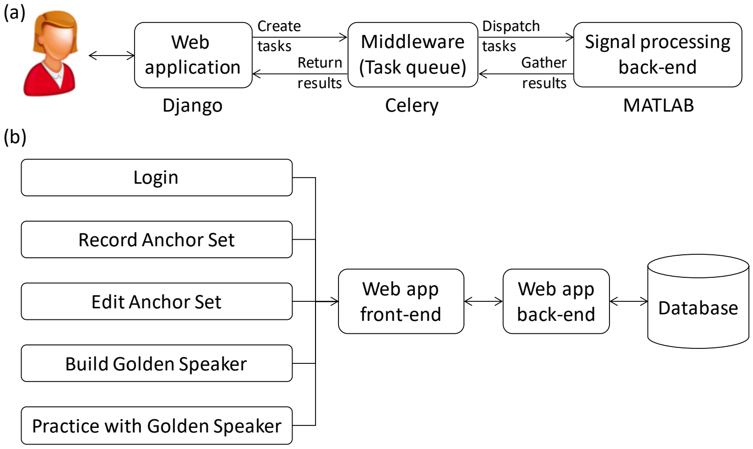


Figure 6.1: (a) Overall software architecture. (b) Architecture of the web application

stores the learner's data (i.e., login information, speech recordings, and golden speakers) onto a database – see Figure 6.1b. The signal processing back-end runs the accent conversion algorithms, which generates synthesized speech for each Golden Speaker model. Finally, the middleware layer provides communication between the web application and the signal processing back-end via an asynchronous task queue. Detailed descriptions of each component are included in the following subsections.

6.4.1. Web application

We implemented the web application using the Django framework[14]. The web-app front-end was written in HTML5 and Javascript, and decorated with Bootstrap[15], whereas the

[14] https://www.djangoproject.com/
[15] https://getbootstrap.com/

web-app back-end was written in Python with Django internal modules. User data is managed by an SQLite database engine[16] on a standard Linux file system. We hosted the web application through Nginx[17]. To follow the workflow described below, we provide five functional modules: Login; Record Anchor Set; Edit Anchor Set; Build Golden Speaker; and Practice with Golden Speaker.

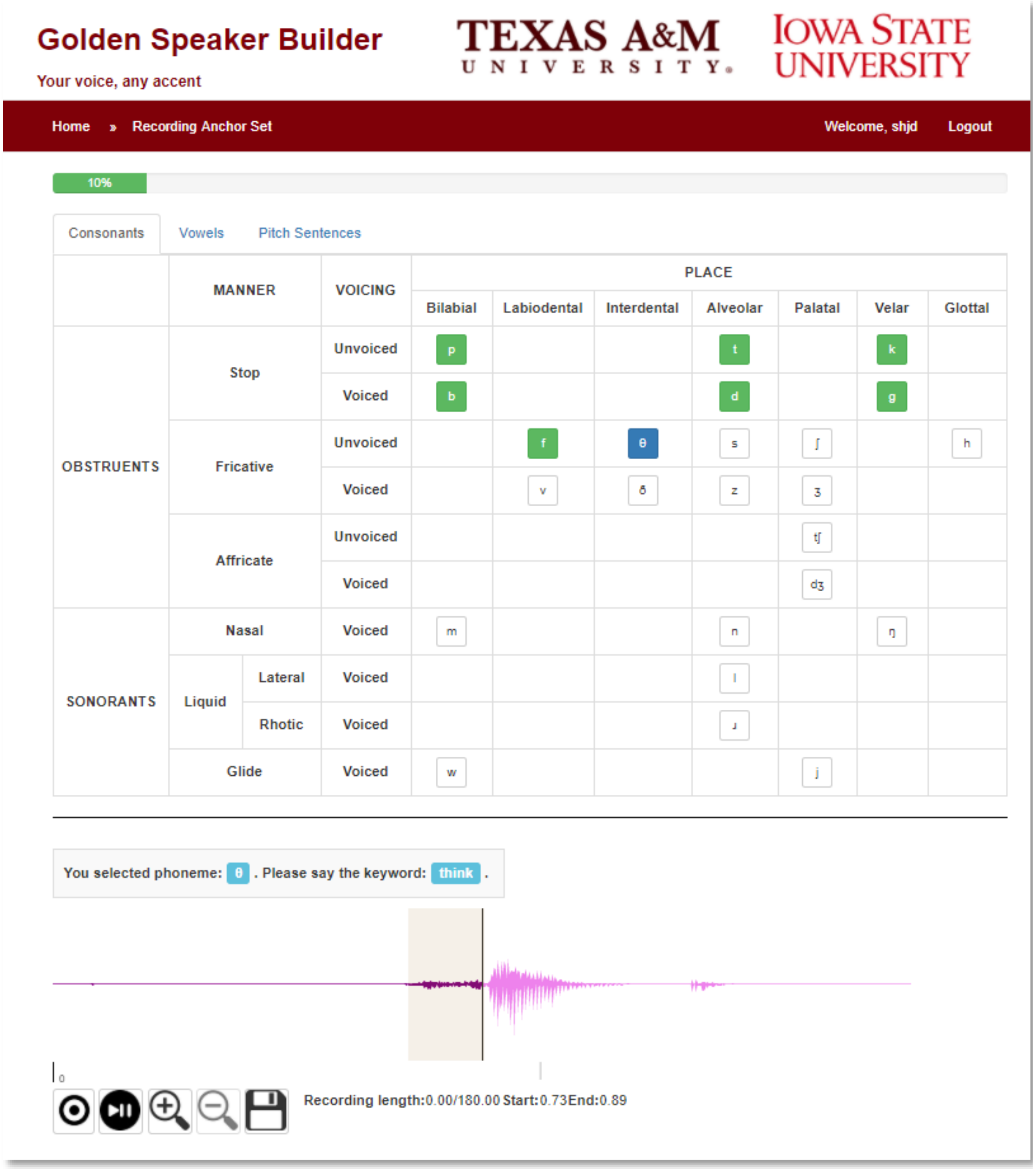
The **Login** module provides registration and login functions. To use GSB, learners must register an account using their email, and login with their registered account and password. We implemented this module using Auth0 authentication[18], and connected Auth0 to the SQLite database to save the users' account information. This module guarantees the privacy of learners' information and ensures that each learner can only operate on their own information and data.

The **Record Anchor Set** module enables learners to record keywords and prosody sentences, later used to build a Golden Speaker model. As shown in Figure 6.2, the learner must record a keyword for each of the 40 phones in American English (CMU phone set[19]). Once a user records a keyword, the interface allows the learner to segment the phone segment (or "Anchor") by highlighting the corresponding region of the speech

[16] https://www.sqlite.org/
[17] https://www.nginx.com/
[18] https://auth0.com/
[19] http://www.speech.cs.cmu.edu/cgi-bin/cmudict

waveform. Separate tabs are used for consonants, vowels, and pitch sentences. Consonants are arranged according to their place and manner of articulation, and vowels are arranged according to their frontness and height (not shown). This arrangement allows the teacher and learner to review the basic organization of speech sounds in English, as the learner records the various keywords. The “Pitch Sentences” tab includes 30 sentences representative of conversational speech (e.g., “What time does the bus leave for the airport?”) that were deliberately selected to provide good coverage of various prosodic contexts, and a free-speech exercise in which the learner first watches a 3-minute short film[20] and then records a 1-2 minute audio summary. Recordings for all the keywords and pitch sentences are saved on the file system, whereas the segmentation information is saved in the database. In a final step, both the recordings and the segmentation information are sent to the signal processing back-end.

[20] “Spellbound” by Ying Wu and Lizzia Xu; available at youtube.com/watch?v=W_B2UZ_ZoxU



Golden Speaker Builder

Your voice, any accent

	MANNER		VOICING	PLACE						
				Bilabial	Labiodental	Interdental	Alveolar	Palatal	Velar	Glottal
OBSTRUENTS	Stop		Unvoiced	p			t		k	
			Voiced	b			d		g	
	Fricative		Unvoiced		f	θ	s	ʃ		h
			Voiced		v	ð	z	ʒ		
	Affricate		Unvoiced					tʃ		
			Voiced					dʒ		
SONORANTS	Nasal		Voiced	m			n		ŋ	
	Liquid	Lateral	Voiced				l			
		Rhotic	Voiced				ɹ			
	Glide		Voiced	w				j		

You selected phoneme: θ . Please say the keyword: think .

Recording length:0.00/180.00 Start:0.73End:0.89

Figure 6.2: Graphical user interface for recording consonants in American English. In the example shown, the learner has already recorded keywords for all the stop consonants (highlighted in green), has recorded the phone /θ/ (highlighted in blue) and is in the process of selecting the appropriate section in the speech waveform shown at the bottom of the page.

Table 6.2: Keyword selection. The following is a list of keywords used to build anchor sets for L2 learners in the GSB application. Phoneme names are shown on the left column in ARPABET notation, and the words used to elicit the phoneme on the left.

AA	*father*	**CH**	*cheat*	**HH**	*heat*	**NG**	*sing*	**TH**	*think*
AE	*ash*	**D**	*deep*	**IH**	*if*	**OW**	*oh*	**UH**	*push*
AH	*us*	**DH**	*this*	**IY**	*east*	**OY**	*toy*	**UW**	*boot*
AO	*horse*	**EH**	*"s"*	**JH**	*jeep*	**P**	*poke*	**V**	*vote*
AW	*ouch*	**ER**	*earth*	**K**	*keep*	**R**	*reads*	**W**	*weeds*
AX	*sofa*	**EY**	*ace*	**L**	*leads*	**S**	*See*	**Y**	*yes*
AY	*ice*	**F**	*feed*	**M**	*make*	**SH**	*sheep*	**Z**	*zoo*
B	*boat*	**G**	*gust*	**N**	*no*	**T**	*tea*	**ZH**	*vision*

We selected one keyword per phoneme to capture an “ideal” example of that phoneme or its main characteristic, e.g., the dominant allophone of that phoneme. Voiceless aspirated stops are more distinct than unvoiced aspirated stops, and were chosen preferentially for that reason. Additionally, final stops were avoided, as well as final rhotics and velarized approximants (e.g. “dark L”). The full selection of keywords is shown in Table 6.2.

The **Edit Anchor Set** module allows learners to make changes to a previously recorded “Anchor Set”. This includes re-recording specific keywords or pitch sentences, and making corrections to the segmentations. Learners also have the option to rename, copy, and delete the Anchor Sets from their profile. Once an Anchor Set is modified, the updated recordings and segmentation information are automatically sent to the signal processing back-end.

The **Build Golden Speaker** module allows learners to select one of several Native Speaker (NS) voices, each containing hundreds of sentences, and pair it with one of their own Anchor Sets. Once a particular NS voice, Anchor Set, and list of sentences has been selected, this information is sent to the signal-processing back-end to build the Golden Speaker model.

The **Practice with Golden Speaker** module allows the learner to practice pronunciation with any of the previously-built Golden Speakers. For example, we used a *backward buildup* exercise as one technique for pronunciation practice, where the learner practices a long sentence starting from the last phrase and adding complexity in a backwards fashion. As an example, given the practice sentence "*We're going to the supermarket to buy vegetables for dinner,*" the learner produces the phrase "*for dinner*," then the phrase "*to buy vegetables for dinner*" and so forth.

6.4.2. Speech processing back-end

To build Golden Speakers, the signal processing back-end uses a Sparse, Anchor-Based Representation (SABR) reported in prior work [247, 248]. The motivation behind SABR is to separate speaker-dependent cues (*how* something was said) from speaker-independent ones (*what* was said). Then, we combine the speaker-independent cues extracted from the source speech with the speaker-dependent cues from the target speaker to produce the accent-conversions. For more details about SABR, we refer readers to the original studies by Liberatore et al. [247, 248]. Noted that other foreign accent conversion algorithms can also be used in the signal processing back-end on GSB (e.g., the state-of-the-art zero-shot foreign accent conversion algorithm described in

Chapter 5), which we leave as future work of this dissertation, as we will discuss in Chapter 7.3.4.

The signal processing back-end is implemented using MATLAB. Specifically, the signal-processing tasks are divided into two folds: (1) building a SABR model for a given Anchor Set, and (2) synthesizing speech for a Golden Speaker. In addition, the two tasks can be further divided to several function units:

- **Speech Analysis:** The pre-process of speech signals, including signal resampling, signal normalization, extracting spectral features (e.g., MFCC), and computing prosody features (e.g., pitch contour).
- **Construct SABR model:** Construct SABR model using extracted MFCC from the learners' speech signal. Once the SABR model is accomplished, it will be saved as a ".mat" file into the file system, whose path will be saved into the database.
- **Construct pitch model:** Construct pitch model using extracted pitch contour**.** Similarly, the pitch model is saved as a ".mat" file into the file system, whose path is saved into the database.
- **Synthesize GS voice:** Synthesize the Golden Speaker voice for learners to practice with. This function is consists of two sub-functions. First, it generates the spectral features and pitch contours of the Golden Speaker voice, using the SABR models and pitch models that is computed previously. Then, it synthesizes the audio waveform of Golden Speaker voice using a vocoder.

6.4.3. Middleware

GSB uses an asynchronous task queue, Celery [249], as the middleware to communicate between the web application and the signal processing back-end. Each time the user submits a request containing signal processing operations, the web application creates a task worker and pushes it into the asynchronous task queue. Tasks in the queue are then dispatched to an available worker, which in turn calls the appropriate signal processing function in the back-end. Once the task is complete, results are sent back to the web application through the asynchronous task queue, and the worker is set to be available.

Two types of signal-processing tasks are included in GSB: (1) building a SABR model for a given Anchor Set, and (2) synthesizing speech for a Golden Speaker. Tasks of the first type are dispatched after a complete Anchor Set is recorded and saved. This involves passing all the recordings (keywords, pitch sentences) and segment information to the signal processing back-end, saving the SABR model to the file system, and passing the corresponding path to the web application so it can be stored in the database. The run time to build a SABR model is 10 minutes, largely due to the STRAIGHT speech analysis (~5 seconds processing time for 1 second of speech). Tasks of the second type are dispatched when the user submits a request to build a Golden Speaker. This involves passing the following information to the signal-processing backend: the teacher’s SABR model (computed far in advance), the learner’s SABR model (computed from the Anchor Set), and a list of sentences the learner wants to synthesize. Once these sentences have been re-synthesized as a Golden Speaker, the recordings are saved to the

Linux file system, and the corresponding path is returned to the web application so it can be stored in the database. The run time for this type of task is approximately 10 seconds/sentence.

6.5. User study[21]

We conducted a user study to validate GSB in a language-instruction setting with a population of Korean L2 learners of English. The study followed a quasi-experimental pre-, immediate post- and delayed post-test at a midwestern university in the USA. Learners took a pre-test followed by three weeks of CAPT using the GSB, followed by an immediate post-test one week after training and a delayed post-test three weeks after training. Learners were interviewed after each test session.

6.5.1. Participants

There were two groups of participants in this study: learners and raters. Learners were 15 Korean learners of English (eight male) majoring in various fields of study. Learners were recruited from undergraduate and graduate ESL courses when one of the researchers introduced the study in a classroom visit. Initially, 18 learners signed up to

[21] The user study of Golden Speaker Builder was conducted by S. Sonsaat, I. Lučić, A. Silpachai, E. Chukharev-Hudilainen, and J. Levis at Iowa State University. Since the focus of this chapter in this dissertation is the development of GSB, only the learners' GSB experiences were reported in this section, corresponding to the second research question. For other results of this study, we refer readers to the original publication [203].

participate the study; however, we did not include the data from three of these participants since they missed at least two training sessions.

Raters included 95 native-English speaking undergraduate students majoring in different areas at the same university. These raters were part of two groups since comprehensibility (n=50), and fluency (n=45) were each rated by a separate group of raters. All raters were recruited from first- and second-year composition classes through the introduction of the study by one of the researchers in a classroom visit. Learners and raters were recruited through convenience sampling; that is, we collected data from all students who were willing to participate.

6.5.1.1. Pronunciation challenges for Korean speakers in English

We chose to use Korean speakers because of the high likelihood that they would have both segmental and suprasegmental difficulties with English. We also chose Korean learners because different Korean learners often have similar types of difficulties, even at more advanced levels of English proficiency. Among the most notable differences between the English and the Korean sound systems are that Korean vowels do not have a tense vs. lax distinction, and voiced and voiceless sounds are not regarded as different [250].

L1 Korean learners find both segmental and suprasegmental features of English challenging. Lee [250] lists the vowel and consonants sounds of English most likely to cause issues. Among vowels, /ɔ/ is problematic, as it does not exist in Korean, so Korean speakers of English tend to assimilate it to a pure /o/ [251]. Additionally, English /ʌ/ is often pronounced by Koreans as /ɑ/, while English /æ/ is assimilated to Korean /e/. The

Korean sound system does not include the sound /ɝ/, which is frequently confused with /ɔ/. Therefore, differentiating words such as “work” and “walk” is difficult both in perception and production.

For consonant sounds, Korean learners of English do not have a voiced vs. voiceless distinction as in English. Therefore, word pairs such as “log” and “lock”, “raised” and “raced”, “beach” and “peach”, etc., are often confused [250]. Voiced and voiceless distinctions are also not found in stops and affricates. Korean has three phonemic voiceless stops (such as /p/, p^h/ and /pp/) for the bilabial, alveolar and velar places of articulation where English has two phonemes distinguished by voicing. The same pattern holds for the post-alveolar affricate /tʃ/. The lack of phonemic stop-fricative distinctions in Korean also leads to challenges with /b/-/v/ and /f/-/p/, as in “defend” and “depend” [251]. Another common challenge is the English distinction between /ɹ/ and /l/, mapping to a single Korean phoneme. Other consonant sounds not found in Korean are /z/, /ð/, and /θ/, and they are frequently assimilated to /dʒ/, /d/, and /s/, respectively. Apart from having difficulties with consonant sounds because they are not present in the Korean sound system, Korean learners of English also have difficulties with certain similar consonant sounds in specific environments. So, /ʃ/ and /tʃ/ are part of Korean but are not found in syllable codas. As a result, Korean learners often add either /ɪ/ or /ə/ to English words ending in these sounds to match Korean syllable structure constraints [250].

Prosodically, in Korean each syllable has similar emphasis, and each word in a sentence has the same prominence. This may sometimes cause it to be characterized as monotonous-sounding [251]. Korean and English also differ in the ways that they use intonation, and especially in how English uses flexibly-placed lexical prominence to call attention to information structure. Korean also has an accentual phrase that is defined by varied tonal patterns that do not map to equivalent patterns in English [252].

6.5.2. Results: Learners' GSB experience

To answer RQ2, "What features of the GSB did learners find useful or in need of improvement?", we interviewed learners following their immediate post- and delayed post-tests. Although both interviews included similar questions (Appendix B), delayed post-test interview included an additional question in which learners were asked to listen to two sentences from their pre- and post-test productions.

When learners were asked about the value of the pronunciation training and the ways they improved their speaking and pronunciation, they named several features. The feature that all learners except for one mentioned was fluency. Fourteen learners stated that GSB was helpful in making their speech sound more fluent and smoother. In fact, eight of these learners noticed how fluent they sounded after they listened to their pre- and post-test sentences during the delayed post-test interview. Learners' perceived improvement in fluency is also supported by our quantitative findings which showed a significant improvement between pre- and post-test. Learners (Excerpts 1 and 2) usually reported how 'choppy', 'cut' or 'slow' they sounded in their pre-test sentences whereas how 'quick' or 'smooth' they were in their post-test productions.

Excerpt 1:

Learner: actually this one is much more better than first.

Interviewer: okay, what is better about it?

Learner: this one, second one.

Interviewer: but what about it is better? What makes it better?

Learner: the first one is just uh how to say that, flow, ***the flow sounds like cut cut cut***.

Interviewer: okay ***so choppy***.

Learner: and the second one isn't, ***more better fluency***.

Excerpt 2:

Learner: uh, oh. I think ***my spoken English is more quick.***

Interviewer: more quick, okay.

Learner: yeah more quick and um I think ***my fluency is better***.

Connections between the words was something that some learners mentioned when they talked about fluency; they believed being able to connect words to each other instead of saying them one by one made their speech sound more smooth and more natural (See Excerpt 3). As a result, fluency and connected speech features were co-occurring topics learners touched on. Connected speech was something that some learners noticed clearly during their GSB training. They referred to the ‘linking’ between words and how they did not notice the connection between sounds before. They stated that they tried to use the GSB voice as a model to be able to produce the linking between words. One of the learners (Excerpt 4) said she knew about connected speech but she did not care about it until her practice with the GSB because she thought connected speech created a noticeable difference between her own pronunciation and that of the model voice. This awareness led her to care about something that she had not cared about before.

Excerpt 3:

Learner: so far more smooth and sounds more naturally.

Interviewer: Okay and anything else other than those?

Learner: mmm, I think just like I changed the way I speaked. Like well first before the training I said all words, speaking really clearly. And after the training like ***more connected and more smooth***.

Excerpt 4:

Interviewer: what are those things that you noticed with this model voice?

Learner: some something like when the ***words connected together very strongly.***

Interviewer: Okay so you have trouble with connected speech. Did you notice that before? Your, did you not know it before?

Learner: actually ***I didn't care about it before***. But I do care right now. After this,

Interviewer: why did it make you to care about it?

Learner: um, I think it's the ***big difference with my voice and model voice***.

Another pronunciation feature that was mentioned by most learners (n = 12) was intonation. Learners often stated how monotonous their speech was compared to the model voice and they did not have much 'ups and downs' or 'highs and lows' in their speech when they spoke English (Excerpt 5). Learners often explained the difference between their intonation and that of English by explaining how Korean works in general. They explained the change between 'high and low' as not something existing in Korean (Excerpt 6). When we asked learners if they would recommend practicing with the GSB to the others, one learner specifically commented on the benefit of hearing his own voice and how it helped with noticing the flow and intonation of the language: "*…it is a good opportunity to listen to your actual voice and then you can practice your pronunciation and you can actually be* ***aware of your voice or flows and intonation***".

Excerpt 5:

Interviewer: did you feel any changes during the training in your pronunciation? Anything you think you are doing better now?

Learner: oh I could some um realize that in terms of like um do question or some, so sometimes I need to ***tone down and tone up in terms of different question types***. That would be helpful to speak in English.

Interviewer: so you improved your intonation with those questions?

Learner: Mm-hmm. Yes I think so.

Excerpt 6:

Interviewer: okay, so how was yours different from the model voice?

Learner: um many ***Koreans pronunciation is not really high or low. just stable*** because Korean yeah, Korean language is kind of that. So um it was helpful to practice how to which part is good and ***what goes off and which part is goes down***.

Interviewer: Mm-hmm. So you started to think about those things?

Learner: Mm-hmm.

Learners also mentioned how GSB helped them notice the stress in individual words and sentences (n = 6). In addition, they mentioned how it helped with the improvement of certain sounds of English. However, the benefit of the GSB in improving segmentals was likely from practicing extensively for three weeks rather than hearing a voice similar to theirs. Extensive fluency practice may impact segmental improvement simply because of practice. Because the learners mostly talked about improvements in fluency and prosody, improvements in segmental quality may have been a side-effect of practice in general, and not connected to practicing with a golden speaker voice.

Three different exercise types were included in the design: say-listen-repeat, listen-repeat, and backward build-up exercises. Several learners (n = 9) stated their favorite exercise type was backward build-up because it gave them a chance to practice pronunciation in smaller chunks of speech. They could listen to the phrases in a sentence separately and this helped them in three ways: a) focus on parts they had more difficulties with, b) listen to words individually, c) focus on tones [i.e., intonation], and d) control the speed better (See Excerpt 7). One of the learners specifically mentioned the normal speed of sentences was too fast for him and backward build-up gave him the chance to practice things step by step, thus helping him with the flow of speech.

Excerpt 7:

> Learner: Mm, I think all of them is great for practicing, but mmm, big words made the small words helpful.
>
> Interviewer: okay, why?
>
> Learner: Mm, all because the two the big words I could ***follow the speed***, and I understand how to ***pronounce the tones***.

Excerpt 8:

> Learner: The difficult part was it was too fast. It was too fast to me and it's ***difficult to follow uh the full sentence***. And the easy part was, I don't know in the third practice, ***the step by step practice*** it was good to learn how to pronounce and how ***to make some flows***. Something like that.

In addition to the benefits for their pronunciation, most learners (n = 10) talked about the benefits of GSB for their listening skills—about how it helped them improve their listening or how it helped them listen critically and notice the problems in their pronunciation. Comments about listening improvement were similar to the comments about pronunciation in the sense that they performed better in hearing the connections between words or were better at catching up with the speed of speech. However, comments about listening critically showed how listening to a voice similar to one's own can help with perceiving the differences between one's self and the target pronunciation. One of the learners said "*I did not realize that there was a problem for me, but when I practicing it, I just realize that oh, model voice is correct and so yeah*."

Learners in the study were also asked about further development of the GSB. One of the topics they commented on frequently was the voice quality. They suggested the voice quality could be improved. Some students stated that the model voice in the GSB was not very much like them and some others said there were parts of some sentences that the voice was not clear or very easy to understand. One learner said "*Uh it was good but one thing, um the models voice sometimes like vague. A little noise, so sometimes I can, I could not figure it out. The clear sounds from model voice*." A similar comment from another learner was "*not clear sounds. So at the time I could not um figure out how to pronounce it like exactly because model voice sometimes very fast and sometimes vague*."

Another place for improvement lay in the design aspects of the GSB because some learners said having only three types of exercises or having a limited number of sentences to work with made their experience boring at times. Thus, adding more exercise types and sentences would be helpful. Another thing recommend by the learners was to be able to control the speed of speech because it was too fast for some learners and it made their effort to focus on pronunciation more challenging. Similar to that, learners also asked to practice individual words instead of only by phrases as in the backward build-up exercises. Suggestions about pronunciation improvement and support of visualization (such as including pictures and videos) were among the other recommendations for the improvement of the GSB.

6.6. Discussion

In this study, we looked at the effectiveness of an interactive CAPT program on what 15 Korean learners' thought about their learning experience with the program. When learners were asked for their opinions of their GSB experience, many learners reported how practice with the GSB helped them hear that their intonation and stress were different than the model voice and they believed they improved these features. Learners said the model voice allowed them to learn prosodic features of the language. While this is encouraging, it does not offer clear support for GSB; the use of any native-like voice prosody may have been equally or more effective. Because there was no control group, we cannot speak to this question.

One concern raised by learners was the speed of the model voice. It was initially too fast for many learners, even though it sounded like a normal speech rate for a native speaker. Fast speech can create problems for learners to catch the words and imitate speech [253]. However, research shows that it does not necessarily mean that slower speech will lead to greater comprehensibility. It is more important to have a speech rate which is similar to a learner's, or just slightly faster, rather than a slower one [8, 254].

The only feedback learners received in the training was the synthesized version of their own voice, and we hypothesized it would help learners in perceiving their pronunciation problems and pronouncing in a more target-like way. Some learners said the GSB model voice did not sound quite like them; for others, learners said they did not hear all words clearly in some sentences, which could be due to either synthesis quality or speed. The voice quality issue is indeed not a new problem, as other studies also showed some distortions in parts of their synthesized speech [255, 256]. But there is a possibility that the synthesized speech, either in quality or speed, may have limited what learners could pay attention to.

6.7. Conclusions

This study suggests that a CAPT program which utilizes feedback from a voice model can be helpful for the improvement of fluency (through attention to suprasegmental features of pronunciation) and for comprehensibility. Learners themselves reported an increase in their awareness for their use of intonation, stress, and connected speech in English. It may be that other types of feedback could be even more effective in promoting improvement.

7. CONCLUSIONS

7.1. Summary

In this dissertation, I propose three novel few-shot VC/FAC systems, and develop Golden Speaker Builder, a web application that applies FAC models to computer-assisted pronunciation training. In the first system, I focus on sparse representation based VC conversion approaches and develop a Cluster-Structured Sparse Representation (CSSR), which consists of two complementary components: a Cluster-Structured Dictionary Learning module that groups atoms in the dictionary into clusters, and a Cluster-Selective Objective Function that encourages each speech frame to be represented by atoms from a small number of clusters. Through a set of visualizations and analyses of CSSR, I illustrate that CSSR is able to learn phonetically meaningful sparse representations without any supervision, and the sparse representations are more speaker-independent compared against previous sparse-representation-based VC algorithms [30]. Compared to conventional GMM-based [11] methods, the proposed approach achieves superior acoustic quality and voice identity. More importantly, the proposed approach significantly reduces the number of utterances required during training (one minute of speech), avoiding the need to collect a large corpus from the users in real-world applications.

In the second system, I explore the effectiveness of neural networks in few-shot VC. Namely, I propose a VC approach based on seq2seq model, which only requires a few seconds of speech from the target speaker during inference and none during training (i.e., zero-shot learning). The seq2seq model has an encoder-decoder structure, and it transforms a sequence of Phonetic-Posteriorgram (PPG) to a sequence of speech features (i.e., Mel-spectrogram), conditioned on the corresponding speaker embedding. Additionally, to avoid the speaker-dependent information from PPG leaking into the voice conversions, I create an adversarial learning paradigm for training, which jointly trains the seq2seq model with an adversarial speaker classifier. The proposed system achieves significantly better acoustic quality and voice identity than sparse representation based few-shot VC algorithms. Meanwhile, with the need of only few seconds of speech from each speaker, the system can convert between any speakers, which is particularly fascinating for practical applications. Lastly, I verify the adversarial learning paradigm's effectiveness – when comparing it against a baseline without using an adversarial speaker classifier, there is a notable improvement in voice identity.

The third work generalizes the second work to FAC and builds a zero-shot FAC approach based on the seq2seq model. In this work, I use three independent models to disentangle the three aspects of an utterance: (1) a speaker-independent acoustic model to extract a linguistic content representation sequence (denoted as a bottleneck feature vector), (2) a speaker encoder to generate a speaker embedding, and (3) an accent encoder to obtain an accent embedding. To achieve FAC, I train a novel seq2seq model to synthesize speech using the linguistic content and accent representations from an L1 speaker along with the voice identity representation of an L2 speaker. To verify the effectiveness of the approach, I conduct experiments under two settings: standard FAC setting and zero-shot FAC setting. Under standard FAC setting, the proposed approach outperforms a state-of-the-art FAC model [27] in terms of accentedness, while retaining the acoustic quality and voice identity. Under zero-shot FAC setting, the proposed approach performs equally well as under the standard FAC setting. Additionally, similar to the second work, the system can produce FAC syntheses for arbitrary L2 speakers after training, with only few seconds of speech from each of them. These results are encouraging for pronunciation training applications, since it avoids the need to collect a large corpus for testing L2 speakers, improving user experience and pronunciation training efficiency.

In the fourth work, I develop Golden Speaker Builder, an interactive web application for computer-assisted pronunciation training. In Golden Speaker Builder, users can practice with the “golden speaker” voice with their own identity but a native accent, which has been shown to be more effective in pronunciation training [8]. To use Golden Speaker Builder, users first need to go through an enrollment process, which collects users’ speech and trains foreign accent conversion models. They can then select a set of sentences they would like to practice with, and the web application will synthesize the “golden speaker” voice of these sentences for them to practice with. Results reported from a user study suggest an increase in their awareness for their use of intonation, stress, and connected speech in English. As the first interface under the “golden speaker” paradigm, it beneficially promotes future research progress on computer-assisted pronunciation training and provides a more efficient way for L2 learners to improve their pronunciation.

7.2. Contributions

The main contributions of this dissertation can be summarized as,

- Constructed a novel few-shot voice conversion algorithm based on sparse representation, Cluster-Structured Sparse Representation (CSSR), which improves both acoustic quality and voice identity over previous GMM-based and sparse representation-based voice conversion algorithms.
- Verified that CSSR can achieve reasonable performance using only around one minute of training speech.
- Illustrated that CSSR is able to learn phonetically meaningful sparse representations without any supervision.
- Developed a zero-shot voice conversion algorithm based on seq2seq model that is able to convert between any speakers after training, with the need of only 3 second of speech from each speaker.
- Examined the effectiveness of adversarial learning paradigm during training for removing the speaker-dependent information from Phonetic PosteriorGrams.
- Proposed a zero-shot foreign accent conversion algorithm based on seq2seq model that can synthesize accent converted speech for arbitrary L2 learners, using only few seconds of speech from each of them.
- Verified that the zero-shot foreign accent conversion algorithm reduced accentedness over a state-of-the-art approach, and it performed equally well under standard and zero-shot foreign accent conversion settings.

- Developed Golden Speaker Builder, an interactive web application for pronunciation training, which applied foreign accent conversion algorithms to real-world scenarios.
- Demonstrated the effectiveness of Golden Speaker Builder in pronunciation training through a set of user studies.

7.3. Future work

7.3.1. Improvements on the first work

In the first work, I proposed CSSR for few-shot voice conversion. However, in the current CSSR system, the dictionary-learning algorithm (CSDL) to learn structured dictionaries is based on the Expectation-Maximization (EM) algorithm with *hard-decision* rules. A natural generalization of CSDL is the EM dictionary-learning algorithm [257], which computes the probability of each sample belonging to each cluster in the "E" step and updates each sub-dictionary using the samples with non-zero probabilities in the "M" step. The EM dictionary-learning algorithm avoids the hard decision but use a weighted sum of the samples during the "E" step, learning more representative dictionaries and thus improving voice conversion performance. The second substantial future work is to generalize CSSR to non-parallel training corpora. Currently, CSSR requires the training corpora to be parallel utterances (i.e., the source and target utterances have the same linguistic content). To relax this limitation, dynamic time warping algorithm used for time alignment can be replaced with the phonetic similarity based frame-paring technique proposed by Zhao et al. [24], which can directly generate frame pairs from non-parallel utterances. This modification makes it easier for the system to be used in practical purposes, since parallel corpora would not be required.

7.3.2. Improvements on the second work

In the second work, I proposed a zero-shot voice conversion algorithm based on seq2seq model, and an adversarial learning paradigm during model training. First, one valuable future direction is to improve the speaker transferability of the speaker recognition model to unseen speakers. To achieve this, the number of training speakers must be increased, so that the model can better capture the speaker space. Another issue that might impact the transferability to unseen speakers in the current system is overfitting. To address this issue, one can explore the use of regularization techniques such as dropout [174] and batch normalization [162] in speaker recognition models. It is also worthwhile to explore advanced loss functions that alleviate the overfitting for speaker recognition models, such as AM-Softmax loss [258] and Triplet loss [259]. Second, the current input sequences to the model are 40-dimensional mono-phone Phonetic PosteriorGrams (PPG) sequences, which do not contain information in co-articulations between phonemes, and therefore, degrade the intelligibility and naturalness of the voice conversions. As a result, a future work worth noting is to test the effectiveness of different input sequences (e.g., tri-phone PPG, bottle-neck feature extracted from the hidden layer of the acoustic model, spectral features) and compare them with the current system. Another potential future research line is to explore different adversarial learning schemes. Currently, adversarial learning is accomplished by jointly training the seq2seq model with an adversarial speaker classifier. However, this adversarial learning scheme is based on speaker classification, which may not work well when there are several speakers that are very similar. Therefore, it is worthwhile to examine other adversarial learning schemes (e.g., explicitly enforcing the distribution of the hidden representation from each speaker to be identical through an extra term in loss function [49]) and compare the effectiveness among them. A better adversarial learning scheme can further remove speaker-dependent information from the hidden representations, and therefore, improve the voice identity of voice conversions.

7.3.3. Improvements on the third work

Future works of the third work are two-fold. The first part will focus on improving the robustness and synthesis quality of the proposed approach. One possible future direction is to improve its robustness in generating long utterances. Currently, the system uses a location-sensitive attention mechanism [163] in the seq2seq model, which can fail when the utterances are too long [260]. To solve this problem, it can be replaced with alternative attention mechanisms, for instance, the Gaussian mixture attention mechanism [261], which has been shown to be robust in generating long utterances [260]. An additional potential improvement would be to add a second decoder for phonetic recognition (during training), following [262, 263]. Such an auxiliary decoder would guide the hidden representation produced by the encoder to preserve phonetic information, enforcing the synthesized speech to be phonetically reasonable and improving synthesis quality [262, 263]. Second, the current system still requires an L1 speaker's utterances as the reference, which requires extra human interference in practical applications. A possible solution to mitigate this issue is to use a text-to-speech synthesizer (e.g., Tacotron2 [88]) to generate the L1 utterances from text. As a result, it avoids the need for L1 speakers to produce learning materials. Instead, the system could synthesize accent conversions from any text.

7.3.4. Improvements on the fourth work

Future works focus on improving the Golden Speaker Builder (GSB) experience regarding both the quality of the golden speakers and design issues with the learning interface. First, the signal processing back-end in the first version of GSB system is based on Sparse, Anchor-Based Representation (SABR) [247, 248], which was the state-of-the-art few-shot accent conversion algorithm during the time that I was developing GSB. Afterwards, I upgraded the signal processing back-end PPG-GMM [24], as it outperforms SABR in both acoustic quality and voice identity. However, more recently, seq2seq models based accent conversion have been proven to boost synthesis performance significantly. As a result, a beneficial future work would be to replace the signal processing back-end from PPG-GMM to a more advanced accent conversion system, for example, the one presented in Chapter 5. By replacing the signal processing back-end, GSB will provide the learners with high-quality syntheses, possibly improving pronunciation training effectiveness. Learners should also be able to control the speech rate, making it slower or faster depending on their needs. It is likely that learners will use the speed control to slow down and increase rate in practice for different purposes. In addition, giving learners the ability to work on small chunks of speech by allowing them to select a region of interest of their speech waveform would also allow them to target a particular part of speech depending on their personal difficulties. The GSB learning interface can also be developed more with different exercises types (such as directed perception tasks), feedback that highlights individual problems, learning aids such as brief explanations about how to work on pronunciation features, and guidance on what features are most important in a particular sentence.

REFERENCES

[1] A. Kain and M. W. Macon, "Spectral voice conversion for text-to-speech synthesis," in *ICASSP*, 1998, pp. 285-288.

[2] T. Toda, M. Nakagiri, and K. Shikano, "Statistical voice conversion techniques for body-conducted unvoiced speech enhancement," *IEEE Transactions on Audio, Speech, and Language Processing,* vol. 20, pp. 2505-2517, 2012.

[3] D. E. Eslava and A. M. Bilbao, "Intra-lingual and cross-lingual voice conversion using harmonic plus stochastic models," *Barcelona, Spain: PhD Thesis, Universitat Politechnica de Catalunya,* 2008.

[4] D. Felps, H. Bortfeld, and R. Gutierrez-Osuna, "Foreign accent conversion in computer assisted pronunciation training," *Speech Communication,* vol. 51, pp. 920-932, 2009.

[5] K. Probst, Y. Ke, and M. Eskenazi, "Enhancing foreign language tutors - in search of the golden speaker," *Speech Communication,* vol. 37, pp. 161-173, 2002.

[6] S. Ding, C. Liberatore, S. Sonsaat, I. Lučić Rehman, A. Silpachai, G. Zhao, *et al.*, "Golden speaker builder - An interactive tool for pronunciation training," *Speech Communication,* vol. 115, pp. 51-66, 2019.

[7] R. Wang and J. Lu, "Investigation of golden speakers for second language learners from imitation preference perspective by voice modification," *Speech Communication,* vol. 53, pp. 175-184, February 2011 2011.

[8] K. Probst, Y. Ke, and M. Eskenazi, "Enhancing foreign language tutors–in search of the golden speaker," *Speech Communication*, vol. 37, pp. 161-173, 2002.

[9] K. Nagano and K. Ozawa, "English speech training using voice conversion," in *First International Conference on Spoken Language Processing*, 1990.

[10] Y. Stylianou, O. Cappé, and E. Moulines, "Continuous probabilistic transform for voice conversion," *IEEE Transactions on speech and audio processing,* vol. 6, pp. 131-142, 1998.

[11] T. Toda, A. W. Black, and K. Tokuda, "Voice conversion based on maximum-likelihood estimation of spectral parameter trajectory," *IEEE Transactions on Audio, Speech, and Language Processing,* vol. 15, pp. 2222-2235, 2007.

[12] L.-H. Chen, Z.-H. Ling, L.-J. Liu, and L.-R. Dai, "Voice conversion using deep neural networks with layer-wise generative training," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP),* vol. 22, pp. 1859-1872, 2014.

[13] S. Desai, A. W. Black, B. Yegnanarayana, and K. Prahallad, "Spectral mapping using artificial neural networks for voice conversion," *IEEE Transactions on Audio, Speech, and Language Processing,* vol. 18, pp. 954-964, 2010.

[14] T. Nakashika, T. Takiguchi, Y. Minami, T. Nakashika, T. Takiguchi, and Y. Minami, "Non-parallel training in voice conversion using an adaptive restricted boltzmann machine," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP),* vol. 24, pp. 2032-2045, 2016.

[15] F.-L. Xie, F. K. Soong, and H. Li, "A KL Divergence and DNN-Based Approach to Voice Conversion without Parallel Training Sentences," in *Interspeech*, 2016, pp. 287-291.

[16] G. Zhao, S. Sonsaat, J. Levis, E. Chukharev-Hudilainen, and R. Gutierrez-Osuna, "Accent Conversion Using Phonetic Posteriorgrams," *ICASSP,* 2018.

[17] H. Miyoshi, Y. Saito, S. Takamichi, and H. Saruwatari, "Voice conversion using sequence-to-sequence learning of context posterior probabilities," *arXiv preprint arXiv:1704.02360,* 2017.

[18] 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),* 2019, pp. 6820-6824.

[19] L. Sun, K. Li, H. Wang, S. Kang, and H. Meng, "Phonetic posteriorgrams for many-to-one voice conversion without parallel data training," in *2016 IEEE International Conference on Multimedia and Expo (ICME)*, 2016, pp. 1-6.

[20] Y. Zhou, X. Tian, H. Xu, R. K. Das, and H. Li, "Cross-lingual voice conversion with bilingual phonetic posteriorgram and average modeling," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 6790-6794.

[21] S. Aryal, D. Felps, and R. Gutierrez-Osuna, "Foreign accent conversion through voice morphing," in *INTERSPEECH*, 2013, pp. 3077-3081.

[22] M. Huckvale and K. Yanagisawa, "Spoken language conversion with accent morphing," 2007.

[23] S. Aryal and R. Gutierrez-Osuna, "Can voice conversion be used to reduce non-native accents?," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 7879-7883.

[24] G. Zhao and R. Gutierrez-Osuna, "Using Phonetic Posteriorgram Based Frame Pairing for Segmental Accent Conversion," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, pp. 1649-1660, 2019.

[25] S. Aryal and R. Gutierrez-Osuna, "Articulatory-based conversion of foreign accents with Deep Neural Networks," in *INTERSPEECH*, 2015.

[26] S. Aryal and R. Gutierrez-Osuna, "Reduction of non-native accents through statistical parametric articulatory synthesis," *The Journal of the Acoustical Society of America*, vol. 137, pp. 433-446, 2015.

[27] G. Zhao, S. Ding, and R. Gutierrez-Osuna, "Foreign Accent Conversion by Synthesizing Speech from Phonetic Posteriorgrams," *Proc. Interspeech 2019*, pp. 2843-2847, 2019.

[28] S. Liu, D. Wang, Y. Cao, L. Sun, X. Wu, S. Kang, *et al.*, "End-To-End Accent Conversion Without Using Native Utterances," in *ICASSP 2020-2020 IEEE*

International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2020, pp. 6289-6293.

[29] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, "Generalizing from a few examples: A survey on few-shot learning," *ACM Computing Surveys (CSUR)*, vol. 53, pp. 1-34, 2020.

[30] R. Takashima, T. Takiguchi, and Y. Ariki, "Exemplar-based voice conversion in noisy environment," in *SLT*, 2012, pp. 313-317.

[31] Z. Wu, E. S. Chng, and H. Li, "Exemplar-based voice conversion using joint nonnegative matrix factorization," *Multimedia Tools and Applications*, vol. 74, pp. 9943-9958, 2015.

[32] R. Aihara, T. Nakashika, T. Takiguchi, and Y. Ariki, "Voice conversion based on non-negative matrix factorization using phoneme-categorized dictionary," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 7894-7898.

[33] P. Kenny, "Bayesian speaker verification with heavy-tailed priors," in *Odyssey*, 2010.

[34] E. Variani, X. Lei, E. McDermott, I. L. Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *ICASSP*, 2014, pp. 4052-4056.

[35] S. H. Mohammadi and A. Kain, "A Voice Conversion Mapping Function Based on a Stacked Joint-Autoencoder," in *INTERSPEECH*, 2016, pp. 1647-1651.

[36] L. Sun, S. Kang, K. Li, and H. Meng, "Voice conversion using deep bidirectional long short-term memory based recurrent neural networks," in *ICASSP*, 2015, pp. 4869-4873.

[37] D. Erro, A. Moreno, and A. Bonafonte, "INCA algorithm for training voice conversion systems from nonparallel corpora," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, pp. 944-953, 2010.

[38] C. Liberatore, S. Aryal, Z. Wang, S. Polsley, and R. Gutierrez-Osuna, "SABR: Sparse, Anchor-Based Representation of the Speech Signal," in *INTERSPEECH*, 2015.

[39] C.-C. Hsu, H.-T. Hwang, Y.-C. Wu, Y. Tsao, and H.-M. Wang, "Voice conversion from non-parallel corpora using variational auto-encoder," in *APSIPA*, 2016, pp. 1-6.

[40] S. Ding and R. Gutierrez-Osuna, "Group Latent Embedding for Vector Quantized Variational Autoencoder in Non-Parallel Voice Conversion," *Proc. Interspeech 2019,* pp. 724-728, 2019.

[41] Y. Saito, Y. Ijima, K. Nishida, and S. Takamichi, "Non-parallel voice conversion using variational autoencoders conditioned by phonetic posteriorgrams and d-vectors," in *ICASSP*, 2018, pp. 5274-5278.

[42] W.-N. Hsu, Y. Zhang, and J. Glass, "Unsupervised learning of disentangled and interpretable representations from sequential data," in *NIPS*, 2017, pp. 1878-1889.

[43] H. Kameoka, T. Kaneko, K. Tanaka, and N. Hojo, "ACVAE-VC: Non-parallel many-to-many voice conversion with auxiliary classifier variational autoencoder," *arXiv preprint arXiv:1808.05092,* 2018.

[44] W.-C. Huang, Y.-C. Wu, H.-T. Hwang, P. L. Tobing, T. Hayashi, K. Kobayashi, *et al.*, "Refined WaveNet Vocoder for Variational Autoencoder Based Voice Conversion," *arXiv preprint arXiv:1811.11078,* 2018.

[45] S. Liu, J. Zhong, L. Sun, X. Wu, X. Liu, and H. Meng, "Voice Conversion Across Arbitrary Speakers Based on a Single Target-Speaker Utterance," *Proc. Interspeech 2018,* pp. 496-500, 2018.

[46] S. H. Mohammadi and T. Kim, "One-Shot Voice Conversion with Disentangled Representations by Leveraging Phonetic Posteriorgrams," *Proc. Interspeech 2019,* pp. 704-708, 2019.

[47] H. Lu, Z. Wu, D. Dai, R. Li, S. Kang, J. Jia, *et al.*, "One-shot Voice Conversion with Global Speaker Embeddings," *Proc. Interspeech 2019,* pp. 669-673, 2019.

[48] T. Kaneko, H. Kameoka, K. Hiramatsu, and K. Kashino, "Sequence-to-sequence voice conversion with similarity metric learned using generative adversarial networks," in *Proc. Interspeech*, 2017, pp. 1283-1287.

[49] J. Zhang, Z. Ling, and L.-R. Dai, "Non-parallel sequence-to-sequence voice conversion with disentangled linguistic and speaker representations," *IEEE/ACM Transactions on Audio, Speech, and Language Processing,* 2019.

[50] C.-C. Hsu, H.-T. Hwang, Y.-C. Wu, Y. Tsao, and H.-M. Wang, "Voice conversion from unaligned corpora using variational autoencoding wasserstein generative adversarial networks," *arXiv preprint arXiv:1704.00849,* 2017.

[51] A. van den Oord and O. Vinyals, "Neural discrete representation learning," in *NIPS*, 2017, pp. 6306-6315.

[52] D. Felps, C. Geng, and R. Gutierrez-Osuna, "Foreign Accent Conversion Through Concatenative Synthesis in the Articulatory Domain," *IEEE Transactions on Audio, Speech, and Language Processing,* vol. 20, pp. 2301-2312, 2012.

[53] S. Aryal and R. Gutierrez-Osuna, "Data driven articulatory synthesis with deep neural networks," *Computer Speech & Language,* vol. 36, pp. 260-273, 2016.

[54] M. Brand, "Voice puppetry," in *26th Annual Conference on Computer Graphics and Interactive Techniques*, 1999, pp. 21-28.

[55] B. Denby and M. Stone, "Speech synthesis from real time ultrasound images of the tongue," in *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2004, pp. I-685.

[56] R. Mumtaz, S. Preuß, C. Neuschaefer-Rube, C. Hey, R. Sader, and P. Birkholz, "Tongue contour reconstruction from optical and electrical palatography," *IEEE Signal Processing Letters,* vol. 21, pp. 658-662, 2014.

[57] A. Toutios, T. Sorensen, K. Somandepalli, R. Alexander, and S. S. Narayanan, "Articulatory synthesis based on real-time magnetic resonance imaging data," in *Interspeech*, 2016, pp. 1492-1496.

[58] C. H. Lampert, H. Nickisch, and S. Harmeling, "Learning to detect unseen object classes by between-class attribute transfer," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 951-958.

[59] H. Qi, M. Brown, and D. G. Lowe, "Low-shot learning with imprinted weights," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 5822-5830.

[60] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap, "Meta-learning with memory-augmented neural networks," in *International conference on machine learning*, 2016, pp. 1842-1850.

[61] Y. Zhang, H. Tang, and K. Jia, "Fine-grained visual categorization using meta-learning optimization with sample selection of auxiliary data," in *Proceedings of the european conference on computer vision (ECCV)*, 2018, pp. 233-248.

[62] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, *et al.*, "Specaugment: A simple data augmentation method for automatic speech recognition," *arXiv preprint arXiv:1904.08779,* 2019.

[63] T. Pfister, J. Charles, and A. Zisserman, "Domain-adaptive discriminative one-shot learning of gestures," in *European Conference on Computer Vision*, 2014, pp. 814-829.

[64] S. Motiian, Q. Jones, S. Iranmanesh, and G. Doretto, "Few-shot adversarial domain adaptation," in *Advances in Neural Information Processing Systems*, 2017, pp. 6670-6680.

[65] Z. Hu, X. Li, C. Tu, Z. Liu, and M. Sun, "Few-shot charge prediction with discriminative legal attributes," in *Proceedings of the 27th International Conference on Computational Linguistics*, 2018, pp. 487-498.

[66] E. Triantafillou, R. Zemel, and R. Urtasun, "Few-shot learning through an information retrieval lens," *Advances in Neural Information Processing Systems,* vol. 30, pp. 2255-2265, 2017.

[67] M. Fink, "Object classification from a single example utilizing class relevance metrics," *Advances in neural information processing systems,* vol. 17, pp. 449-456, 2004.

[68] L. Yan, Y. Zheng, and J. Cao, "Few-shot learning for short text classification," *Multimedia Tools and Applications,* vol. 77, pp. 29799-29810, 2018.

[69] S. Caelles, K.-K. Maninis, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, and L. Van Gool, "One-shot video object segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 221-230.

[70] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," *arXiv preprint arXiv:1703.03400,* 2017.

[71] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM,* vol. 60, pp. 84-90, 2017.

[72] H. Lee, A. Battle, R. Raina, and A. Y. Ng, "Efficient sparse coding algorithms," in *Advances in neural information processing systems*, 2007, pp. 801-808.

[73] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM review,* vol. 43, pp. 129-159, 2001.

[74] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Transactions on Image processing,* vol. 15, pp. 3736-3745, 2006.

[75] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online dictionary learning for sparse coding," in *Proceedings of the 26th annual international conference on machine learning*, 2009, pp. 689-696.

[76] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng, "Self-taught learning: transfer learning from unlabeled data," in *Proceedings of the 24th international conference on Machine learning*, 2007, pp. 759-766.

[77] S. Bengio, F. Pereira, Y. Singer, and D. Strelow, "Group sparse coding," in *Advances in neural information processing systems*, 2009, pp. 82-89.

[78] R. Jenatton, J. Mairal, G. Obozinski, and F. R. Bach, "Proximal Methods for Sparse Hierarchical Dictionary Learning," in *ICML*, 2010, pp. 487-494.

[79] Z. Szabó, B. Póczos, and A. Lőrincz, "Online group-structured dictionary learning," in *CVPR*, 2011, pp. 2865-2872.

[80] Y. Sun, Y. Quan, and J. Fu, "Sparse coding and dictionary learning with class-specific group sparsity," *Neural Computing and Applications*, vol. 30, pp. 1265-1275, 2018.

[81] Z. He, L. Liu, R. Deng, and Y. Shen, "Low-rank group inspired dictionary learning for hyperspectral image classification," *Signal Processing*, vol. 120, pp. 209-221, 2016.

[82] H.-T. T. Duong, Q.-C. Nguyen, C.-P. Nguyen, T.-H. Tran, and N. Q. Duong, "Speech enhancement based on nonnegative matrix factorization with mixed group sparsity constraint," in *Proceedings of the Sixth International Symposium on Information and Communication Technology*, 2015, pp. 247-251.

[83] A. Jukić, T. van Waterschoot, T. Gerkmann, and S. Doclo, "Group sparsity for MIMO speech dereverberation," in *Applications of Signal Processing to Audio and Acoustics (WASPAA), 2015 IEEE Workshop on*, 2015, pp. 1-5.

[84] A. Hurmalainen, R. Saeidi, and T. Virtanen, "Group sparsity for speaker identity discrimination in factorisation-based speech recognition," 2012.

[85] D. L. Sun and G. J. Mysore, "Universal speech models for speaker independent single channel source separation," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, 2013, pp. 141-145.

[86] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104-3112.

[87] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, *et al.*, "Tacotron: Towards end-to-end speech synthesis," *arXiv preprint arXiv:1703.10135,* 2017.

[88] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, *et al.*, "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 4779-4783.

[89] Y. Jia, Y. Zhang, R. Weiss, Q. Wang, J. Shen, F. Ren, *et al.*, "Transfer learning from speaker verification to multispeaker text-to-speech synthesis," in *Advances in neural information processing systems*, 2018, pp. 4480-4490.

[90] J. Martinez, H. Perez, E. Escamilla, and M. M. Suzuki, "Speaker recognition using Mel frequency Cepstral Coefficients (MFCC) and Vector quantization (VQ) techniques," presented at the Proc. 22nd International Conference on Electrical Communications and Computers, 2012.

[91] V. Tiwari, "MFCC and its applications in speaker recognition," *International journal on emerging technologies,* vol. 1, pp. 19-22, 2010.

[92] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-End Factor Analysis for Speaker Verification," *IEEE Transactions on Audio, Speech, and Language Processing,* vol. 19, pp. 788-798, 2011.

[93] E. Variani, X. Lei, E. McDermott, I. L. Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," presented at the Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2014.

[94] S. J. Prince and J. H. Elder, "Probabilistic linear discriminant analysis for inferences about identity," presented at the Proc. International Conference on Computer Vision, 2007.

[95] P. Matějka, O. Glembek, F. Castaldo, M. J. Alam, O. Plchot, P. Kenny, *et al.*, "Full-covariance UBM and heavy-tailed PLDA in i-vector speaker verification," presented at the Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2011.

[96] S. Cumani, O. Plchot, and P. Laface, "Probabilistic linear discriminant analysis of i-vector posterior distributions," presented at the Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2013.

[97] Y. K. Muthusamy, E. Barnard, and R. A. J. I. S. P. M. Cole, "Reviewing automatic language identification," vol. 11, pp. 33-41, 1994.

[98] D. Martinez, O. Plchot, L. Burget, O. Glembek, and P. Matějka, "Language recognition in ivectors space," presented at the Proc. Interspeech, 2011.

[99] F. Biadsy, J. Hirschberg, and D. P. Ellis, "Dialect and accent recognition using phonetic-segmentation supervectors," presented at the Proc. Twelfth Annual Conference of the International Speech Communication Association, 2011.

[100] F. Biadsy, "Automatic dialect and accent recognition and its application to speech recognition," Columbia University, 2011.

[101] A. Jain, M. Upreti, and P. Jyothi, "Improved Accented Speech Recognition Using Accent Embeddings and Multi-task Learning," presented at the Proc. Interspeech, 2018.

[102] X. Yang, K. Audhkhasi, A. Rosenberg, S. Thomas, B. Ramabhadran, and M. Hasegawa-Johnson, "Joint modeling of accents and acoustics for multi-accent speech recognition," presented at the Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2018.

[103] G. Brown, "Exploring forensic accent recognition using the Y-ACCDIST system," presented at the Proc. 16th Speech Science and Technology Conference, 2016.

[104] P. Altbach, "Perspectives on internationalizing higher education," *International Higher Education*, 2015.

[105] H.-K. Koh, "Trends in international student flows to the United States," *International Higher Education,* 2015.

[106] T. K. Mackey and B. A. Liang, "Rebalancing brain drain: Exploring resource reallocation to address health worker migration and promote global health," *Health Policy,* vol. 107, pp. 66-73, 2012.

[107] W. R. Kerr, "U.S. High-Skilled Immigration, Innovation, and Entrepreneurship: Empirical Approaches and Evidence," N. B. o. E. Research, Ed., ed, 2013.

[108] K. Tajima, R. Port, and J. Dalby, "Effects of temporal correction on intelligibility of foreign-accented English," *Journal of Phonetics,* vol. 25, pp. 1-24, 1997.

[109] J. A. Foote, A. K. Holtby, and T. M. Derwing, "Survey of the teaching of pronunciation in adult ESL programs in Canada, 2010," *TESL Canada Journal,* vol. 29, pp. 1-22, 2012.

[110] J. M. Murphy, "Intelligible, comprehensible, non-native models in ESL/EFL pronunciation teaching," *System,* vol. 42, pp. 258-269, 2014.

[111] K. Probst, Y. Ke, and M. Eskenazi, "Enhancing foreign language tutors - In search of the golden speaker," *Speech Communication,* vol. 37, pp. 161-173, 2002.

[112] M. P. Bissiri, H. R. Pfitzinger, and H. G. Tillmann, "Lexical Stress Training of German Compounds for Italian Speakers by means of Resynthesis and Emphasis," presented at the Proceedings of the 11th Australian International Conference on Speech Science & Technology, University of Auckland, New Zealand, 2006.

[113] M. P. Bissiri and H. R. Pfitzinger, "Italian speakers learn lexical stress of German morphologically complex words," *Speech Communication,* vol. 51, pp. 933-947, 2009.

[114] E. Pellegrino and D. Vigliano, "Self-imitation in prosody training: A study on Japanese learners of Italian," presented at the Proceedings of the Workshop on

Speech and Language Technology in Education (SLaTE), Leipzig, Germany, 2015.

[115] A. De Meo, M. Vitale, M. Pettorino, F. Cutugno, and A. Origlia, "Imitation/self-imitation in computer-assisted prosody training for Chinese learners of L2 Italian," presented at the Proceedings of the 4th Pronunciation in Second Language Learning and Teaching Conference, 2012.

[116] E. Moulines and F. Charpentier, "Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones," *Speech communication,* vol. 9, pp. 453-467, 1990.

[117] A. Brown, "Functional load and the teaching of pronunciation," *TESOL quarterly,* vol. 22, pp. 593-606, 1988.

[118] J. C. Catford, "Phonetics and the teaching of pronunciation: A systemic description of English phonology," *Current perspectives on pronunciation: Practices anchored in theory,* pp. 87-100, 1987.

[119] T. M. Derwing, R. I. Thomson, and M. J. Munro, "English pronunciation and fluency development in Mandarin and Slavic speakers," *System,* vol. 34, pp. 183-193, 2006.

[120] B. W. Zielinski, "The listener: No longer the silent partner in reduced intelligibility," *System,* vol. 36, pp. 69-84, 2008.

[121] A. Cutler, "Lexical stress in English pronunciation," *The handbook of English pronunciation,* pp. 106-124, 2015.

[122] Z. Wu and H. Li, "Voice conversion and spoofing attack on speaker verification systems," in *APSIPA*, 2013, pp. 1-9.

[123] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological),* pp. 267-288, 1996.

[124] C. Ding, D. Zhou, X. He, and H. Zha, "R 1-PCA: rotational invariant L 1-norm principal component analysis for robust subspace factorization," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 281-288.

[125] J. Kominek and A. W. Black, "The CMU Arctic speech databases," in *Fifth ISCA Workshop on Speech Synthesis*, 2004.

[126] Y. Chen, M. Chu, E. Chang, J. Liu, and R. Liu, "Voice conversion with smoothed GMM and MAP adaptation," in *Eighth European Conference on Speech Communication and Technology*, 2003.

[127] F. Fang, J. Yamagishi, I. Echizen, and J. Lorenzo-Trueba, "High-quality nonparallel voice conversion based on cycle-consistent adversarial network," *arXiv preprint arXiv:1804.00425*, 2018.

[128] E. Helander, T. Virtanen, J. Nurminen, and M. Gabbouj, "Voice conversion using partial least squares regression," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, pp. 912-921, 2010.

[129] T. Masuko, K. Tokuda, T. Kobayashi, and S. Imai, "Speech synthesis using HMMs with dynamic features," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 1996, pp. 389-392.

[130] G. Zhao and R. Gutierrez-Osuna, "Exemplar selection methods in voice conversion," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 5525-5529.

[131] S.-W. Fu, P.-C. Li, Y.-H. Lai, C.-C. Yang, L.-C. Hsieh, and Y. Tsao, "Joint dictionary learning-based non-negative matrix factorization for voice conversion to improve speech intelligibility after oral surgery," *IEEE Transactions on Biomedical Engineering*, vol. 64, pp. 2584-2594, 2017.

[132] R. Aihara, T. Takiguchi, and Y. Ariki, "Parallel Dictionary Learning for Voice Conversion Using Discriminative Graph-Embedded Non-Negative Matrix Factorization," in *INTERSPEECH*, 2016, pp. 292-296.

[133] B. Sisman, H. Li, and K. C. Tan, "Sparse Representation of Phonetic Features for Voice Conversion with and without Parallel Data," in *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2017.

[134] Z. Wu, T. Virtanen, E. S. Chng, and H. Li, "Exemplar-based sparse representation with residual compensation for voice conversion," *IEEE/ACM Transactions on Audio, Speech, and Language Processing,* vol. 22, pp. 1506-1521, 2014.

[135] C. Liberatore, G. Zhao, and R. Gutierrez-Osuna, "Voice Conversion through Residual Warping in a Sparse, Anchor-Based Representation of Speech," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* 2018.

[136] Y.-C. Wu, H.-T. Hwang, C.-C. Hsu, Y. Tsao, and H.-M. Wang, "Locally Linear Embedding for Exemplar-Based Spectral Conversion," in *INTERSPEECH,* 2016, pp. 1652-1656.

[137] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *Journal of the Royal Statistical Society: Series B (Statistical Methodology),* vol. 68, pp. 49-67, 2006.

[138] L. Jacob, G. Obozinski, and J.-P. Vert, "Group lasso with overlap and graph lasso," in *Proceedings of the 26th annual international conference on machine learning*, 2009, pp. 433-440.

[139] S. Kim and E. P. Xing, "Tree-guided group lasso for multi-task regression with structured sparsity," in *ICML*, 2010, p. 1.

[140] P. Zhao, G. Rocha, and B. Yu, "The composite absolute penalties family for grouped and hierarchical variable selection," *The Annals of Statistics,* vol. 37, pp. 3468-3497, 2009.

[141] S. Mosci, L. Rosasco, M. Santoro, A. Verri, and S. Villa, "Solving structured sparsity regularization with proximal methods," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 2010, pp. 418-433.

[142] R. Aihara, T. Takiguchi, and Y. Ariki, "Activity-mapping non-negative matrix factorization for exemplar-based voice conversion," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, 2015, pp. 4899-4903.

[143] S. B. Cohen and N. A. Smith, "Viterbi training for PCFGs: Hardness results and competitiveness of uniform initialization," in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, 2010, pp. 1502-1511.

[144] R. Samdani, M.-W. Chang, and D. Roth, "Unified expectation maximization," in *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2012, pp. 688-698.

[145] D. J. MacKay, *Information theory, inference and learning algorithms*: Cambridge university press, 2003.

[146] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability,* 1967, pp. 281-297.

[147] Y.-C. Chen, V. M. Patel, J. K. Pillai, R. Chellappa, and P. J. Phillips, "Dictionary learning from ambiguously labeled data," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, 2013, pp. 353-360.

[148] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *The Annals of statistics*, vol. 32, pp. 407-499, 2004.

[149] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *Journal of Machine Learning Research*, vol. 11, pp. 19-60, 2010.

[150] S. Ding, G. Zhao, C. Liberatore, and R. Gutierrez-Osuna, "Improving Sparse Representations in Exemplar-Based Voice Conversion with a Phoneme-Selective Objective Function," *Proc. Interspeech 2018,* pp. 476-480, 2018.

[151] 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, pp. 1877-1884, 2016.

[152] M. Morise, "D4C, a band-aperiodicity estimator for high-quality speech synthesis," *Speech Communication,* vol. 84, 2016.

[153] D. J. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series," in *KDD workshop*, 1994, pp. 359-370.

[154] S. Ding, C. Liberatore, and R. Gutierrez-Osuna, "Learning Structured Dictionaries for Exemplar-based Voice Conversion," *Proc. Interspeech 2018*, pp. 481-485, 2018.

[155] D. Felps and R. Gutierrez-Osuna, "Developing objective measures of foreign-accent conversion," *IEEE Transactions on Audio, Speech, and Language Processing,* vol. 18, 2010.

[156] S. Buchholz and J. Latorre, "Crowdsourcing preference tests, and how to detect cheating," in *INTERSPEECH*, 2011.

[157] S. Ding, G. Zhao, C. Liberatore, and R. Gutierrez-Osuna, "Learning Structured Sparse Representations for Voice Conversion," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 343-354, 2019.

[158] C. Veaux, J. Yamagishi, and K. MacDonald, "CSTR VCTK Corpus: English Multi-speaker Corpus for CSTR Voice Cloning Toolkit," 2017.

[159] W.-C. Huang, H. Luo, H.-T. Hwang, C.-C. Lo, Y.-H. Peng, Y. Tsao*, et al.*, "Unsupervised Representation Disentanglement using Cross Domain Features and Adversarial Learning in Variational Autoencoder based Voice Conversion," *arXiv preprint arXiv:2001.07849,* 2020.

[160] L. Wan, Q. Wang, A. Papir, and I. L. Moreno, "Generalized end-to-end loss for speaker verification," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 4879-4883.

[161] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* 2016, pp. 4960-4964.

[162] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167,* 2015.

[163] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Advances in neural information processing systems*, 2015, pp. 577-585.

[164] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," *arXiv preprint arXiv:1409.7495,* 2014.

[165] Z. Meng, J. Li, Y. Gong, and B.-H. Juang, "Adversarial teacher-student learning for unsupervised domain adaptation," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5949-5953.

[166] X. Zhang, J. Trmal, D. Povey, and S. Khudanpur, "Improving deep neural network acoustic models using generalized maxout networks," in *2014 IEEE international conference on acoustics, speech and signal processing (ICASSP),* 2014, pp. 215-219.

[167] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel*, et al.,* "The Kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding,* 2011.

[168] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* 2015, pp. 5206-5210.

[169] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep speaker recognition," *arXiv preprint arXiv:1806.05622,* 2018.

[170] N. Kalchbrenner, E. Elsen, K. Simonyan, S. Noury, N. Casagrande, E. Lockhart, *et al.*, "Efficient neural audio synthesis," *arXiv preprint arXiv:1802.08435*, 2018.

[171] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito*, et al.*, "Automatic differentiation in pytorch," 2017.

[172] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467,* 2016.

[173] R. Kubichek, "Mel-cepstral distance measure for objective speech quality assessment," in *IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, 1993, pp. 125-128.

[174] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research,* vol. 15, pp. 1929-1958, 2014.

[175] O. Turk and L. M. Arslan, "Subband based voice conversion," in *Seventh International Conference on Spoken Language Processing*, 2002.

[176] L. Sun, H. Wang, S. Kang, K. Li, and H. M. Meng, "Personalized, cross-lingual TTS using phonetic posteriorgrams," in *Interspeech*, 2016, pp. 322-326.

[177] Y. Oshima, S. Takamichi, T. Toda, G. Neubig, S. Sakti, and S. Nakamura, "Non-native speech synthesis preserving speaker individuality based on partial correction of prosodic and phonetic characteristics," in *Interspeech*, 2015, pp. 299-303.

[178] F. Biadsy, R. J. Weiss, P. J. Moreno, D. Kanvesky, and Y. Jia, "Parrotron: An end-to-end speech-to-speech conversion model and its applications to hearing-impaired speech and speech separation," in *Interspeech*, 2019, pp. 4115-4119.

[179] S. O. Arik, J. Chen, K. Peng, W. Ping, and Y. Zhou, "Neural voice cloning with a few samples," *arXiv preprint arXiv:1802.06006,* 2018.

[180] A. Das, G. Zhao, J. Levis, E. Chukharev-Hudilainen, and R. Gutierrez-Osuna, "Understanding the Effect of Voice Quality and Accent on Talker Similarity," 2020.

[181] M. Leikin, R. Ibrahim, Z. Eviatar, and S. Sapir, "Listening with an accent: Speech perception in a second language by late bilinguals," *Journal of psycholinguistic research,* vol. 38, p. 447, 2009.

[182] R. C. Major, S. F. Fitzmaurice, F. Bunta, and C. Balasubramanian, "The effects of nonnative accents on listening comprehension: Implications for ESL assessment," *TESOL quarterly,* vol. 36, pp. 173-190, 2002.

[183] M. Van Heugten, C. Bergmann, and A. Cristia, "The effects of talker voice and accent on young children's speech perception," in *Individual differences in speech production and perception,* ed: Peter Lang, 2015, pp. 57-88.

[184] A. Cristia, A. Seidl, C. Vaughn, R. Schmale, A. Bradlow, and C. Floccia, "Linguistic processing of accented speech across the lifespan," *Frontiers in psychology,* vol. 3, p. 479, 2012.

[185] S. Ding, G. Zhao, and R. Gutierrez-Osuna, "Improving the Speaker Identity of Non-Parallel Many-to-Many Voice Conversion with Adversarial Speaker Recognition," *Proc. Interspeech 2020,* pp. 776-780, 2020.

[186] D. Griffin and J. Lim, "Signal estimation from modified short-time Fourier transform," *IEEE Transactions on Acoustics, Speech, and Signal Processing,* vol. 32, pp. 236-243, 1984.

[187] A. Tamamori, T. Hayashi, K. Kobayashi, K. Takeda, and T. Toda, "Speaker-Dependent WaveNet Vocoder," in *INTERSPEECH,* 2017, pp. 1118-1122.

[188] G. Zhao, S. Sonsaat, A. O. Silpachai, I. Lucic, E. Chukharev-Khudilaynen, J. Levis*, et al.*, "L2-ARCTIC: A Non-Native English Speech Corpus," *Perception Sensing Instrumentation Lab,* 2018.

[189] S. H. Mohammadi and A. Kain, "An overview of voice conversion systems," *Speech Communication,* 2017.

[190] B. Sisman, J. Yamagishi, S. King, and H. Li, "An Overview of Voice Conversion and its Challenges: From Statistical Modeling to Deep Learning," *arXiv preprint arXiv:2008.03648,* 2020.

[191] D. Povey, G. Cheng, Y. Wang, K. Li, H. Xu, and S. Khudanpur, "Semi-orthogonal low-rank matrix factorization for deep neural networks," in *Interspeech*, 2018, pp. 3743-3747.

[192] V. Peddinti, D. Povey, and S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *Interspeech*, 2015, pp. 3214-3218.

[193] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770-778.

[194] R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural computation,* vol. 1, pp. 270-280, 1989.

[195] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: a large-scale speaker identification dataset," *arXiv preprint arXiv:1706.08612,* 2017.

[196] I. Loshchilov and F. Hutter, "Sgdr: Stochastic gradient descent with warm restarts," *arXiv preprint arXiv:1608.03983,* 2016.

[197] S. Weinberger, "Speech accent archive," *George Mason University,* 2015.

[198] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of machine learning research,* vol. 9, 2008.

[199] G. E. Henter, J. Lorenzo-Trueba, X. Wang, M. Kondo, and J. Yamagishi, "Cyborg speech: Deep multilingual speech synthesis for generating segmental foreign accent with natural prosody," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* 2018, pp. 4799-4803.

[200] M. J. Munro and T. M. Derwing, "Foreign accent, comprehensibility, and intelligibility in the speech of second language learners," *Language learning,* vol. 45, pp. 73-97, 1995.

[201] S. Buchholz and J. Latorre, "Crowdsourcing preference tests, and how to detect cheating," in *Interspeech*, 2011, pp. 3053-3056.

[202] C. Wang, M. Rivière, A. Lee, A. Wu, C. Talnikar, D. Haziza*, et al.*, "VoxPopuli: A Large-Scale Multilingual Speech Corpus for Representation Learning, Semi-Supervised Learning and Interpretation," *arXiv preprint arXiv:2101.00390,* 2021.

[203] S. Ding, C. Liberatore, S. Sonsaat, I. Lučić, A. Silpachai, G. Zhao*, et al.*, "Golden speaker builder–An interactive tool for pronunciation training," *Speech Communication,* vol. 115, pp. 51-66, 2019.

[204] R. Hincks, "Speech technologies for pronunciation feedback and evaluation," *ReCALL,* vol. 15, pp. 3-20, 2003.

[205] J. Burgess and S. Spencer, "Phonology and pronunciation in integrated language teaching and teacher education," *System,* vol. 28, pp. 191-215, 2000.

[206] G. Couper, "Teacher Cognition of Pronunciation Teaching: Teachers' Concerns and Issues," *TESOL Quarterly,* vol. 51, pp. 820-843, 2017.

[207] S. MacDonald, "Pronunciation-views and practices of reluctant teachers," 2002.

[208] J. Levis and S. Sonsaat, "Pronunciation in the CLT era," *The Routledge handbook of English pronunciation,* pp. 267-283, 2017.

[209] P. Warren, I. Elgort, and D. Crabbe, "Comprehensibility and prosody ratings for pronunciation software development," 2009.

[210] K. Egan and S. LaRocca, "Speech recognition in language learning: A must," *Proceedings of InSTILL 2000,* pp. 4-9, 2000.

[211] M. Eskenazi, "Using a computer in foreign language pronunciation training: What advantages?," *Calico Journal,* pp. 447-469, 1999.

[212] A. Mackey and J.-Y. Choi, "Review of Tripleplay Plus! English," 1998.

[213] M. E. Rypa and P. Price, "VILTS: A tale of two technologies," *CALICO journal,* pp. 385-404, 1999.

[214] D. M. Hardison, "Generalization of computer assisted prosody training: Quantitative and qualitative findings," 2004.

[215] J. Levis, "Computer technology in teaching and researching pronunciation," *Annual Review of Applied Linguistics,* vol. 27, pp. 184-202, 2007.

[216] H. Bliss, S. Bird, P. A. Cooper, S. Burton, and B. Gick, "Seeing Speech: Ultrasound-based Multimedia Resources for Pronunciation Learning in Indigenous Languages," 2018.

[217] T. A. Barriuso and R. Hayes-Harb, "High Variability Phonetic Training as a Bridge from Research to Practice," *CATESOL Journal,* vol. 30, pp. 177-194, 2018.

[218] R. I. Thomson, "Computer assisted pronunciation training: Targeting second language vowel perception improves pronunciation," *Calico Journal*, vol. 28, p. 744, 2011.

[219] R. I. Thomson, "Improving L2 listeners' perception of English vowels: A computer-mediated approach," *Language Learning,* vol. 62, pp. 1231-1258, 2012.

[220] A. Mackey and R. Abbuhl, "Input and interaction," *Mind and Context in Adult Second Language Acquisition: Methods, Theory and Practice,* pp. 207-233, 2005.

[221] M. Swain and S. Lapkin, "Problems in output and the cognitive processes they generate: A step towards second language learning," *Applied linguistics*, vol. 16, pp. 371-391, 1995.

[222] M. Swain, "The output hypothesis and beyond: Mediating acquisition through collaborative dialogue," *Sociocultural theory and second language learning,* vol. 97, p. 114, 2000.

[223] T. Heift, "Corrective feedback and learner uptake in CALL," *ReCALL,* vol. 16, pp. 416-431, 2004.

[224] B. Mak, M. Siu, M. Ng, Y.-C. Tam, Y.-C. Chan, K.-W. Chan, *et al.*, "PLASER: pronunciation learning via automatic speech recognition," in *Proceedings of the HLT-NAACL 03 workshop on Building educational applications using natural language processing-Volume 2,* 2003, pp. 23-29.

[225] M. Eskenazi, "An overview of spoken language technology for education," *Speech Communication,* vol. 51, pp. 832-844, 2009.

[226] K. Hirose, F. Gendrin, and N. Minematsu, "A pronunciation training system for Japanese lexical accents with corrective feedback in learner's voice," presented at the Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH), 2003.

[227] M. Peabody and S. Seneff, "Towards Automatic Tone Correction in Non-native Mandarin," in *Chinese Spoken Language Processing,* ed, 2006, pp. 602-613.

[228] K. Nagano and K. Ozawa, "English Speech Training Using Voice Conversion," presented at the First International Conference on Spoken Language Processing (ICSLP), Kobe, Japan, 1990.

[229] Q. Yan, S. Vaseghi, D. Rentzos, and H. Ching-Hsiang, "Analysis and Synthesis of Formant Spaces of British, Australian, and American Accents," *IEEE Transactions on Audio, Speech, and Language Processing,* vol. 15, pp. 676-689, 2007.

[230] M. Huckvale and K. Yanagisawa, "Spoken language conversion with accent morphing," presented at the Proceedings of ISCA Speech Synthesis Workshop, Bonn, Germany, 2007.

[231] T. M. Derwing and M. J. Munro, *Pronunciation fundamentals: Evidence-based perspectives for L2 teaching and research* vol. 42: John Benjamins Publishing Company, 2015.

[232] T. M. Derwing and M. J. Munro, "Accent, intelligibility, and comprehensibility: Evidence from four L1s," *Studies in second language acquisition,* vol. 19, pp. 1-16, 1997.

[233] T. Isaacs and P. Trofimovich, "Deconstructing comprehensibility: Identifying the linguistic influences on listeners' L2 comprehensibility ratings," *Studies in Second Language Acquisition,* vol. 34, pp. 475-505, 2012.

[234] T. M. Derwing, M. J. Munro, and G. Wiebe, "Evidence in favor of a broad framework for pronunciation instruction," *Language learning,* vol. 48, pp. 393-410, 1998.

[235] J. Gordon and I. Darcy, "The development of comprehensible speech in L2 learners," *Journal of Second Language Pronunciation,* vol. 2, pp. 56-92, 2016.

[236] R. Ejzenberg, "The juggling act of oral fluency: A psycho-sociolinguistic metaphor," in *Perspectives on fluency,* 2000, pp. 287-313.

[237] C. H. Nakatani and J. Hirschberg, "A corpus-based study of repair cues in spontaneous speech," *The Journal of the Acoustical Society of America,* vol. 95, pp. 1603-1616, 1994.

[238] A. Wennerstrom, "The role of intonation in second language fluency," in *Perspectives on fluency,* 2000, pp. 102-127.

[239] H. Riggenbach, "Toward an understanding of fluency: A microanalysis of nonnative speaker conversations," *Discourse processes,* vol. 14, pp. 423-441, 1991.

[240] P. Lennon, "Investigating fluency in EFL: A quantitative approach," *Language learning,* vol. 40, pp. 387-417, 1990.

[241] N. Segalowitz, "Access fluidity, attention control, and the acquisition of fluency in a second language," *Tesol Quarterly,* vol. 41, pp. 181-186, 2007.

[242] C. Cucchiarini, H. Strik, and L. Boves, "Quantitative assessment of second language learners' fluency by means of automatic speech recognition technology," *The Journal of the Acoustical Society of America,* vol. 107, pp. 989-999, 2000.

[243] J. Kormos and M. Dénes, "Exploring measures and perceptions of fluency in the speech of second language learners," *System,* vol. 32, pp. 145-164, 2004.

[244] K. Saito, "Effects of instruction on L2 pronunciation development: A synthesis of 15 quasi-experimental intervention studies," *Tesol Quarterly,* vol. 46, pp. 842-854, 2012.

[245] J. Lee, J. Jang, and L. Plonsky, "The effectiveness of second language pronunciation instruction: A meta-analysis," *Applied Linguistics,* vol. 36, pp. 345-366, 2014.

[246] R. I. Thomson and T. M. Derwing, "The effectiveness of L2 pronunciation instruction: A narrative review," *Applied Linguistics,* vol. 36, pp. 326-344, 2014.

[247] C. Liberatore, S. Aryal, Z. Wang, S. Polsley, and R. Gutierrez-Osuna, "SABR: Sparse, Anchor-Based Representation of the Speech Signal," in *Interspeech,* 2015, pp. 608-612.

[248] C. Liberatore, G. Zhao, and R. Gutierrez-Osuna, "Voice Conversion through Residual Warping in a Sparse, Anchor-Based Representation of Speech," presented at the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2018.

[249] A. Solem, "Celery: Distributed Task Queue," 4.0.0 ed: Celery, 2016.

[250] K.-N. Lee, "Automatic generation of pronunciation variants for Korean continuous speech recognition," *The Journal of the Acoustical Society of Korea,* vol. 20, pp. 35-43, 2001.

[251] S.-M. Cho, "An acoustic study of the pronunciation of Korean vowels uttered by Japanese speakers," *Speech Sciences,* vol. 11, 2004.

[252] S. A. Jun, "A phonetic study of stress in Korean," *The Journal of the Acoustical Society of America*, vol. 98, pp. 2893-2893, 1995.

[253] S. T. Lee, "Teaching pronunciation of English using computer assisted learning software: An action research study in an institute of technology in Taiwan," 2008.

[254] M. J. Munro and T. M. Derwing, "Modeling perceptions of the accentedness and comprehensibility of L2 speech the role of speaking rate," *Studies in second language acquisition*, vol. 23, pp. 451-468, 2001.

[255] A. Sundström, "Automatic prosody modification as a means for foreign language pronunciation training," in *Proc. ISCA Workshop on Speech Technology in Language Learning (STILL 98), Marholmen, Sweden,* 1998, pp. 49-52.

[256] K. Yoon, "Imposing native speakers' prosody on non-native speakers' utterances," *현대영미어문학,* vol. 25, pp. 197-215, 2007.

[257] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society: Series B (Methodological),* vol. 39, pp. 1-22, 1977.

[258] Y.-Q. Yu, L. Fan, and W.-J. Li, "Ensemble additive margin softmax for speaker verification," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* 2019, pp. 6046-6050.

[259] C. Li, X. Ma, B. Jiang, X. Li, X. Zhang, X. Liu, *et al.*, "Deep speaker: an end-to-end neural speaker embedding system," *arXiv preprint arXiv:1705.02304,* vol. 650, 2017.

[260] R. Skerry-Ryan, E. Battenberg, Y. Xiao, Y. Wang, D. Stanton, J. Shor, *et al.*, "Towards end-to-end prosody transfer for expressive speech synthesis with tacotron," *arXiv preprint arXiv:1803.09047,* 2018.

[261] A. Graves, "Generating sequences with recurrent neural networks," *arXiv preprint arXiv:1308.0850,* 2013.

[262] F. Biadsy, R. J. Weiss, P. J. Moreno, D. Kanevsky, and Y. Jia, "Parrotron: An end-to-end speech-to-speech conversion model and its applications to hearing-impaired speech and speech separation," *arXiv preprint arXiv:1904.04169,* 2019.

[263] Y. Jia, R. J. Weiss, F. Biadsy, W. Macherey, M. Johnson, Z. Chen, *et al.*, "Direct Speech-to-Speech Translation with a Sequence-to-Sequence Model}}," *Proc. Interspeech 2019,* pp. 1123-1127, 2019.

APPENDIX A

A LIST OF PUBLICATIONS

Below is the list of publications related to this dissertation work.

Journal articles:

1. **S. Ding**, G. Zhao, C. Liberatore, and R. Gutierrez-Osuna, "Learning Structured Sparse Representations for Voice Conversion," *IEEE/ACM Transactions on Audio, Speech, and Language Processing,* vol. 28, pp. 343-354, 2019.
2. **S. Ding**, C. Liberatore, S. Sonsaat, I. Lučić Rehman, A. Silpachai, G. Zhao*, et al.*, "Golden speaker builder - An interactive tool for pronunciation training," *Speech Communication*, vol. 115, pp. 51-66, 2019.
3. G. Zhao, **S. Ding**, and R. Gutierrez-Osuna, "Reference-free foreign accent conversion," submitted to *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.

Conference proceedings:

1. **S. Ding**, C. Liberatore, G. Zhao, S. Sonsaat, E. Chukharev-Hudilainen, J. Levis, and R. Gutierrez-Osuna, "Golden Speaker Builder: an interactive online tool for 179 L2 learners to build pronunciation models," in *Pronunciation in Second Language Learning and Teaching*, 2017, pp. 25-26.
2. **S. Ding**, G. Zhao, C. Liberatore, and R. Gutierrez-Osuna, "Improving Sparse Representations in Exemplar-Based Voice Conversion with a Phoneme-Selective Objective Function," *Proc. Interspeech 2018,* pp. 476-480, 2018.

3. **S. Ding**, C. Liberatore, and R. Gutierrez-Osuna, "Learning Structured Dictionaries for Exemplar-based Voice Conversion," *Proc. Interspeech 2018,* pp. 481-485, 2018.
4. **S. Ding** and R. Gutierrez-Osuna, "Group Latent Embedding for Vector Quantized Variational Autoencoder in Non-Parallel Voice Conversion," *Proc. Interspeech 2019*, pp. 724-728, 2019.
5. G. Zhao, **S. Ding**, and R. Gutierrez-Osuna, "Foreign Accent Conversion by Synthesizing Speech from Phonetic Posteriorgrams," *Proc. Interspeech 2019*, pp. 2843-2847, 2019.
6. **S. Ding**, G. Zhao, and R. Gutierrez-Osuna, "Improving the Speaker Identity of Non-Parallel Many-to-Many Voice Conversion with Adversarial Speaker Recognition," *Proc. Interspeech 2020,* pp. 776-780, 2020.

Under review

1. **S. Ding**, G. Zhao, and R. Gutierrez-Osuna, " Foreign Accent Conversion to arbitrary non-native speakers using zero-shot learning," submitted *Computer Speech and Language*. (Journal article)

APPENDIX B

POST-TEST AND DELAYED POST-TEST INTERVIEW QUESTIONS IN GOLDEN SPEAKER BUILDER STUDY

Post-Test Interview

1. In what ways was the pronunciation training valuable to you? In what ways do you feel you have improved?
2. What was it like practicing with the golden speaker model?
3. How long and how often did you practice?
4. Was the visual feedback helpful?
5. Do you feel like your ability to listen to English speech has improved?
6. Do you feel like your pronunciation has improved? In what ways?
7. Which types of pronunciation were the most difficult to improve?
8. Did you notice any other pronunciation or language items that you had difficulty with during your practice? What were they?
9. What was difficult about practicing with the golden speaker?
10. What kind of suggestions would you give for trying this in the future?
11. What did you notice when you were practicing?
12. Was it easy to repeat the sentences at the same speed?
13. Was it easy to get the consonant sounds correctly?
14. Was it easy to get the vowel sound correctly?
15. What kinds of things did you pay most attention to?
16. What kind of thins did you practice most and why?

17. How do you like the interface of the Golden Speaker?

18. How easy was it to use the website to practice?

19. How comfortable were you using the website?

20. Did you have any technical problems?

21. Would you recommend that others try out the golden speaker builder?

Delayed Post-test Interview

1. Since finishing the training, in what ways was the pronunciation training continued to be valuable to you?

2. Have you continued to use the training materials?

3. Has the training affected how you approach your English pronunciation?

4. Do you feel like your ability to listen to English speech has improved?

5. Do you feel like your pronunciation has improved? In what ways?

6. Which types of pronunciation continue to be difficult to improve?

7. Have you noticed any other pronunciation or language items that have been difficult after your practice? What were they?

8. What things would you suggest for more effective practice?

9. What kind of suggestions would you give for trying this in the future?

10. What features do you most remember about practicing – consonants, vowels or other features of speech?

11. Was it helpful to have someone helping you to practice?

12. What kinds of things do you remember paying attention to?

13. Are there any things you have tried to change in your own speech since the training?

14. Would you recommend that others try out the golden speaker builder?