Converting Foreign Accent Speech Without a Reference

Guanlong Zhao, Shaojin Ding, and Ricardo Gutierrez-Osuna, *Senior Member, IEEE*

***Abstract*—Foreign accent conversion (FAC) is the problem of generating a synthetic voice that has the voice identity of a second-language (L2) learner and the pronunciation patterns of a native (L1) speaker. This synthetic voice has been referred to as a "golden-speaker" in the pronunciation-training literature. FAC is generally achieved by building a voice-conversion model that maps utterances from a source (L1) speaker onto the target (L2) speaker. As such, FAC requires that a reference utterance from the L1 speaker be available at synthesis time. This greatly restricts the application scope of the FAC system. In this work, we propose a "reference-free" FAC system that eliminates the need for reference L1 utterances at synthesis time, and transforms L2 utterances directly. The system is trained in two steps. First, a conventional FAC procedure is used to create a golden-speaker using utterances from a reference L1 speaker (which are then discarded) and the L2 speaker. Second, a pronunciation-correction model is trained to convert L2 utterances to match the golden-speaker utterances obtained in the first step. At synthesis time, the pronunciation-correction model directly transforms a novel L2 utterance into its golden-speaker counterpart. Our results show that the system reduces foreign accents in novel L2 utterances, achieving a 20.5% relative reduction in word-error-rate of an American English automatic speech recognizer and a 19% reduction in perceptual ratings of foreign accentedness obtained through listening tests. Over 73% of the listeners also rated golden-speaker utterances as having the same voice identity as the original L2 utterances.**

***Index Terms*—Accent conversion, acoustic model, sequence-to-sequence voice conversion, speech modification, speech synthesis.**

I. Introduction

FOREIGN accent conversion (FAC) [1] aims to create a synthetic voice that has the voice identity (or timbre) of a non-native speaker but the pronunciation patterns (or accent)[1] of a native speaker. In the context of computer-assisted pronunciation training [1]–[4], this synthetic voice is often referred to as a "golden speaker" for the non-native speaker –a second-language (L2) learner. The rationale is that the golden speaker is a better target for the L2 learner to imitate than an arbitrary native speaker, because the only difference between the golden speaker and the L2 learner's own voice is the accent, which makes mispronunciations more salient. In addition to pronunciation training, FAC finds applications in movie dubbing [5], personalized Text-To-Speech (TTS) synthesis [6], [7], and improving automatic speech recognition (ASR) performance [8].

The main challenge in FAC is that one does not have ground-truth data for the desired golden speaker, since, in general, the L2 learner is unable to produce speech with a native accent. Therefore, it is not feasible to apply conventional voice-conversion techniques to the FAC problem. Previous solutions work around this issue by requiring a reference utterance from a native (L1) speaker at synthesis time. But this limits the types of pronunciation practice that FAC techniques can provide, e.g., the L2 learner can only practice sentences that have already been prerecorded by the reference L1 speaker.

To address this issue, **we propose a new FAC system that does not require a reference L1 utterance at *inference time*. We refer to this type of FAC system as *reference-free*.** Assume that we have a training set of parallel utterances from the L2 learner and from a reference L1 speaker. The training pipeline consists of two steps. In step one, we build an L2 speech synthesizer [9] that maps speech embeddings (see below) from L2 utterances into their corresponding Mel-spectrograms. The speech embeddings are extracted using an acoustic model trained on a large corpus of native speech, so they are speaker-independent [10], [11]. We then drive the L2 synthesizer with speech embeddings extracted from the L1 utterances. This results in a set of golden-speaker utterances that have the voice identity of the L2 learner (since they are generated from the L2 synthesizer) and the pronunciation patterns of the L1 speaker (since the input is obtained from an L1 utterance). The L1 utterances can be discarded at this point. In the second (and key) step, we train a pronunciation-correction model that converts the

[1]We use the terms "accent" and "pronunciation pattern" interchangeably in this manuscript. A foreign accent can be defined as the systematic deviation from the standard norm of a spoken language. The deviations can be observed at the segmental level (e.g., substitution, deletion, or insertion of phones) and/or at the suprasegmental level (prosody deviations; such as differences in intonation, tone, stress, and rhythm.)

Manuscript received July 1, 2020; revised November 9, 2020 and December 24, 2020; accepted February 17, 2021. Date of publication July 8, 2021; date of current version July 30, 2021. This work was supported by NSF Awards 1619212 and 1623750. The work of Guanlong Zhao was supported by solely with TAMU resources. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Yu Tsao. *(Corresponding author: Guanlong Zhao.)*

Guanlong Zhao was with the Department of Computer Science and Engineering, Texas A&M University (TAMU), College Station, TX 77843 USA. He is now with the Google LLC, New York, NY 10011 USA (e-mail: zhao@aggienetwork.com).

Shaojin Ding and Ricardo Gutierrez-Osuna are with the Department of Computer Science and Engineering, Texas A&M University (TAMU), College Station, TX 77843 USA (e-mail: shjd@tamu.edu; rgutier@tamu.edu).

Digital Object Identifier 10.1109/TASLP.2021.3060813

L2 utterances to match the golden-speaker utterances obtained in the first step, which serve as a target. During inference time, we can then feed a new L2 utterance to the pronunciation-correction model, which then generates its "accent free" counterpart.

The pronunciation-correction model is based on a state-of-the-art sequence-to-sequence (seq2seq) voice conversion framework proposed by Zhang *et al.* [12], which we use as a baseline. Their system consists of an encoder to extract hidden representations of the input features (e.g., Mel-spectra), an attention mechanism to learn the alignment between the input and output sequences, a decoder to predict the output Mel-spectrograms, and multi-task phoneme classifiers to help stabilize the training process. During our internal evaluation of the baseline system, we found that it had difficulty converting between an L2 and an L1 speaker because L2 utterances tend to have a significant amount of disfluency and hesitations, which makes it hard for the attention mechanism to properly align input and output sequences. To address this issue, our system includes a forward-and-backward decoding technique [13], [14] in the pronunciation-correction model to help the attention mechanism and decoder to fully utilize the information in the input data. The rationale is that, by forcing the decoder to compute the attention alignments from both the forward and backward directions during training, we can make the decoder incorporate useful contextual information from both the past and future when producing the alignment. Throughout this study, we use a high-quality WaveGlow [15] real-time neural vocoder to convert Mel-spectrograms to speech waveform.

The manuscript is organized as follows. Section II reviews prior approaches on FAC as well as related work in seq2seq voice conversion. Section III describes the proposed reference-free FAC system. Sections V, IV, and VI present the objective and subjective evaluation results and an in-depth discussion of these results. Lastly, we summarize the findings of this work in Section VII and point out future research directions. We include three appendices that provide related details.

II. Related Work

A. Conventional FAC Methods

FAC is related to the more general problem of voice conversion (VC) [16]. In VC, one seeks to transform a source speaker's speech into that of a (known) target speaker. The conversion aims to match the voice characteristics of the target speaker, which include vocal tract configurations, glottal characteristics, pitch range, pronunciation, and speaking rate; ideally, the only information retained from the source speech is its linguistic content, i.e., the words that were uttered. In contrast with VC, FAC seeks to combine the linguistic content and pronunciation characteristics of the source speaker with the voice identity of the target speaker. This is a more challenging problem than VC for two reasons. First, FAC lacks ground-truth since generally there are no recordings of the L2 speaker producing speech with the desired native target accent. But, more importantly, FAC requires decomposing the speech into voice identity and accent, whereas VC does not. Several techniques have been proposed to perform this decomposition, which can be grouped into articulatory and acoustic methods. The basic strategy in articulatory methods is to build an articulatory synthesizer for the L2 speaker, that is, a mapping from the speaker's articulatory trajectories (e.g., tongue and lip movements) to his or her acoustics features (e.g., Mel Cepstra.) Once complete, the L2 speaker's articulatory synthesizer is driven by articulatory trajectories from an L1 speaker to produce "accent-free" speech.[2] A number of techniques can be used to build the articulatory synthesizer, including unit-selection [18], GMMs [19], and DNNs [20].

Decoupling voice identity from accent in the articulatory domain is intuitive, but impractical in most cases since collecting articulatory data is expensive and requires specialized equipment.[3] In contrast, decoupling voice identity from accent in the acoustic domain is more practical since it only requires recording speech with a microphone, but is more challenging from a speech-processing standpoint. The conventional approach used in VC (pairing source and target frames via dynamic time warping; DTW) cannot be used in FAC, since it would result in a model that maps native-accented source into non-native-accented target speech. Instead, source and target frames have to be paired based on their linguistic similarity. In early work, Aryal and Gutierrez-Osuna [24] replaced DTW with a technique that matched source (L1) and target (L2) frames based on their MFCC similarity after performing vocal tract length (VTL) normalization. Then, they trained a GMM with those frame pairs to map source L1 utterances to have the target L2 speaker's identity, while retaining the native pronunciations. More recently, Zhao *et al.* [25] used a speaker-independent acoustic model (i.e., from an ASR system) to estimate the posterior probability that each frame belonged to a set of pre-defined phonetic units –a phonetic posteriorgram (PPG) [26]. Once a PPG had been computed for each source and target frame in the corpus, the two were paired in a many-to-many fashion based on the similarity between their respective PPGs [11], [25]. In their study, matching source and target frames based on their PPG similarity achieved better ratings on accentedness and acoustic quality than matching them based on the VTL-normalized MFCC similarity of Aryal and Gutierrez-Osuna [24].

B. FAC Methods Using Sequence-to-Sequence Models

More recently, Zhao *et al.* [27] have used sequence-to-sequence (seq2seq) models to perform FAC. In their approach, a seq2seq speech synthesizer is trained to convert PPGs to Mel-spectra using recordings from the L2 speaker. Then, golden-speaker utterances are generated by driving the seq2seq synthesizer with PPGs extracted from an L1 utterance, a process that reminisces articulatory-based methods (i.e., if PPGs are viewed as articulatory information). Their method produced speech that was significantly less accented than the original L2 speech. Seq2seq models have also garnered much attention in the VC literature since, unlike prior frame-by-frame VC

[2] This process can be likened to "voice puppetry" [17] where the puppet is the articulatory synthesizer and the strings are the native speaker's articulations.

[3] Articulatory measurements can be performed via electromagnetic articulography [18] ultrasound imaging [21], palatography [22], and more recently, real-time MRI [23].

models [28]–[33], they can convert segmental and prosody features simultaneously, leading to better conversion performance. Miyoshi *et al.* [34] built a seq2seq model that mapped source context posterior probabilities to the target's; they obtained better speech individuality ratings (but worse audio quality) than a baseline without the context posterior mapping process. Zhang *et al.* [35] concatenated bottleneck features and Mel-spectrograms from a source speaker, used a seq2seq model to convert the concatenated source features into the target Mel-spectrogram, and finally recovered the speech waveform with a WaveNet [36] vocoder. This model outperformed the best-performing system from the 2018 Voice Conversion Challenge [37]. Zhang *et al.* then applied text supervision [12] on top of [35] to resolve some of the mispronunciations and artifacts in the converted speech. More recently, they extended their framework to the non-parallel condition [38] with trainable linguistic and speaker embeddings. Other notable sequence-to-sequence VC works include [39], which proposed a novel loss term that enforced attention weight diagonality to stabilize the seq2seq training; the Parrotron [8] system, which used large-scale corpora and seq2seq models to normalize arbitrary speaker voices to a synthetic TTS voice; and [40], which used a fully convolutional seq2seq model instead of conventional recurrent neural networks (RNNs, e.g., LSTM) because RNNs are costly to train and difficult to optimize for parallel computing.

C. Prior Reference-Free FAC Approach

To the best of our knowledge, the only prior work on reference-free FAC is a recent study by Liu *et al.* [41]. Their system used a speaker encoder, a multi-speaker TTS model, and an ASR encoder. The speaker encoder and the TTS model are trained with L1 speech only, and the ASR encoder is trained on speech data from L1 speakers and the target L2 speaker. During testing, they use the speaker encoder and ASR encoder to extract speaker embeddings and linguistic representations from the input L2 testing utterance, respectively. Then, they concatenate the two and feed them to the multi-speaker TTS model, which then generates the accent-converted utterance. Their evaluations suggested that the converted speech had a near-native accent, but did not capture the voice identity of the target L2 speaker because it had to be interpolated by their multi-speaker TTS. Our proposed method avoids this problem since our pronunciation-correction module is trained on golden-speaker utterances that have been pre-generated for the L2 speaker using a conventional FAC framework.

III. Method

Our proposed approach to reference-free FAC is illustrated in Figure 1. The system requires a parallel corpus of utterances from the L2 speaker and a reference L1 speaker. As outlined in the introduction and shown in the figure, the training process consists of two steps. In a first step, we build a speech synthesizer for the L2 speaker that converts speech embeddings into Mel-spectrograms. We then drive the L2 synthesizer with a set of utterances from the reference L1 speaker, to produce a set of golden-speaker utterances (i.e., L2 voice identity with L1

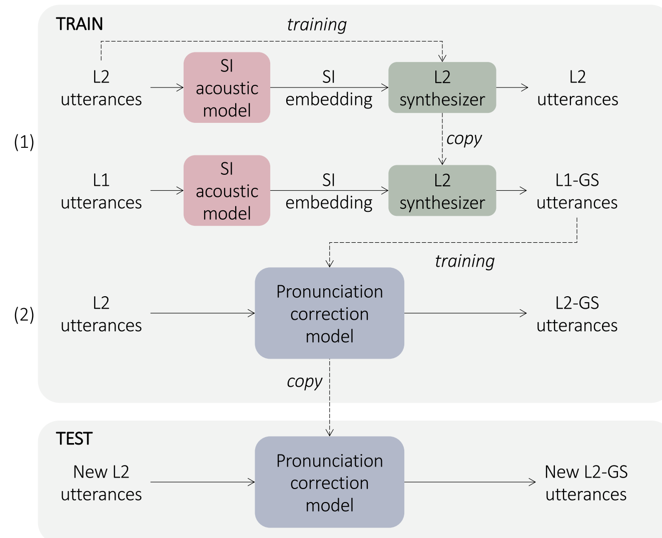


Fig. 1. Overall workflow of the proposed system. L1: native; L2: non-native; GS: golden speaker; SI: speaker independent. The training stage consists of two steps. In step 1, we use a conventional FAC procedure to generate a set of golden-speaker utterances (L1-GS), which serve as targets for step 2. In step 2, we train a pronunciation-correction model that converts L2 utterances into the L1-GS utterances obtained earlier. Once the pronunciation correction model is trained, in the testing stage, a new L2 utterance is processed by the pronunciation-correction model to create its "accent-free" counterpart (L2-GS).

pronunciation patterns). We refer to these as L1 golden-speaker (L1-GS) utterances, since they are obtained using L1 utterances as a reference. The L1 utterances can be discarded at this point. In a second step, we build a pronunciation-correction model that directly transforms L2 utterances to match their corresponding L1-GS utterances obtained in the previous step, that is, without the need for the L1 reference. We refer to the outputs of the pronunciation-correction model as L2-GS utterances since they are generated directly from L2 utterances (i.e., in a reference-free fashion). Critical in this process is the generation of the speaker embeddings, which we describe first.

A. Extracting Speaker-Independent Speech Embeddings

We use an acoustic model (AM) to generate a speaker-independent (SI) speech embedding for an input (L1 or L2) utterance. Our AM is a Factorized Time Delayed Neural Network (TDNN-F) [42], [43], a feedforward neural network that utilizes time-delayed input in its hidden layers to model long term temporal dependencies. TDNN-F can achieve performance on Large Vocabulary Continuous Speech Recognition (LVCSR) tasks that is comparable to that of AMs based on recurrent structures (e.g., Bi-LSTMs), but is more efficient during training and inference due to its feedforward nature [42]. To produce an SI speech embedding, we concatenate each acoustic feature vector (40-dim MFCC) with an i-vector (100-dim) of the corresponding speaker [44] and use them as inputs to the AM, which we then train on a large corpus from a few thousand native speakers (Librispeech [45]).[4]

[4]The AM is trained following the Kaldi [46] "tdnn_1 d" configuration of the TDNN-F model. We use the full training set (960 hours) in the Librispeech

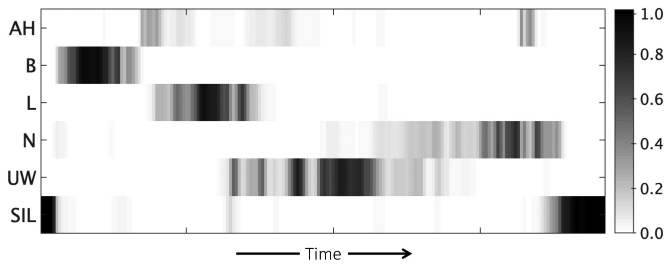


Fig. 2. Mono-PPG of a spoken word balloon, whose pronunciation is "B AH L UW N" in the ARPAbet phoneme set. "SIL" means silence. AQThe colorbar shows the probability values from zero to one. For visualization purposes, we omitted rows (monophones) with low values, and also aggregated the probability mass of all monophones that only differ in stressing and word positions (e.g., we added the probability mass of AH{∅, 0, 1, 2}_{initial, mid, final} into a single entry AH). An American English speaker uttered this word.

As part of this study, we evaluated three different speech embeddings:

- **Senone phonetic posteriorgram (Senone-PPG)**: The output from the final softmax layer of the AM, which is high dimensional (6024 senones) and contains fine-grained information about the pronunciation pattern in the input utterance.
- **Bottleneck feature (BNF)**: The output of the layer prior to the final softmax layer of the AM. The BNF contains rich classifiable information for a phoneme recognition task, but lower dimensionality (256).
- **Monophone phonetic posteriorgram (Mono-PPG)**: The phonetic posteriorgram obtained by collapsing the senones into monophone symbols (346 monophones with word positions, e.g., word-initials, word-finals). For each monophone symbol, we aggregate the probability mass of all the senones that share the same root monophone. Figure 2 visualizes the Mono-PPG of a spoken word. We omit the visualization of the other two speech embeddings since they are more difficult to interpret.

B. Step 1: Generating a Reference-Based Golden-Speaker (L1-GS)

The speech synthesizer is based on a modified Tacotron2 architecture[5] [9], and is illustrated in Figure 3. The model follows a general encoder-decoder (or seq2seq) paradigm with an attention mechanism. Conceptually, an encoder-decoder architecture uses an encoder (usually a recurrent neural network; RNN) to "consume" input sequences and generate a high-level hidden representation sequence. Then, a decoder (an RNN with an attention mechanism) processes the hidden representation sequence. The attention mechanism allows the decoder to decide which parts of the hidden representation sequence contain useful information to make the predictions. At each output time step, the attention mechanism computes an attention context vector (a weighted sum of the hidden representation sequence) to summarize the contextual information. The decoder RNN reads the attention context vectors and predicts the output sequence in an autoregressive manner.

Our speech synthesizer takes the speech embeddings as input. Then, if the input speech embeddings have high dimensionality (e.g., Senone-PPGs), we reduce their dimensions through a learnable input PreNet. This step is essential for the model to converge when using high-dimensional speech embeddings as input. For speech embeddings with lower dimensionality, such as Mono-PPGs and BNFs, we skip the input PreNet. The speech embeddings are then passed through multiple 1-D convolutional layers, which model longer-term context. Next, an encoder (one Bi-LSTM) converts the convolutions into a hidden linguistic representation sequence. Finally, we pass the hidden linguistic representation sequence to the decoder, which consists of a location-sensitive attention mechanism [47] and a decoder LSTM, to predict the raw Mel-spectrogram. We note that the input and output sequences of the speech synthesizer have the same length,[6] and thus, the speech synthesizer only models the speaker identity and retains the phonetic and prosodic cues carried by the input speech embeddings.

Formally, let $[a; b]$ represent the operation of concatenating vectors a and b, $h = [h_1, \ldots, h_T]$ be the full sequence of hidden linguistic representation from the encoder and $(\cdot)^\top$ denote the matrix transpose. At the i-th decoding time step, applying the location-sensitive attention mechanism, the attention context vector c_i is the weighted sum of h,

$$c_i = \alpha_i \cdot h^\top, \tag{1}$$

$$\alpha_i = \text{AttentionLayers}\left(q_i, \alpha_{i-1}, h\right) = \left[\alpha_i^1, \ldots, \alpha_i^T\right], \tag{2}$$

$$q_i = \text{AttentionLSTM}\left(q_{i-1}, \left[c_{i-1}; \text{DecoderPreNet}\left(\hat{y}_{i-1}^{mel}\right)\right]\right), \tag{3}$$

$$\alpha_i^j = \frac{\exp\left(e_{ij}\right)}{\sum_{j=1} \exp\left(e_{ij}\right)}, \tag{4}$$

$$e_{ij} = v^\top \tanh\left(W q_i + V h_j + U f_i^j + b\right), \tag{5}$$

$$f_i = F * \alpha_{i-1} = \left[f_i^1, \ldots, f_i^T\right], F \in \mathbb{R}^{k \times r}. \tag{6}$$

$\alpha_i = [\alpha_i^1, \ldots, \alpha_i^T]$ are the attention weights. q_i is the output of the attention LSTM, and $\hat{y}_{i-1}^{mel}$ is the predicted raw Mel-spectrum from the previous time step. v, W, V, U, b, F are learnable parameters of the attention layers. F contains k 1-D learnable kernels with kernel size r, and $f_i^j \in \mathbb{R}^k$ is the result of convolving α_{i-1} at position j with F.

Next, let d_i be the output of the decoder LSTM at decoding time step i, and $\hat{y}_i^{mel}$ be the new raw Mel-spectrum prediction, we have,

$$d_i = \text{DecoderLSTM}\left(d_{i-1}, [q_i; c_i]\right), \tag{7}$$

corpus for acoustic modeling. A subset (200 hours) of the training set is used to train the i-vector extractor.

[5]To facilitate the method description and maintain consistency with prior literature, we adopt the following terminologies from Tacotron2: PreNet: Two fully connected layers with a ReLU nonlinearity; PostNet: Five stacked 1-D convolutional layers; LinearProjection: One fully connected layer.

[6]A recent study [48] (published while this manuscript was under review) used a conversion model similar to the one used in our work. The authors observed that if the temporal structure (such as the length) of the input and output sequences were the same, then removing the attention module did not hurt performance, which suggests a potential path to further simplify the model structure of the speech synthesizer we used here.

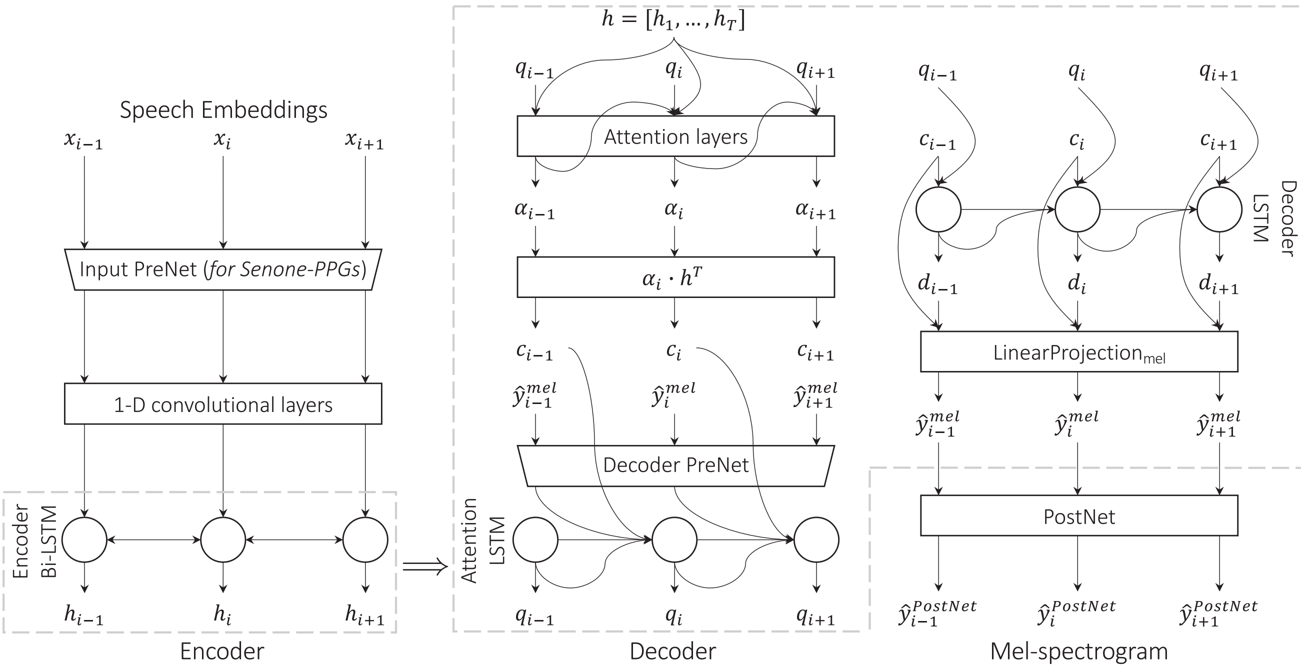


Fig. 3. Speech embedding to Mel-spectrogram synthesizer. The speech embeddings are sequentially processed by an input PreNet (optional, for Senone-PPGs only), convolutional layers, an encoder, a decoder, and a PostNet to generate their corresponding Mel-spectra. We omitted the stop token predictions in the figure for better visualization.

$$\hat{y}_i^{mel} = \text{LinearProjection}_{\text{mel}}\left([d_i; c_i]\right). \tag{8}$$

At each time step, to determine if the decoder prediction reaches the end of an utterance, we compute a binary stop token (1: stop; 0: continue) using a separate trainable fully connected layer,

$$\hat{y}_i^{stop} = \begin{cases} 1 & \text{Sigmoid}\left(\text{LinearProjection}_{\text{stop}}\left([d_i; c_i]\right)\right) \geq 0.5 \\ 0 & \text{Sigmoid}\left(\text{LinearProjection}_{\text{stop}}\left([d_i; c_i]\right)\right) < 0.5 \end{cases} \tag{9}$$

The original Tacotron 2 was designed to accept character sequences as input, which are significantly shorter than our speech embedding sequences. For example, each sentence in our corpus contains 41 characters on average, whereas the corresponding speech embedding sequence has a few hundred frames. Therefore, the vanilla location-sensitive attention mechanism might fail, as pointed out in [35]. As a result, the inference would be ill-conditioned and would generate non-intelligible speech. Following a preliminary study [27] of this work, we add locality constraint to the attention mechanism. Speech signals have a strong temporal-continuity and progressive nature. To capture the phonetic context, we only need to look at the speech embeddings in a small local window. Inspired by this, at each decoding step during training, we constrain the attention mechanism to only consider the hidden linguistic representation within a fixed window centered on the current frame, i.e., let,

$$\tilde{h} = [0, \ldots, 0, h_{i-w}, \ldots, h_i, \ldots, h_{i+w}, 0, \ldots, 0], \tag{10}$$

where w is the window size. Consequentially, we replace eq. (2) with eq. (11),

$$\alpha_i = \text{AttentionLayers}\left(q_i, \alpha_{i-1}, \tilde{h}\right). \tag{11}$$

Finally, to further improve the synthesis quality, the speech synthesizer appends a PostNet after the decoder to predict residual spectral details from the raw Mel-spectrum prediction, and then adds the spectral residuals to the raw Mel-spectrum,

$$\hat{y}_i^{PostNet} = \hat{y}_i^{mel} + \text{PostNet}\left(\hat{y}_i^{mel}\right). \tag{12}$$

The advantage of the PostNet is that it can see the entire decoded sequence. Therefore, the PostNet can use both past and future information to correct the prediction error for each individual frame [49].

The loss function for training this speech synthesizer is,

$$L = w_1 \left(\|Y_{mel} - \hat{Y}_{mel}^{Decoder}\|_2 + \||Y_{mel} - \hat{Y}_{mel}^{PostNet}\|_2\right) + w_2 \text{CE}\left(Y_{stop}, \hat{Y}_{stop}\right), \tag{13}$$

where Y_{mel} is the ground-truth Mel-spectrogram; $\hat{Y}_{mel}^{Decoder}$ and $\hat{Y}_{mel}^{PostNet}$ are the predicted Mel-spectrograms from the decoder and PostNet, respectively; Y_{stop} and $\hat{Y}_{stop}$ are the ground-truth and predicted stop token sequences; CE$(\cdot)$ is the cross-entropy loss; w_1 and w_2 control the relative importance of each loss term.

The predicted Mel-spectrograms are converted back to audio waveforms using a WaveGlow neural vocoder trained on the L2 utterances (cf. Section III-D for more details). We then drive the L2 synthesizer with a set of utterances from the reference L1 speaker, to produce the L1-GS utterances that are used in Step 2.

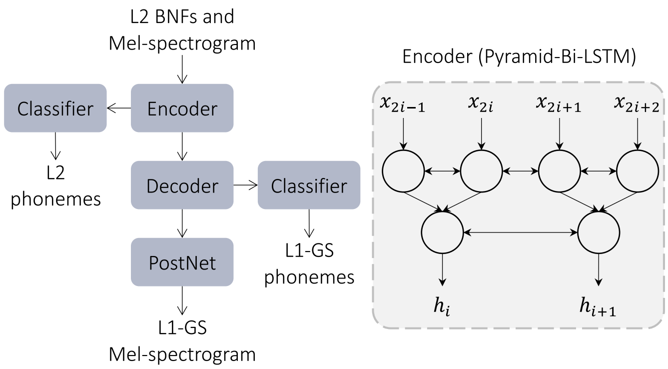


Fig. 4. Training pipeline of the baseline pronunciation-correction model. The input feature sequence (concatenation of bottleneck features [BNFs] and Mel-spectra) from the L2 speaker is converted into the L1-GS Mel-spectrogram. The phoneme classifications are only applied to stabilize the model training and are discarded during testing. The encoder is constructed with a two-layer Pyramid-Bi-LSTM. The decoder has the same neural network structure as the one in Figure 3.

C. Step 2: Generating the Reference-Free Golden Speaker (L2-GS) Via Pronunciation-Correction

Our pronunciation-correction model is based on a state-of-the-art seq2seq VC system proposed by Zhang *et al.* [12]. We chose this system as a baseline since it outperformed the best system in the Voice Conversion Challenge 2018 [37]. The rationale behind using a VC system as the pronunciation-correction model is that VC can convert both the voice identity and the accent to match the target speaker. In our application scenario, we treat the L2 speaker and the L1-GS as the source and target speakers in a VC task, respectively. Since the two speakers already share the same voice identity, the VC model only needs to match the accent of the target speaker (i.e., the golden speaker). During the inference stage, we can directly input L2 speech into the pronunciation-correction model, and the output will share similar pronunciation patterns as the L1-GS. The difficulty of this procedure is that L2 speakers tend to have disfluencies, hesitations, and inconsistent pronunciations, making the conversion much harder than converting between two native speakers, as discussed in prior literature [11]. To overcome this difficulty, we propose to use a variation of the forward-and-backward decoding technique [13], [14], in addition to the baseline pronunciation model, to achieve better pronunciation-correction performance. We first formally introduce the baseline system, and then describe the proposed improvement.

The baseline system is also based on an encoder-decoder paradigm with an attention mechanism. Figure 4 shows an overview of the baseline system. Unlike conventional frame-by-frame VC systems (e.g., GMM, feedforward neural networks), which need time-alignment between the source and target speakers to generate the training frame pairs, seq2seq systems use an attention mechanism to produce learnable alignments between the input and output sequences. Therefore, they can also adjust for prosodic differences (e.g., pitch, duration, and stressing) between the input and output sequences. In our application, this is crucial since prosody errors also contribute to foreign accentedness.

Specifically, let x_i be the i-th feature vector in the sequence, the input $X = [x_1, \ldots, x_{T_{in}}]$ to the conversion system is the concatenation of the bottleneck features[7] (i.e., BNFs, cf. Section III-A) and Mel-spectrogram computed from the L2 utterance. The output sequence is denoted by $Y_{mel} = [y_1^{mel}, \ldots, y_{T_{out}}^{mel}]$ where y_i^{mel} is the i-th Mel-spectrum of the L1-GS utterance. A two-layer Pyramid-Bi-LSTM encoder [50] with a down-sampling rate of two consumes the input sequence and produces the encoder hidden embeddings $h = [h_1, \ldots, h_{\lfloor i/2 \rfloor}, \ldots, h_{\lfloor T_{in}/2 \rfloor}]$, where $h_{\lfloor i/2 \rfloor}$ is one encoder hidden embedding vector, and $\lfloor \cdot \rfloor$ is the floor-rounding operator. The first Bi-LSTM layer does the recurrent computations on X and outputs $h_{layer1} = [h_{layer1}^1, \ldots, h_{layer1}^{T_{in}}]$. We then concatenate each two of the consecutive frames in h_{layer1} to form $[[h_{layer1}^1; h_{layer1}^2], \ldots, [h_{layer1}^{T_{in}-1}; h_{layer1}^{T_{in}}]]$. Finally, we feed the concatenated vectors to the second Bi-LSTM layer to produce h. In the case that we have an odd number of frames in the input sequence, we drop the last frame, which is generally a silent frame. The down-sampling effectively reduces the sequence length of the input, which speeds up the encoder computation by a factor of two and makes it easier for the attention mechanism to learn a meaningful alignment between the input and output sequences.

The decoder in this model has a similar neural-network structure as the speech synthesizer decoder in Section III-B (Figure 3), with only two differences: (1) to replicate Zhang *et al.* [12], we use the forward-attention technique [51] instead of eq. (4) to normalize the attention weights; (2) the locality constraint defined in equations (10) and (11) is discarded. The decoder predicts the output raw Mel-spectrogram sequence $\hat{Y}_{mel}^{Decoder} = [\hat{y}_1^{mel}, \ldots, \hat{y}_{T_{out}}^{mel}]$ and the stop token sequence $\hat{Y}_{stop} = [\hat{y}_1^{stop}, \ldots, \hat{y}_{T_{out}}^{stop}]$ following equations (8) and (9), respectively. $\hat{Y}_{mel}^{Decoder}$ is also processed through a PostNet to generate a residual-compensated Mel spectrogram $\hat{Y}_{mel}^{PostNet}$, following eq. (12). As in the previous step, $\hat{Y}_{mel}^{PostNet}$ is converted back to audio waveforms using a WaveGlow neural vocoder trained on the L2 utterances.

In addition, the baseline system uses multi-task learning [52], [53] to make the synthesized pronunciations more stable. Two independent phoneme classifiers, each containing one fully-connected layer and a softmax operation, are added to predict the input and output phoneme sequences $\hat{Y}_{inP} = [\hat{y}_1^{inP}, \ldots, \hat{y}_{T_{in}}^{inP}]$ and $\hat{Y}_{outP} = [\hat{y}_1^{outP}, \ldots, \hat{y}_{T_{out}}^{outP}]$, respectively. These phoneme classifiers are only used during training and are discarded in inference. c_i and q_i are defined in the same manner as in equations (1) and (3).

$$\hat{y}_i^{inP} = \text{PhonemeClassifier}_{\text{in}}\left(h_i\right). \tag{14}$$

$$\hat{y}_i^{outP} = \text{PhonemeClassifier}_{\text{out}}\left([q_i; c_i]\right). \tag{15}$$

[7] Zhang *et al.* [12] use BNFs in their implementation, and we follow this design choice to replicate their system.

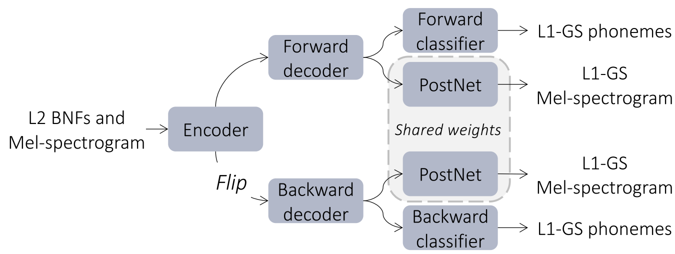


Fig. 5. Proposed forward-and-backward decoding model for pronunciation-correction. The existing decoder in the baseline model is denoted as the forward decoder here. We omitted the other common components it shares with the baseline model. The PostNet of the two decoders shares the same set of weights. This forward-and-backward decoding procedure is only activated during training.

The final loss function of the baseline system becomes,

$$\begin{aligned} L_{base} = w_1 &\left(\|Y_{mel} - \hat{Y}_{mel}^{Decoder}\|_2 + \||Y_{mel} - \hat{Y}_{mel}^{PostNet}\|_2\right) \\ &+ w_2 \mathrm{CE}\left(Y_{stop}, \hat{Y}_{stop}\right) \\ &+ w_3 \left(\mathrm{CE}\left(Y_{inP}, \hat{Y}_{inP}\right) + \mathrm{CE}\left(Y_{outP}, \hat{Y}_{outP}\right)\right), \end{aligned} \tag{16}$$

where Y_{inP}, Y_{outP} are the ground-truth input and output phoneme sequence, respectively.

To improve predictive performance, we propose a modification to the baseline system that applies forward-and-backward decoding during the training process. The forward-and-backward decoding technique maintains two separate decoders, i.e., the forward and backward decoders. The forward decoder processes the encoder outputs in the forward direction, whereas the backward decoder reads the encoder outputs reversely. Different variations of this technique have been applied to TTS [14] and ASR [13]. Figure 5 shows an overview of this procedure. During training, we add a backward decoder to the baseline model. The backward decoder has the same structure as the existing decoder (denoted as the forward decoder) but with a different set of weights. The backward decoder functions the same as the forward decoder except that it processes the encoder's output in *reverse* order and predicts the output Mel-spectrogram $\hat{Y}_{mel}^{bwd}$ *reversely* as well. The backward decoder, like its forward counterpart, also predicts its own set of stop tokens $\hat{Y}_{stop}^{bwd}$, output phoneme labels $\hat{Y}_{outP}^{bwd}$, and uses the shared PostNet to predict a refined Mel-spectrogram $\hat{Y}_{mel\text{-}PostNet}^{bwd}$. The loss terms contributed by adding this backward decoder are,

$$\begin{aligned} L_{bwd} = w_1 &\left(\|Y_{mel} - \hat{Y}_{mel}^{bwd}\|_2 + \||Y_{mel} - \hat{Y}_{mel\text{-}PostNet}^{bwd}\|_2\right) \\ &+ w_2 \mathrm{CE}\left(Y_{stop}, \hat{Y}_{stop}^{bwd}\right) + w_3 \left(\mathrm{CE}\left(Y_{outP}, \hat{Y}_{outP}^{bwd}\right)\right). \end{aligned} \tag{17}$$

Additionally, to force the two decoders to learn complementary information from each other, we train the two decoders to produce the same attention weights by including the following loss term,

$$L_{att} = w_4 \|\alpha_{fwd} - \alpha_{bwd}\|_2, \tag{18}$$

where α_{fwd} and α_{bwd} are the attention weights of the forward and backward decoder, respectively.

The final loss term of the proposed system is,

$$L_{proposed} = L_{base} + L_{bwd} + L_{att}. \tag{19}$$

The rationale behind the forward-and-backward decoding is that RNNs are generally more accurate at the initial decoding time steps, but performance decreases as the predicted sequence becomes longer because the prediction errors accumulate due to the autoregression. By including two decoders that model the input data in two different directions, and by constraining them to produce similar attention weights, we force the two decoders to incorporate information from both the past and future, thus improving their modeling power. Note that we only use both decoders during training. During inference time, we keep either the forward or backward decoder and discard the other. Therefore, the model size is exactly the same as the baseline model.

D. WaveGlow Vocoder

We use a WaveGlow vocoder [15] to convert the output of the speech synthesizer back into a speech waveform. WaveGlow is a flow-based [54] network capable of generating high-quality speech from Mel-spectrograms. It takes samples from a zero mean spherical Gaussian (with variance σ) with the same number of dimensions as the desired output and passes those samples through a series of layers that transform the simple distribution to one that has the desired distribution. In the case of training a vocoder, we use WaveGlow to model the distribution of audio samples conditioned on a Mel-spectrogram. During inference, random samples from the zero-mean spherical Gaussian are concatenated with the up-sampled (matching the speech sampling rate) Mel-spectrogram to predict the audio samples. WaveGlow can achieve real-time inference speed, whereas WaveNet takes a long time to synthesize an utterance due to its auto-regressive nature. For more details about the WaveGlow vocoder, we refer readers to the original study by Prenger *et al.* [15], which also showed that WaveGlow generates speech with quality comparable to WaveNet.

IV. Experimental Setup

For the FAC task (training the speech synthesizers, WaveGlow neural vocoders, and pronunciation-correction models), we used one native speaker (BDL; American accent)[8] from CMU-ARCTIC corpus [55] and two non-native speakers (YKWK, Korean; TXHC, Chinese) from the L2-ARCTIC corpus[9] [56]. We split the data from all speakers into non-overlapping training (1032 utterances), validation (50 utterances), and testing (50 utterances) sets. Recordings from BDL were sampled at 16 kHz. Recordings in the L2-ARCTIC corpus were resampled from 44.1 kHz to 16 kHz to match BDL's sampling rate and were pre-processed with Audacity [57] to remove any ambient

[8]We chose to use BDL as the native speaker since our AM has reasonable recognition accuracy on his speech (cf. Table I). If the AM were to perform poorly on the native speaker, then the L1-GS utterances would include more mispronunciations and therefore degrade the overall accent conversion performance.

[9][Online]. Available: https://psi.engr.tamu.edu/l2-arctic-corpus

background noise. In all FAC tasks, we extracted 80-dim Mel-spectrogram with a 10 ms shift and 64 ms window size. All neural network models were implemented in PyTorch [58] and trained with an NVIDIA Tesla P100 GPU. In all experiments, we trained speaker-dependent WaveGlow neural vocoders for L2 speakers using the official implementation provided by Prenger *et al.* [15][10].

V. Experiments And Results

We conducted two experiments to evaluate the proposed FAC system on a thorough set of objective measures (e.g., word error rates, Mel Cepstral distortion) and subjective measures (degree of foreign accent, audio quality, and voice similarity). In experiment 1, we evaluated the reference-based golden speaker (L1-GS) generated by the L2 speech synthesizer (Section III-B). Then, in experiment 2, we evaluated the reference-free golden speaker (L2-GS) produced by the pronunciation-correction model (Section III-C).

A. Experiment 1: Evaluating the Reference-Based Golden Speaker (L1-GS)

We constructed the following three systems and compared their performance in generating L1-GS utterances. The objectives of this experiment were to determine the optimal speech embedding, and more importantly, to establish that L1-GS utterances captured the native accent and the L2 speaker identity, which is critical since they would be used as targets for the reference-free FAC task. Details of the model configurations and training are summarized in Appendix A.

- **Senone-PPG**: use the senone-PPG as the input (6024 dimensions).
- **Mono-PPG**: use the monophone PPG as the input (346 dimensions).
- **BNF**: use the bottleneck feature vector as the input (256 dimensions).

To generate the L1-GS utterances for testing, we extracted the three speech embeddings from speaker BDL's test set and drove the systems with their respective input. The output Mel-spectrograms were then converted to speech through the WaveGlow vocoders.

1) Objective Evaluation: In a first experiment, we computed the word error rate (WER) of L1-GS utterances synthesized using each of the three speaker embeddings. In our case, the speech recognizer consisted of the TDNN-F acoustic model combined with an unpruned 3-gram language model trained on the Librispeech transcripts. As a reference, we also computed WERs on test utterances from the L1 speaker (BDL) and the two L2 speakers (YKWK, TXHC). Results are summarized in Table I. L1-GS utterances from the three systems achieve lower WERs than the corresponding utterances from the L2 speakers. Since the acoustic model had been trained on American English speech, a reduction in lower WERs can be interpreted as a reduction in the foreign-accentedness. The BNF system performs markedly better than the other two systems, achieving WERs that are close to those on L1 utterances. The Senone-PPG system performed the worst, despite the fact that it contains the most fine-grained triphone-level phonetic information. We offer an explanation of this result in the discussion.

TABLE I
Word Error Rates (%) on Test Utterances and the Original Speech

	Senone-PPG	*Mono-PPG*	*BNF*	*Original speech*
YKWK	37.56	23.30	**9.50**	45.82
TXHC	28.05	23.53	**7.47**	44.57
Average	32.81	23.42	**8.49**	45.20
BDL	N/A			4.98

TABLE II
Accentedness (The Lower, the Better) and MOS Ratings (The Higher, the Better) of the Golden, Native, and Non-Native Speakers; the Error Ranges Show the 95% Confidence Intervals; the Same Convention Applies to the Rest of the Results

	Senone-PPG	*Mono-PPG*	*BNF*	*Original L2*	*Original L1*
Accentedness	6.01±0.26	5.48±0.19	**4.30±0.16**	6.77±0.20	1.04±0.04
MOS	3.43±0.13	3.54±0.09	**3.78±0.05**	3.70±0.06	4.63±0.06

2) Subjective Evaluation: To further evaluate the three L1-GS systems, we conducted formal listening tests to rate three perceptual attributes of the synthesized speech: accentedness, acoustic quality, and voice similarity. All listening tests were conducted through the Amazon Mechanical Turk platform[11]. Instructions were given in each test to help the participants focus on the target speech attribute. All tests included five calibration samples to detect cheating behaviors, as suggested by Buchholz and Latorre [59]; responses from participants who were deemed to have cheated were excluded. Ratings for the calibration samples were excluded, too. All participants received monetary compensation. All samples were randomly selected from the test set, and the presentation order of samples in every listening test was randomized and counter-balanced. All participants resided in the United States at the time of the recruitment and passed a qualification test where they identified several regional dialects in the United States. All participants were self-reported native English speakers. AQAll listening tests in this study have been approved by the Institutional Review Board of Texas A&M University.

Accentedness test. Listeners were asked to rate the foreign accentedness of an utterance on a nine-point Likert-scale (1: no foreign accent; 9: heavily accented), which is used in the pronunciation training community [60]. Listeners were told that the native accent in this task was General American. Participants ($\mathrm{N} = 20$) rated 20 randomly selected utterances per system per L2 speaker. The utterances shared the same linguistic content in all conditions to ensure a fair comparison. As a reference, listeners also rated the same set of sentences for the L1 and L2 speakers. The results are summarized in the first row of Table II. L1-GS utterances from the three systems were rated significantly ($p \ll 0.001$) more native-like than the original L2 speech, though not as much as the original L1 speech. Among the three systems, the BNF system significantly outperformed

[10][Online]. Available: https://github.com/NVIDIA/waveglow

[11][Online]. Available: https://www.mturk.com

TABLE III
VOICE SIMILARITY RATINGS. THE FIRST ROW SHOWS THE PERCENTAGE OF THE RATERS THAT BELIEVED THE SYNTHESIS AND THE REFERENCE AUDIO CLIP WERE PRODUCED BY THE SAME SPEAKER; THE SECOND ROW IS THE AVERAGE RATING OF THESE RATERS' CONFIDENCE LEVEL WHEN THEY MADE THE CHOICE

	Senone-PPG	*Mono-PPG*	*BNF*
Prefer "same speaker"	70.00±9.12%	71.25±6.38%	**73.75±6.46%**
Average rater confidence	4.82	4.89	**4.93**

Mono-PPG, while Mono-PPG was rated significantly more native-like than Senone-PPG, all with $p \ll 0.001$.

Acoustic quality. Listeners were asked to rate the acoustic quality of an utterance using a standard five-point (1: poor; 2: bad; 3: fair; 4: good; 5: excellent) Mean Opinion Score (MOS) [61]. Participants (N = 20) listened to 20 randomly-selected sentences per L2 speaker per system. As in the accentedness test, listeners also rated the original utterances from the L1 and L2 speakers. The results are summarized in the second row of Table II. As expected, the original native speech received the highest MOS. Among the three golden speaker voices, BNF achieved the highest MOS compared with the other two systems ($p \ll 0.001$). The Mono-PPG system obtained better acoustic quality than the Senone-PPG system ($p = 0.045$). Interestingly, L1-GS utterances from the BNF system received higher MOS than the original L2 speech (3.78 vs. 3.70, $p = 0.02$), a surprising result for which we offer a possible explanation in Section VI.

Voice similarity test. Listeners were presented with a pair of speech samples –an L1-GS synthesis, and the original utterance from the corresponding L2 speaker. In the test, listeners first had to decide if the two samples were from the same speaker, and then rate their confidence level on a seven-point scale (1: not confident at all; 3: somewhat confident; 5: quite a bit confident; 7: extremely confident) [1], [27]. To minimize the influence of accent, the two utterances had different linguistic contents and were played in reverse, following [1]. For each system, participants (N = 20) rated 10 utterance pairs per speaker (20 utterance pairs for each system). Results are summarized in Table III. Across the three systems, more than 70% of the listeners were "quite a bit" confident (4.82-4.93 out of 7) that the L1-GS utterance and the original L2 utterance had the same voice identity. Significance tests showed that there was no statistically significant difference between the preference percentages for the three systems.

These results show that the BNF system outperforms the other two systems significantly in both objective and subjective measures. Therefore, for the remainder of this manuscript, we focus our evaluation on the BNF system, i.e., target L1-GS utterances for the reference-free (pronunciation-correction) system are those from the BNF system.

B. Experiment 2: Evaluating the Reference-Free Golden Speaker (L2-GS)

In the second experiment, we directly converted L2 test utterances with the proposed pronunciation-correction model and compared it against the baseline systems. Detailed model architecture configurations and training setups are included in Appendix B.

- **Baseline 1**: the system of Zhang *et al.* [12], a state-of-the-art VC system capable of modifying segmental and prosodic attributes between different speakers. The loss function of this system was eq. (16), i.e., L_{base}.
- **Baseline 2**: the system of Liu *et al.* [41], the only other reference-free accent conversion system that we are aware of (cf. Section II-C). The audio samples were generated by passing the test set utterances through the Liu system, as prescribed in [41], which was pre-trained on 105 VCTK [62] speakers. The test samples were provided as a courtesy by Liu et al. and we only performed two post-processing steps to ensure a fair comparison. First, we resampled the test samples provided by Liu *et al.* from 22.05 kHz to 16 kHz to match the sampling rate of the other systems. Second, we manually trimmed the trailing white noises in some of the test samples. The accent conversion model was pre-trained on VCTK not L2-ARCTIC, which made its stop-token prediction not stable, and some of the synthesized utterances have a few seconds of white noise after the end of speech.
- **Proposed (without att loss)**: the proposed system without the attention loss term described in eq. (18). We included this variation of the proposed system to study the contribution of adding the backward decoder alone. The loss function of this system was $L_{base} + L_{bwd}$.
- **Proposed**: the proposed system with the full forward-and-backward decoding technique, which included both the backward decoder and the attention loss term. The loss function of this system was eq. (19), i.e., $L_{base} + L_{bwd} + L_{att}$.

For both variations of the proposed system, we performed accent conversion using the backward decoder during testing since it produced significantly better-quality speech compared to the forward decoder on the validation set. Please refer to Appendix C for a qualitative comparison between the two decoders.

1) Objective Evaluations: For objective evaluations, we computed three measures, as suggested by [12], plus WER as a fourth:

- **MCD**: the Mel-Cepstral Distortion [28] between the L2-GS (actual output) and L1-GS speech (desired output). It was computed on time-aligned (Dynamic Time Warping) Mel-cepstra between the L2-GS and the L1-GS audio. Lower MCD correlates with better spectral predictions. We used SPTK [63] and the WORLD vocoder [64] to extract the Mel-cepstra with a shift size of 10 ms.
- **F_0 RMSE**: the F_0 RMSE between the L2-GS and L1-GS speech on voiced frames. Lower F_0 RMSE represents better pitch conversion performance. The F_0 and voicing features were extracted by the WORLD vocoder with the Harvest pitch tracker [65].
- **DDUR**: the absolute difference in duration between the L2-GS and L1-GS speech. Lower DDUR implies better duration conversion performance.
- **WER**: the word error rate for the L2-GS speech. Ideally, the L2-GS speech should have a lower WER than the original

TABLE IV
OBJECTIVE EVALUATION RESULTS OF THE REFERENCE-FREE FAC SYSTEM (PRONUNCIATION-CORRECTION). THE FIRST ROW IN EACH BLOCK SHOWS THE SCORES BETWEEN THE ORIGINAL L2 UTTERANCES AND THE L1-GS UTTERANCES. THE LAST BLOCK SHOWS THE AVERAGE VALUES OF THE FIRST TWO BLOCKS. FOR ALL MEASUREMENTS, A LOWER VALUE SUGGESTS BETTER PERFORMANCE

L2 speaker	*System*	*WER (%)*	*MCD (dB)*	*F_0 RMSE (Hz)*	*DDUR (sec)*
YKWK	Original	45.82	8.07	23.38	1.15
	Baseline 1	41.31	6.26	**18.43**	0.18
	Baseline 2	82.81		N/A	
	Proposed (w/o att loss)	36.12	6.16	19.41	**0.14**
	Proposed	**34.54**	**6.10**	20.78	0.15
	L1-GS	9.50	0.00	0.00	0.00
TXHC	Original	44.57	8.00	25.73	1.29
	Baseline 1	43.67	6.32	**19.40**	0.17
	Baseline 2	84.39		N/A	
	Proposed (w/o att loss)	40.05	**6.26**	22.33	0.15
	Proposed	**37.33**	6.29	21.37	**0.15**
	L1-GS	7.47	0.00	0.00	0.00
Average	Original	45.20	8.04	24.56	1.22
	Baseline 1	42.49	6.29	**18.92**	0.18
	Baseline 2	83.60		N/A	
	Proposed (w/o att loss)	38.09	6.21	20.87	0.15
	Proposed	**35.94**	**6.20**	21.08	**0.15**
	L1-GS	8.49	0.00	0.00	0.00

TABLE V
ACCENTEDNESS (THE LOWER, THE BETTER) AND MOS (THE HIGHER, THE BETTER) RATINGS OF THE REFERENCE-FREE ACCENT CONVERSION SYSTEMS AND ORIGINAL L1 AND L2 UTTERANCES. THE L1-GS SCORES ARE FROM THE BNF RESULTS IN TABLE II, WHICH SERVE AS AN UPPER-BOUND FOR THIS EXPERIMENT, SINCE BASELINE 1 AND THE PROPOSED SYSTEM USED THE L1-GS UTTERANCES AS THEIR TRAINING TARGETS

	Baseline 1	*Baseline 2*	*Proposed*	*L1-GS*	*Original L2*	*Original L1*
Accentedness	5.56±0.23	6.04±0.31	**5.33±0.28**	4.30±0.16	6.58±0.26	1.07±0.04
MOS	2.95±0.12	2.86±0.12	**3.22±0.10**	3.78±0.05	3.68±0.10	4.80±0.06

TABLE VI
VOICE SIMILARITY RATINGS OF THE REFERENCE-FREE ACCENT CONVERSION TASK. THE L1-GS SCORES ARE FROM THE BNF RESULTS IN TABLE III, WHICH SERVE AS AN UPPER-BOUND FOR THIS EXPERIMENT, SINCE BASELINE 1 AND THE PROPOSED SYSTEM USED THE L1-GS UTTERANCES AS THEIR TRAINING TARGETS

	Baseline 1	*Baseline 2*	*Proposed*	*L1-GS*
Prefer "same speaker"	69.25±11.08%	47.50±6.65%	**73.00±7.55%**	73.75±6.46%
Average rater confidence	5.00	4.57	**5.12**	4.93

non-native speech, implying that the conversion reduced the foreign accent.

Results are summarized in Table IV. For all measures, we also computed the scores between the original L2 speech and the L1-GS speech as a reference. In addition, we included the WER of the L1-GS speech as an upper-bound. By definition, the other three measures on the L1-GS speech are all zero. For Baseline 2, we only computed the WER since the system was not trained to predict L1-GS, which makes computing the other objective scores ill-defined.

The two variations of the proposed method obtained better WER, MCD, and DDUR scores, while the Baseline 1 method performed slightly better on the F_0 RMSE. More importantly, Baseline 1 and the two variations of the proposed method were able to reduce the WER of the input L2 utterance. The Proposed method (with attention loss) reduced WERs by 20.5% (relative) on average, which was significantly higher than the WER reduction of the Baseline 1 system (6.0% relative). Baseline 2 performed poorly on the WER metric. Among the two variations of the proposed method, the one that included both the backward decoder and attention loss performed equally-well or better on the WER, MCD, and DDUR scores.

2) Subjective Evaluations: Following the same protocol described in Section V-A2, we asked participants to rate the accentedness, acoustic quality, and voice similarity of synthesized L2-GS utterances. We used the samples from the Proposed system (with the attention loss during training) based on the objective evaluations in the previous section.

Accentedness test. Participants (N = 20) rated 20 random samples per speaker per system, as well as the corresponding original audio. Results are compiled in the first row of Table V. All systems obtained significantly more native-like ratings than the original L2 utterances ($p \ll 0.001$). More specifically, the Baseline 1 system reduced the accentedness rating by 15.5% (relative) and the Baseline 2 system reduced the accentedness rating by 8.2% (relative), while the Proposed system achieved a 19.0% relative reduction, a difference that was statistically significant (Proposed and Baseline 1, $p = 0.04$; Proposed and Baseline 2, $p \ll 0.001$). As expected, the original L1 speech was rated less accented than all other systems.

MOS test. Participants (N = 20) rated 20 audio samples per speaker per system. We used the same MOS test as in experiment 1 to measure the acoustic quality of the synthesis. Results are shown in the second row of Table V. The Proposed system achieved significantly better audio quality than the baselines (9.15% relative improvement compared with Baseline 1; 12.59% relative improvement compared with Baseline 2; $p \ll 0.001$ in both cases).

Voice similarity test. Participants (N = 20) rated 10 utterance pairs per speaker per system (i.e., 20 utterance pairs for each system). This last experiment verified that the accent conversion retained the voice identity of the L2 speakers. The results are shown in Table VI. For Baseline 1 and the Proposed system, the majority of the participants thought the synthesis and the reference speech were from the same speaker, and they were "quite a bit confident" (5.00-5.12 out of 7) about their ratings. Although the Proposed system obtained higher ratings than the Baseline 1 system in terms of voice identity, the difference between the preference percentages was not statistically significant ($p = 0.12$), which was expected. The reason is that the input and output speech had different accents, but very similar voice identity. Therefore, both systems were not trained to modify the voice identity of the input audio. As a result, both the Baseline 1 system and the Proposed system were able to keep the voice identity unaltered during the conversion process. The Baseline 2 system, on the other hand, performed significantly worse than Baseline 1 and the Proposed system in terms of voice similarity; on average, 47.5% of the participants thought that the synthesis and the reference speech were from the same speaker, which is lower than chance level, indicating that the syntheses produced by Baseline 2 did not capture the voice identity of the L2 speakers

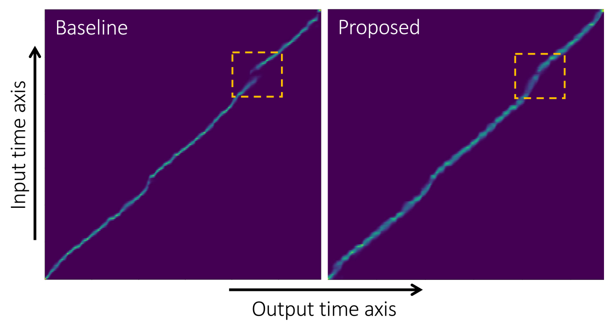


Fig. 6. A qualitative comparison of the attention weights generated by the baseline and the proposed pronunciation-correction systems on one testing utterance.

well. This result echoes with the findings of Liu *et al.* [41], where they also identified voice identify issues of the Baseline 2 system.

Aside from the objective and subjective scores, we provide an example of the attention weights produced by Baseline 1 and the Proposed system on a test utterance in Figure 6. Qualitatively, we can observe that the attention weights of the Baseline 1 system contained an abnormal jump towards the end of the synthesis, while the Proposed system produced smooth alignments at the same time steps. Additionally, the Proposed method appears to have used a broader window to compute the attention context compared with Baseline 1, as reflected by the width of the attention alignment path. Therefore, the Proposed system utilized more contextual information during the decoding process.

VI. Discussion

A. Experiment 1

In experiment 1, we tested three versions of the L1-GS system that used different speech embeddings at the input: senone-PPGs, monophone-PPGs, and bottleneck features (BNFs). Both objective and subjective tests suggested that the BNF system outperforms the other two, both in terms of audio quality and native accentedness. Further, we find that L1-GS utterances on the BNF system achieve similar WERs as the original utterances from the L1 speaker, a remarkable result that further supports the effectiveness of the system in reducing foreign accents. The majority of the human raters (73.75%) had high confidence that the BNF L1-GS shared the same voice identity as the target L2 speaker, suggesting that the accent conversion was also able to preserve the desired (i.e., the L2 speaker's) voice identity. A surprising result from the listening tests is that BNF L1-GS utterances were rated to have higher audio quality than the original natural speech from the L2 speaker. Although this result speaks of the high acoustic quality that the BNF L1-GS system is able to achieve, it is likely that native listeners associated acoustic quality with intelligibility, rating the original foreign-accented speech to be of lower acoustic quality because of that; see Felps *et al.* [1].

Two probable factors explain why BNFs outperformed the other two speech embeddings. First, during the training process, we observed that the BNF system converges to a better terminal validation loss. This result suggests that the speech

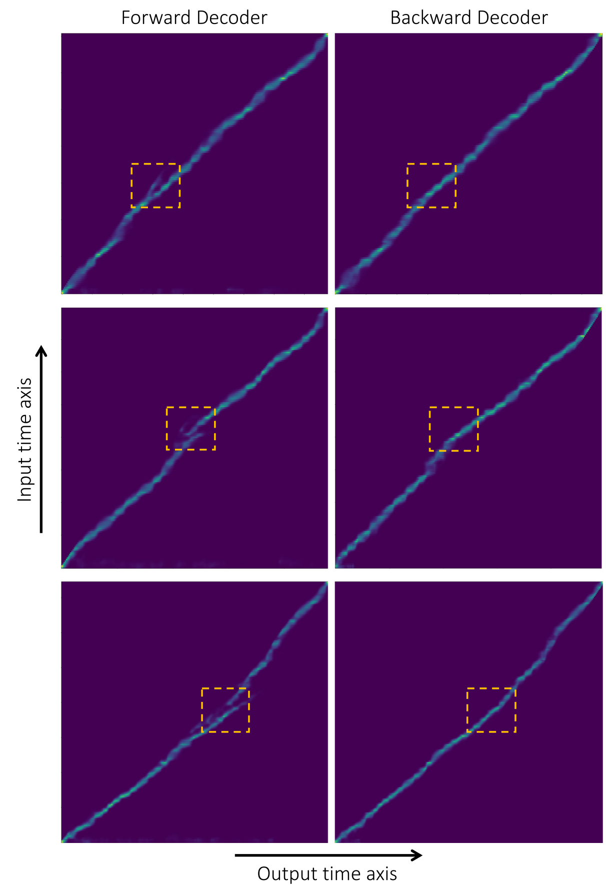


Fig. 7. A qualitative comparison of the attention weights generated by the forward and backward decoders of the proposed pronunciation-correction systems on three utterances from the validation set.

synthesizer can model Mel-spectrograms more accurately using BNFs as the input rather than the other two speech embeddings. Second, although BNFs and PPGs contain similar linguistic information, the process that converted BNFs to PPGs was a phoneme classification task. Therefore, errors that do not exist in BNFs may occur in PPGs due to the enforcement of the extra classification step. Those additional classification errors are then translated to the speech synthesizer as mispronunciations and speech artifacts. One possible explanation for differences between the two PPGs is dimensionality reduction strategies; the monophone-PPG system used an empirical rule (reducing senones to monophones) to summarize the high-dimensional senone-PPG, while the senone-PPG system constructed a learnable transformation (an input PreNet). Although it is possible for data-driven transforms to outperform empirical rules given enough data, the limited amount of data (∼one hour of speech per speaker) available for the FAC task was probably not enough to produce a good transformation for senone-PPGs.

B. Experiment 2

In experiment 2, we achieved reference-free FAC by constructing a pronunciation-correction model that converted L2 utterances directly to match the L1-GS. Our results are encouraging; both the baseline model of Zhang *et al.* [12] (Baseline 1) and our proposed system were able to reduce the foreign accentedness of the input speech significantly, while retaining the voice identity of the L2 speaker. More importantly, the proposed system outperformed the Baseline 1 system significantly in terms of MOS and accentedness ratings. A possible explanation for this result is that the proposed method computes the alignment between each pair of input and output sequences from two directions at training time. Thus, by forcing the forward and the backward decoders to produce similar alignment weights, we force the decoders to incorporate information from both the past and future when generating the alignment. During inference time, only one decoder is needed to perform the reference-free accent conversion; therefore, the proposed system consumes exactly the same amount of inference resources as the baseline system. In summary, the better accentedness and audio quality ratings obtained by the proposed system can largely be attributed to the better alignments provided by the forward-and-backward decoding training technique, as illustrated in Figure 6. The proposed system also outperformed a state-of-the-art reference-free FAC system by Liu *et al.* [41] (Baseline 2) in all objective and subjective evaluation metrics. The comparison of the proposed method and Baseline 2 shows that there is still a large performance gap between a speaker-specific reference-free FAC system (the proposed method) and a many-to-many reference-free FAC system (Baseline 2), which encourages future work in both areas.

The L2-GS generated by the reference-free FAC was rated as significantly less accented than the L2 speaker, though it still had a noticeable foreign accent compared with the original L1 speech. This suggests that the pronunciation-correction model did not fully eliminate the foreign accent in heavily mispronounced or disfluent speech segments, and therefore some foreign-accent cues from the input were carried over to the output speech. One likely explanation for this result is that the proposed reference-free FAC model can only correct error patterns that have occurred in the training data. Due to the high variability of L2 pronunciations, the amount of training data available for each L2 speaker ($\sim$one hour of speech) was not sufficient to cover a portion of the error patterns manifested in the test data, and therefore those errors were not corrected and resulted in the residual foreign accents in the L2-GS utterances. Finally, the MOS ratings of the pronunciation-correction models were lower than those of the BNF L1-GS, which was expected since the output of the pronunciation-correction model is a re-synthesis of the L1-GS utterances.

VII. Conclusion and Future Work

In this work, we propose a new reference-free FAC system[12] that transforms input L2 utterances to reduce their foreign accentedness. This is in contrast to the majority of the existing FAC systems, which require native reference utterances at inference time. Training the system requires two steps. In a first step, we train a FAC model to transform utterances from a reference L1 speaker, so they have the voice identity of the L2 learner. We refer to these transformed utterances as L1-GS utterances. In a second step, we train a pronunciation-correction model that can transform utterances from the L2 learner to match the L1-GS utterances obtained in the first step. Our evaluations indicate that the reference-free FAC system can significantly reduce the foreign accentedness in L2 speech while retaining the voice identity.

[12]Project webpage: https://guanlongzhao.github.io/demo/reference-free-ac

One possible future direction of this work is to use transfer learning [66] to reduce the amount of training data needed for the golden-speaker generation process. This would require first training a multi-speaker speech synthesizer with speech embeddings and speaker embeddings (e.g., i-vectors) as the input, then performing inference using speech embeddings from the reference L1 speaker and the speaker embeddings from the L2 speaker. The benefit of this strategy is that training a multi-speaker speech synthesizer generally only requires a small number of recordings from a particular speaker (e.g., the L2 speaker).

Another future research direction is to improve the quality of the pronunciation-correction model. A direct extension of the current system that might improve the audio quality is to jointly optimize the pronunciation-correction model and the neural vocoder. The current setup of the system trains the WaveGlow model with "clean" original Mel-spectrograms, which leads to a mismatch between the output of the pronunciation-correction model (synthetic Mel-spectrogram) and the expected input of the neural vocoder. Another possibility for quality improvement is to directly convert between foreign-accented and native speech embeddings to correct the mispronunciations. This seems feasible since the speech embeddings (e.g., BNFs) contain rich classifiable phonetic information, which is decoupled from other speaker-specific cues that might interfere with the correction process. The benefits of this approach are two-fold. First, it would eliminate the need to generate the L1-GS, since we can directly use the speech embeddings from L1 teachers as training targets. Second, by combining data from speakers that share the same foreign accent, this approach would enable us to train specific pronunciation-correction models for each first language (e.g., for Chinese L2 learners of English) that can cover more mispronunciation variations compared with speaker-dependent models, as we have done in this current work, thus improving the accentedness ratings of the syntheses. AQFinally, we intend to study other simpler attention regularization techniques [67] as alternatives to the forward-and-backward decoding technique used in this work. A simpler attention regularization technique would help the pronunciation-correction model lower its training cost.

Appendix A
Model Details of the Speech Synthesizers

Table VII summarizes the neural network architectures of the three speech synthesizers. It is worth noting that the input PreNet produced a 512-dim summarization from the Senone-PPG, which is higher than the dimensionality of the Mono-PPG and BNF. We did experiment on a lower dimensionality (256) in the input PreNet, which lead to significant artifacts and

TABLE VII
NEURAL NETWORK ARCHITECTURE OF THE SPEECH EMBEDDING TO MEL-SPECTROGRAM SYNTHESIZERS

Component	***Parameters***
Input-dim	6024 (Senone-PPG); 346 (Mono-PPG); 256 (BNF)
Input PreNet (Senone-PPG only)	Two fully connected (FC) layers, each has 512 ReLU units, 0.5 dropout [71] rate Output-dim: 512
Convolutional layers	Three 1-D convolution layers (kernel size 5) Batch normalization [72] after each layer Output-dim: 512 (Senone-PPG); 346 (Mono-PPG); 256 (BNF)
Encoder	One-layer Bi-LSTM, 256 cells in each direction Output-dim: 512
Decoder PreNet	Two FC layers, each has 256 ReLU units, 0.5 dropout rate Output-dim: 256
Attention LSTM	One-layer LSTM, 0.1 dropout rate Output-dim: 512
Attention layers	v in eq. (5) has 256 dims; eq. (6), $k = 32$, $r = 31$; eq. (10), $w = 20$
Decoder LSTM	One-layer LSTM, 0.1 dropout rate Output-dim: 512
PostNet	Five 1-D convolution layers (kernel size 5), 0.5 dropout rate 512 channels in first four layers and 80 channels in last layer Output-dim: 80

TABLE VIII
NEURAL NETWORK ARCHITECTURE OF THE BASELINE PRONUNCIATION-CORRECTION MODEL

Component	***Parameters***
Input layer	80-dim Mel-spectrum + 256-dim BNF
Encoder	Two-layer Pyramid Bi-LSTM, 256 cells / direction / layer Frame sub-sampling rate: 2 With layer normalization [73] Output-dim: 512
Decoder PreNet	Two FC layers, each has 256 ReLU units, 0.5 dropout rate Output-dim: 256
Attention mechanism	One-layer LSTM Forward-attention technique [51] for attention weights Output-dim: 512
Decoder LSTM	One-layer LSTM Output-dim: 512
PostNet	Five 1-D convolution layers (kernel size 5), 0.5 dropout rate 512 channels in first four layers and 80 channels in last layer Output-dim: 80
Input Phoneme Classifier	One FC layer + softmax Output-dim: 346
Output Phoneme Classifier	One FC layer + softmax Output-dim: 346

mispronunciations. Therefore, we used the current setting for the Senone-PPG system in order to generate intelligible speech syntheses to compare with the other two systems.

The models were trained using the Adam optimizer [68] with a constant learning rate of 1×10^{-4} until convergence, which was monitored by the validation loss. We applied a 1×10^{-6} weight decay [69] and a gradient clipping [70] of 1.0 during training. The batch size was set to 8 and the weight terms w_1 and w_2 in eq. (13) were set to 1.0 and 0.005, based on preliminary experiments [27].

APPENDIX B
MODEL DETAILS OF THE PRONUNCIATION-CORRECTION MODELS

Table VIII summarizes the model details of the Baseline 1 pronunciation-correction model. On top of the Baseline 1 model, the Proposed model adds a backward decoder that has the same structure (attention modules, decoder LSTM, and decoder PreNet) as the Baseline 1 model's decoder. The phoneme prediction ground-truth labels were per-frame phoneme labels (with word positions) that were produced by force-aligning the audio to its orthographic transcriptions. We note that the phoneme predictions were only required in training, not testing. For both models, the training was performed with the Adam optimizer with a weight decay of 1×10^{-6} and a gradient clip of 1.0. The initial learning rate was 1×10^{-3} and was kept constant for the first 20 epochs, then exponentially decreased by a factor of 0.99 at each epoch for the next 280 epochs, and then kept constant at the terminal learning rate. The batch size was 16. The loss term weights w_1, w_2, w_3, and w_4 in equations (16)-(19) were empirically set to 1.0, 0.05, 0.5, and 100.0.

APPENDIX C
QUANTITATIVE COMPARISON BETWEEN THE FORWARD AND BACKWARD DECODER OF THE PROPOSED SYSTEM

As a qualitative comparison between the forward and backward decoder in the proposed system, we plot the attention weights generated by both decoders on a few utterances from the validation set. Good alignment of the attention weights generally indicates better performance. We can see in the figures that the backward decoder produces attention weights that have less discontinuity, which may explain why the backward decoder generates speech with better quality compared to the forward decoder.

ACKNOWLEDGMENT

We would like to thank Dennis R. da Cunha Silva, Adam Hair, and Pedro Moreno for reviewing early versions of the manuscript. We appreciate the help of Songxiang Liu (The Chinese University of Hong Kong) for providing test samples for Baseline 2. We are grateful for the feedback from the reviewers, which helped improve the quality of the manuscript.

REFERENCES

[1] D. Felps, H. Bortfeld, and R. Gutierrez-Osuna, "Foreign accent conversion in computer assisted pronunciation training," *Speech Commun.*, vol. 51, no. 10, pp. 920–932, 2009.
[2] K. Probst, Y. Ke, and M. Eskenazi, "Enhancing foreign language tutors-in search of the golden speaker," *Speech Commun.*, vol. 37, no. 3/4, pp. 161–173, 2002.
[3] S. Ding *et al.*, "Golden speaker builder-an interactive tool for pronunciation training," *Speech Commun.*, vol. 115, pp. 51–66, 2019.
[4] R. Wang and J. Lu, "Investigation of golden speakers for second language learners from imitation preference perspective by voice modification," *Speech Commun.*, vol. 53, no. 2, pp. 175–184, 2011.

[5] O. Turk and L. M. Arslan, "Subband based voice conversion," in *Proc. 7th Int. Conf. Spoken Lang. Process.*, 2002, pp. 289–292.

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

[7] 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 *Proc. Interspeech*, 2015, pp. 299–303.

[8] 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 *Proc. Interspeech*, 2019, pp. 4115–4119.

[9] J. Shen *et al.*, "Natural TTS synthesis by conditioning wavenet on mel spectrogram predictions," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2018, pp. 4779–4783.

[10] F.-L. Xie, F. K. Soong, and H. Li, "A KL divergence and DNN-based approach to voice conversion without parallel training sentences," in *Proc. Interspeech*, 2016, pp. 287–291.

[11] G. Zhao and R. Gutierrez-Osuna, "Using phonetic posteriorgram based frame pairing for segmental accent conversion," *IEEE/ACM Transactions on Audio, Speech, Lang. Process.*, vol. 27, no. 10, pp. 1649–1660, Oct. 2019.

[12] J.-X. Zhang, Z.-H. Ling, Y. Jiang, L.-J. Liu, C. Liang, and L.-R. Dai, "Improving sequence-to-sequence acoustic modeling by adding text-supervision," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2019, pp. 6785–6789.

[13] M. Mimura, S. Sakai, and T. Kawahara, "Forward-backward attention decoder," in *Proc. Interspeech*, 2018, pp. 2232–2236.

[14] Y. Zheng *et al.*, "Forward-backward decoding for regularizing end-to-end TTS," in *Proc. Interspeech*, 2019, pp. 1283–1287.

[15] R. Prenger, R. Valle, and B. Catanzaro, "WaveGlow: A flow-based generative network for speech synthesis," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2019, pp. 3617–3621.

[16] S. H. Mohammadi and A. Kain, "An overview of voice conversion systems," *Speech Commun.*, vol. 88, no. 88, pp. 65–82, 2017.

[17] M. Brand, "Voice puppetry," in *Proc. 26th Annu. Conf. Comput. Graph. Interactive Techn.*, 1999, pp. 21–28.

[18] D. Felps, C. Geng, and R. Gutierrez-Osuna, "Foreign accent conversion through concatenative synthesis in the articulatory domain," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 8, pp. 2301–2312, Oct. 2012.

[19] S. Aryal and R. Gutierrez-Osuna, "Reduction of non-native accents through statistical parametric articulatory synthesis," *J. Acoustical Soc. Amer.*, vol. 137, no. 1, pp. 433–446, 2015.

[20] S. Aryal and R. Gutierrez-Osuna, "Data driven articulatory synthesis with deep neural networks," *Comput. Speech Lang.*, vol. 36, pp. 260–273, 2016.

[21] B. Denby and M. Stone, "Speech synthesis from real time ultrasound images of the tongue," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, vol. 1, 2004, pp. I– 685.

[22] R. Mumtaz, S. Preuß, C. Neuschaefer-Rube, C. Hey, R. Sader, and P. Birkholz, "Tongue contour reconstruction from optical and electrical palatography," *IEEE Signal Process. Lett.*, vol. 21, no. 6, pp. 658–662, Jun. 2014.

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

[24] S. Aryal and R. Gutierrez-Osuna, "Can voice conversion be used to reduce non-native accents," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2014, pp. 7879–7883.

[25] G. Zhao, S. Sonsaat, J. Levis, E. Chukharev-Hudilainen, and R. Gutierrez-Osuna, "Accent conversion using phonetic posteriorgrams," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2018, pp. 5314–5318.

[26] T. Hazen, W. Shen, and C. White, "Query-by-example spoken term detection using phonetic posteriorgram templates," in *Proc. IEEE Workshop Automat. Speech Recognit. Understanding*, 2009, pp. 421–426.

[27] G. Zhao, S. Ding, and R. Gutierrez-Osuna, "Foreign accent conversion by synthesizing speech from phonetic posteriorgrams," in *Proc. Interspeech*, 2019, pp. 2843–2847.

[28] T. Toda, A. W. Black, and K. Tokuda, "Voice conversion based on maximum-likelihood estimation of spectral parameter trajectory," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 15, no. 8, pp. 2222–2235, Nov. 2007.

[29] Z. Wu, E. S. Chng, and H. Li, "Exemplar-based voice conversion using joint nonnegative matrix factorization," *Multimedia Tools Appl.*, vol. 74, no. 22, pp. 9943–9958, 2015.

[30] G. Zhao and R. Gutierrez-Osuna, "Exemplar selection methods in voice conversion," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2017, pp. 5525–5529.

[31] K. Kobayashi, T. Toda, G. Neubig, S. Sakti, and S. Nakamura, "Statistical singing voice conversion with direct waveform modification based on the spectrum differential," in *Proc. Interspeech*, 2014, pp. 2514–2518.

[32] S. H. Mohammadi and A. Kain, "Voice conversion using deep neural networks with speaker-independent pre-training," in *Proc. IEEE Spoken Lang. Technol. Workshop*, 2014, pp. 19–23.

[33] L. Sun, S. Kang, K. Li, and H. M. Meng, "Voice conversion using deep bidirectional long short-term memory based recurrent neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2015, pp. 4869–4873.

[34] H. Miyoshi, Y. Saito, S. Takamichi, and H. Saruwatari, "Voice conversion using sequence-to-sequence learning of context posterior probabilities," in *Proc. Interspeech*, 2017, pp. 1268–1272.

[35] J. Zhang, Z. Ling, L.-J. Liu, Y. Jiang, and L.-R. Dai, "Sequence-to-sequence acoustic modeling for voice conversion," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 27, no. 3, pp. 631–644, Mar. 2019.

[36] A. v. d. Oord *et al.*, "WaveNet: A generative model for raw audio," in *Proc. Int. Sci. Community Assoc. Workshop Speech Synth.*, 2016, Paper 125.

[37] J. Lorenzo-Trueba *et al.*, "The voice conversion challenge 2018: Promoting development of parallel and nonparallel methods," in *Proc. Odyssey Speaker Lang. Recognit. Workshop*, 2018, pp. 195–202.

[38] J. Zhang, Z. Ling, and L.-R. Dai, "Non-parallel sequence-to-sequence voice conversion with disentangled linguistic and speaker representations," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 28, pp. 540–552, 2019.

[39] K. Tanaka, H. Kameoka, T. Kaneko, and N. Hojo, "AttS2S-VC: Sequence-to-sequence voice conversion with attention and context preservation mechanisms," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2019, pp. 6805–6809.

[40] H. Kameoka, K. Tanaka, T. Kaneko, and N. Hojo, "ConvS2S-VC: Fully convolutional sequence-to-sequence voice conversion," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 28, pp. 1849–1863, 2020, doi: 10.1109/TASLP.2020.3001456.

[41] S. Liu *et al.*, "End-to-end accent conversion without using native utterances," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2020, pp. 6289–6293.

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

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

[44] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 4, pp. 788–798, May 2011.

[45] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2015, pp. 5206–5210.

[46] D. Povey *et al.*, "The Kaldi speech recognition toolkit," in *Proc. IEEE Workshop Automat. Speech Recognit. Understanding*, 2011.

[47] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 577–585.

[48] L.-J. Liu *et al.*, "Non-parallel voice conversion with autoregressive conversion model and duration adjustment," in *Proc. Joint Workshop Blizzard Challenge Voice Convers. Challenge*, 2020, pp. 126–130.

[49] Y. Wang *et al.*, "Tacotron: Towards end-to-end speech synthesis," in *Proc. Interspeech*, 2017, pp. 4006–4010.

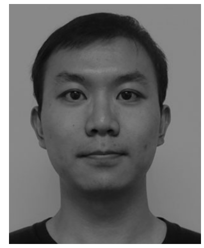
[50] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2016, pp. 4960–4964.

[51] J.-X. Zhang, Z.-H. Ling, and L.-R. Dai, "Forward attention in sequence-to-sequence acoustic modeling for speech synthesis," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2018, pp. 4789–4793.

[52] S. Ruder, "An overview of multi-task learning in deep neural networks," 2017, *arXiv:1706.05098*.

[53] Y. Zhang and Q. Yang, "A survey on multi-task learning," *IEEE Trans. Knowl. Data Eng.*, to be published, doi: 10.1109/TKDE.2021.3070203.

[54] D. P. Kingma and P. Dhariwal, "Glow: Generative flow with invertible 1x1 convolutions," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 10 215–10 224.
[55] J. Kominek and A. W. Black, "The CMU arctic speech databases," in *Proc. 5th Int. Sci. Community Assoc. ITRW Speech Synth. Workshop*, Pittsburgh, PA, 2004, pp. 223–224.
[56] G. Zhao *et al.*, "L2-ARCTIC: A non-native english speech corpus," in *Proc. Interspeech*, 2018, pp. 2783–2787.
[57] *Audacity Team*, "Audacity(R): Free audio editor and recorder [computer application]," 2020. [Online]. Available: http://www.audacityteam.org/
[58] A. Paszke *et al.*, "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 8024–8035.
[59] S. Buchholz and J. Latorre, "Crowdsourcing preference tests, and how to detect cheating," in *Proc. Interspeech*, 2011, pp. 3053–3056.
[60] M. J. Munro and T. M. Derwing, "Foreign accent, comprehensibility, and intelligibility in the speech of second language learners," *Lang. Learn.*, vol. 45, no. 1, pp. 73–97, 1995.
[61] ITU-T, Recommendation P.800.1.mean opinion score (MOS) terminology, International Telecommunication Union-Telecommunication Standardisation Sector (ITU-T), March 2003.
[62] J. Yamagishi, C. Veaux and K. MacDonald, "CSTR VCTK corpus: English multi-speaker corpus for CSTR voice cloning toolkit (version 0.92)," 2019. [Online]. Available: https://datashare.ed.ac.uk/handle/10283/3443
[63] SPTK Working Group, "Speech signal processing toolkit (SPTK) version 3.11," 2017. [Online]. Available: http://sp-tk.sourceforge.net/
[64] M. Morise, F. Yokomori, and K. Ozawa, "World: A vocoder-based high-quality speech synthesis system for real-time applications," *IEICE Trans. Inf. Syst.*, vol. 99, no. 7, pp. 1877–1884, 2016.
[65] M. Morise, "Harvest: A high-performance fundamental frequency estimator from speech signals," in *Proc. Interspeech*, 2017, pp. 2321–2325.
[66] Y. Jia *et al.*, "Transfer learning from speaker verification to multispeaker text-to-speech synthesis," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 4485–4495.
[67] M. He, Y. Deng, and L. He, "Robust sequence-to-sequence acoustic modeling with stepwise monotonic attention for neural TTS," in *Proc. Interspeech*, 2019, pp. 1293–1297, doi: 10.21437/Interspeech.2019-1972.
[68] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, 2015.
[69] A. Krogh and J. A. Hertz, "A simple weight decay can improve generalization," in *Proc. Ad. Neural Inf. Process. Syst.*, 1992, pp. 950–957.
[70] S. Kanai, Y. Fujiwara, and S. Iwamura, "Preventing gradient explosions in gated recurrent units," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 435–444.
[71] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.
[72] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.
[73] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *CoRR*, 2016, *arXiv:1607.06450*.



Guanlong Zhao received the B.S. degree in applied physics from the University of Science and Technology of China, Hefei, China, in 2015 and the Ph.D. degree in computer science from Texas A&M University, College Station, TX, USA, in 2020. He is currently a Software Engineer with Google, where he is working on speech recognition model training infrastructure, automation, and internationalization. His research interests include speech synthesis, acoustic modeling, voice conversion, and accent conversion.

Shaojin Ding received the B.S. degree in automation from Xi'an Jiaotong University, Xi'an, China, in 2015. He is currently working toward the Ph.D. degree in computer science with Texas A&M University, College Station, TX, USA. His research interests include speech synthesis, voice conversion, and speaker recognition.

Ricardo Gutierrez-Osuna (Senior Member, IEEE) received the B.S. degree in electrical engineering from the Polytechnic University of Madrid, Madrid, Spain, in 1992, and the M.S. and Ph.D. degrees in computer engineering from North Carolina State University, Raleigh, NC, USA, in 1995 and 1998, respectively. He is currently a Professor with the Department of Computer Science and Engineering, Texas A&M University, College Station, TX, USA. His current research interests include voice and accent conversion, speech and face perception, wearable physiological sensors, and active sensing.