Improving mispronunciation detection using speech reconstruction

Anurag Das, and Ricardo Gutierrez-Osuna, *Senior Member, IEEE*

***Abstract*—Multi-task learning shows that solving two related machine learning problems simultaneously can improve performance compared to solving them independently. This paper examines whether combining mispronunciation detection and diagnosis (MDD) with text-to-speech (TTS) synthesis can improve MDD accuracy. We propose a MTL model that uses predicted phone sequences from an MDD system and feeds them to a TTS system to reconstruct the original speech. We hypothesize that a multi-objective loss that combines the speech reconstruction error (TTS loss) with the MDD error (i.e., predicted vs. ground-truth phoneme sequences) will boost MDD performance. To test this hypothesis, we compare the proposed sequential MDD system against that of an identical MDD system without the speech reconstruction loss and a state-of-the-art MDD baseline. When evaluated on unseen test sentences, the sequential system achieves higher MDD scores than the other two systems, which suggests that the joint loss helps the system generalize to new test sentences. Further, we examine whether the TTS system can reconstruct non-native accented speech when the predicted phoneme sequence from the MDD has mispronunciations. Results from perceptual listening experiments show that speech generated from non-native phoneme sequences is rated as more accented and less intelligible than that from native phoneme sequences, which suggests that, when trained jointly with an MDD system, the TTS system can capture differences in phoneme sequences between native and non-native speech.**

***Index Terms*—Multi-task learning, mispronunciation detection, text-to-speech, speech reconstruction.**

I. Introduction

MISPRONUNCIATION detection and diagnosis (MDD) can be used to provide automatic feedback to help second language learners (L2) improve their pronunciation. An MDD system takes speech from an L2 learner as input and identifies parts of an utterance that were mispronounced. As such, MDD systems can provide personalized pronunciation feedback at scale, addressing the scarcity of qualified teachers in this critical area of competency for L2 speakers.

Current MDD systems are based on deep learning (DL) techniques. Given sufficient training data, DL models can extract phonological rules from the speech and avoid the need to hand-craft these rules [1] [2]. Further, DL-based sequence-to-sequence (seq2seq) models can also learn to align the speech signal and the target phones via an attention mechanism, thus avoiding the need to perform forced alignment, which is challenging with non-native speech. As an example, Leung et al. [3] proposed a DL model that uses convolutional and recurrent layers with a Connectist Temporal Classification loss [4] to map Mel-spectrograms directly into phone sequences, and showed their model outperforms prior methods that rely on forced alignment. Prior research [5] has also shown that adding text as a feature along with acoustic features can further improve MDD performance. Further performance improvements can be achieved by combining text and acoustic inputs with phonetic embeddings extracted from an automatic speech recognition (ASR) system [6]. Despite these advances, the highest MDD accuracy reported on the L2-ARCTIC corpus [7] (80%) is substantially lower than that on ASR tasks.

Research in multi-task learning (MTL) shows that solving two related tasks simultaneously can improve performance on both tasks. For example, training ASR and automatic speaker verification (ASV) systems simultaneously can improve ASR accuracy [8]; similar results are reported when training ASR and TTS simultaneously [9]. Given that MDD and ASR are related tasks (i.e., both predict phone sequences from input speech), we predict that MDD performance will also improve when combined with a TTS downstream. In particular, we hypothesize that using a TTS system to reconstruct the original speech from the predicted phone sequence, and using the reconstruction loss as additional feedback to the MDD system can help reduce errors in the predicted phone sequence.

To test this hypothesis, we propose a MTL model for MDD that predicts phone transcriptions from the input speech and reconstructs the original speech from them. Our model takes an input speech signal and extracts a wav2vec latent representation [10]. Then, a seq2seq model combines this wav2vec embedding with the original text (which is generally available in pronunciation training tasks) to predict the phone sequence. Finally, a Tacotron-based [11] TTS system reconstructs the original speech from the predicted phone sequence and a speaker embedding from a pre-trained ASV system. The MDD system is trained by combining the phone prediction loss with the speech reconstruction loss. This approach is akin to invertible networks such as CycleGAN [12], which use the predicted output to reconstruct the input signal. Our approach is also related to Karita *et al.* [9], which combines ASR and TTS tasks to improve ASR accuracy. However, their system combines the two tasks in parallel, whereas our approach combines MDD and TTS in a sequential manner. Since the reconstructed speech also captures segmental information, we also hypothesize that segmental differences between phones with and without mispronunciations will be reflected in the reconstructed speech. To test this secondary hypothesis, we compare perceptual ratings of accentedness and intelligibility when speech is reconstructed using either canonical phone sequences (i.e., without mispronunciations) or

Anurag Das 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: adas@tamu.edu; rgutier@tamu.edu).

phone sequences manually annotated by a linguist (i.e., which capture mispronunciations).

II. Background

A. Mispronunciation detection

Traditional MDD methods evaluate mispronunciation using likelihood-based scores, such as the well-known Goodness of Pronunciation (GoP) score [13]. To calculate a GOP score, an L2 learner's utterance first needs to be force-aligned against the canonical phone sequence, Then, a GOP score is computed for each phone as the ratio of the posterior probability of the canonical phoneme relative to the phone with the highest posterior. A phone is deemed to have been mispronounced by comparing the GOP score to a pre-determined threshold optimized through cross-validation. Several modifications of the GoP score have been proposed since its original publication, including scaled log posteriors [14], sub-phonemic transition probability from an Hidden Markov Model (HMM) [15], and classifier-based methods that use features from phone segments to determine if the pronunciation is correct or not. As an example of the latter approach, Hu *et al.* [16] proposed a transfer learning-based method where an acoustic model was first pre-trained on all phones, and then logistic regression classifiers, one for each phone, were built to detect mispronunciations. The proposed method improved mispronunciation detection by 7.4% precision and recall when compared to a GMM-HMM baseline. Other classification methods that have been explored include decision trees [17], [18], and Support Vector Machines (SVMs) [19]. These methods can detect mispronunciations by comparing the target phone to the most likely pronounced phone, but are unable to diagnose the errors.

An alternative to GOP-based scoring methods are techniques that use use Extended Recognition Networks (ERNs). An ERN is a pre-designed set of phonological rules that model L2 learners' typical mispronunciations, and can be used in the ASR decoder lattice. Rules for ERNs can be divided into two categories: hand-crafted rules and data-driven rules. As an example of a hand-crafted rule, Meng *et al.* [20] derived mispronunciations made by Chinese learners of English by comparing their L2 productions against the corresponding native productions. They also used a pronunciation lexicon to generate additional erroneous variants of words. Harrison *et al.* [1] used a hand-crafted lexicon containing the correct pronunciations of a word as well as the mispronunciation variants in the L2 learners' speech to detect and diagnose errors. Data-driven techniques have also been used to derive ERNs automatically from transcribed L2 speech and their corresponding canonical pronunciations [2]. Qian *et al.* [21] developed a two-pass framework to avoid the need to explicitly model errors . However, obtaining building ERNs is a time-consuming process that often requires access to a large amount of speech data. An additional problem with ERN-based approaches is that they rarely cover all possible pronunciation rules. As a result, these systems can only detect mispronunciations that are included in the rule set.

To overcome the limitations of ERNs, DL models have been a major focus of recent research in MDD. These systems can identify errors in pronunciation in a data-driven fashion and do not require explicit pronunciation rules. Li *et al.* [22] developed an acoustic graphemic phonetic model (AGPM) to predict likely pronunciations from acoustic features, graphemes, as well as canonical transcriptions using a multi-distribution DNN. The network outperformed an existing ERN based method with a reduction in false rejection and false acceptance rate by 6.4% and 12.9% respectively. In later work, Mao [23] used an MTL model to separately recognize correct pronunciations and mispronunciation from input acoustic features and phones. The model outperformed an existing network that also processed acoustic and phonetic features. End-to-end models for MDD have also been explored to avoid the need for forced alignment. Leung *et al.* [3] developed an end-to-end MDD system (CNN-RNN-CTC) that uses 1D CNN layers and a bi-directional Gated Recurrent Unite (GRU) with a CTC loss [4] to predict the L2 learners' phone sequences. Their proposed system outperformed an ERN as well as the DL-based system (APGM) by 44.75% and 2.77% respectively, both of which relied on force alignment. To improve on the CNN-RNN-CTC model, Feng *et al.* [5] proposed an SED-MDD model that takes a Mel-spectrogram of the input speech an the text (which the L2 learner was asked to produce) as inputs, and outputs the corresponding phone sequence. The SED-MDD model uses an Encoder-Decoder architecture with an attention mechanism and a cross-entropy loss. The model outperformed the CNN-RNN-CTC model by 18% in both correctness and accuracy. More recent studies have also used text as an additional input for MDD. Fu *et al.* [24] aligned acoustic features with canonical phone sequences from a sentence encoder using an attention mechanism and then decoded the pronounced phone sequence. To improve phone predictions, the authors also proposed a data-augmentation technique that randomly replaced vowels and consonants with alternate vowels and consonants. The system achieved a higher F measure of 56.08% compared to 49.29% for the CNN-RNN-CTC model. Ye *et al.* [6] combined text, acoustic features and phonetic features from an acoustic model as inputs to predict the target phone sequence. Adding all three inputs improved accuracy and F1 score by 9.93% and 6.17%, respectively, compared to a baseline that only used acoustic features and linguistic features as inputs. These studies show that using additional features as inputs boosts MDD performance. Further improvements in MDD accuracy have also been reported when replacing MFCC-based encodings with advanced feature representations such as wav2vec embeddings [25] [26] and fine-tuning on the dataset, training the fine-tuned wav2vec model on pseudo-labels [27] obtained from non-annotated sentences, and using Transformer architectures [28]. The main conclusion from these studies is that using text as an additional input does improve MDD accuracy. Our study contributes to this prior research by examining whether adding a reconstruction loss from a TTS synthesizer that consumes the predicted phone sequences to resynthesize the original utterance can further improve MDD accuracy.

B. Text-to-speech

The goal of TTS is to synthesize speech waveforms from text inputs. Early TTS systems were based on concatenative techniques [29]. A concatenative TTS system identifies speech segments (or units) in a database that match the input text, and concatenates these segments to produce a speech waveform. However, concatenative TTS systems require access to a large database of recordings of speech units, and the synthesized voices generally contain noticeable discontinuities and lack naturalness. Concatenative TTS gave way to statistical parametric speech synthesis (SPSS) methods, which leveraged prior work on HMM-based ASR [30], [31]. These systems generate acoustic parameters associated with the target text, and then recover speech from those acoustic parameters. Synthesized speech from SPSS system does sound more natural than that of concatenative approaches, but it still suffers from artifacts such as muffled sounds or noise.

State-of-the-art TTS systems are based on DL techniques. These systems generally consist of two components: a neural front-end that maps characters or phonemes to intermediate features such as Mel-spectrograms, and a synthesizer that converts those intermediate features into speech waveforms. A primary example of DL-based TTS is Tacotron [11]. Tacotron takes characters/phones directly as inputs, which simplifies the text analysis modules. To generate audio, Tacotron outputs are then passed to a neural synthesizer such as WaveNet [32]. Later systems such as Deep Voice 2 [33] and Deep Voice 3 [34] have replaced Tacotron with a fully convolutional neural front-end, which has been shown to improve the acoustic quality of the output waveforms. Shen *et al.* [35] used a similar architecture to synthesize speech that rivaled the acoustic quality of natural speech. However, these models have slow inference times due to their auto-regressive nature: a Mel-spectra depends on previously generated Mel-spectra [36]. To reduce inference time, more recent TTS models avoid auto-regression by generating sequences in parallel. As an example, FastSpeech [36] uses a Transformer architecture to achieve a 38x speedup in inference speed over auto-regressive TTS synthesis. Parallel Tacotron [37] uses self-attention and a Variational Autoencoder (VAE) style residual encoder. The system achieved comparable naturalness and much faster inference compared to a Tacotron 2 baseline. However, a limitation of these neural TTS systems is that they use a two-stage pipeline, which requires fine-tuning to generate high-quality speech [35].

C. Multi-task learning in speech

Multi-task learning (MTL) involves training a neural network for multiple different but related tasks. Typically, each MTL network has one primary task and one or more complementary auxiliary tasks. In speech research, MTL has found wide application across multiple speech tasks, including ASR [38], [39], TTS [40], [41], ASV [42], [43], and emotion recognition [44], [45].

In ASR, the primary task is to recognize phones and words, but auxiliary tasks may also include recognition of gender [38], phonetic units [39], and symbolic units such as graphemes [46]. It has been shown that predicting phonological features (e.g., manner and place of articulation) along with phoneme labels can improve ASR and phone recognition accuracy [47]. Most methods compute the auxiliary loss from the output of the final layer in the architecture, but the auxiliary loss may also be computed from intermediate layers. For example, Krishna *et al.* [48] used a hierarchical MTL framework to predict words and phones from intermediate layers of the encoder. The hierarchical framework reduced word error rates (WER) compared to a single task baseline on the Switchboard evaluation set [49]. Computing the loss at intermediate layers can also avoid the need to label intermediate sub-word units such as phones or characters [50]. MTL has also been used in multilingual ASR. Huang *et al.* [51] proposed a multilingual DL model that shared hidden layers across languages, followed by separate softmax layers for each language. Sharing the hidden layers was shown to improve ASR for each language. Likewise, Hou *et al.* [52] proposed a hybrid CTC/attention-based MTL framework that used language identification as an auxiliary task. The authors showed that after transferring a model pre-trained on 42 languages to 14 low-resource languages reduced the word error rate (WER) compared to a randomly initialized model.

In TTS, linguistic and/or acoustic features are generally used to predict the parameters that a vocoder uses to synthesize the speech waveform. Several MTL approaches have been proposed for this purpose. Hu *et al.* [40] developed a monolingual TTS system that predicted both the spectral envelope and the log amplitude of the output speech. Their results showed that combining the two tasks improved accuracy on both. Riberio *et al.* [41] used a wavelet-based decomposition of f0 as a secondary task. Listeners rated utterances from the MTL model as more natural (45% preference) than those from the TTS model (36.5% preference) without f0 decomposition. Stacked bottleneck features from a DL model have also been shown to improve synthesis performance. For example, Wu *et al.* [53] used MTL to predict both acoustic features and vocoder features, and found that using bottleneck features improved the naturalness of the synthesized speech. Gains in TTS quality have also been reported by combining phoneme classification with regression on the spectral parameters of a vocoder. For example, Toth *et al.* [54] showed that this joint training improved the performance of both tasks compared to solving each task separately.

MTL has also been used in emotion recognition from speech. Parthasarathy *et al.* [44] developed a model that jointly predicted arousal, valence, and dominance. Though these emotion attributes are generally assumed to be orthogonal, the authors showed that joint prediction improves accuracy when compared to predicting each attribute individually. Li *et al.* [45] showed that predicting the speaker's gender as a secondary task improved emotion recognition by 7.7%. Cai *et al.* [55] reported noticeable improvements in emotion classification from speech when using ASR as an auxiliary task.

MTL has also been shown to improve ASV tasks. As an example, Yu *et al.* [56] showed that using an MTL model for joint ASV (primary task) and ASR as a secondary task im-

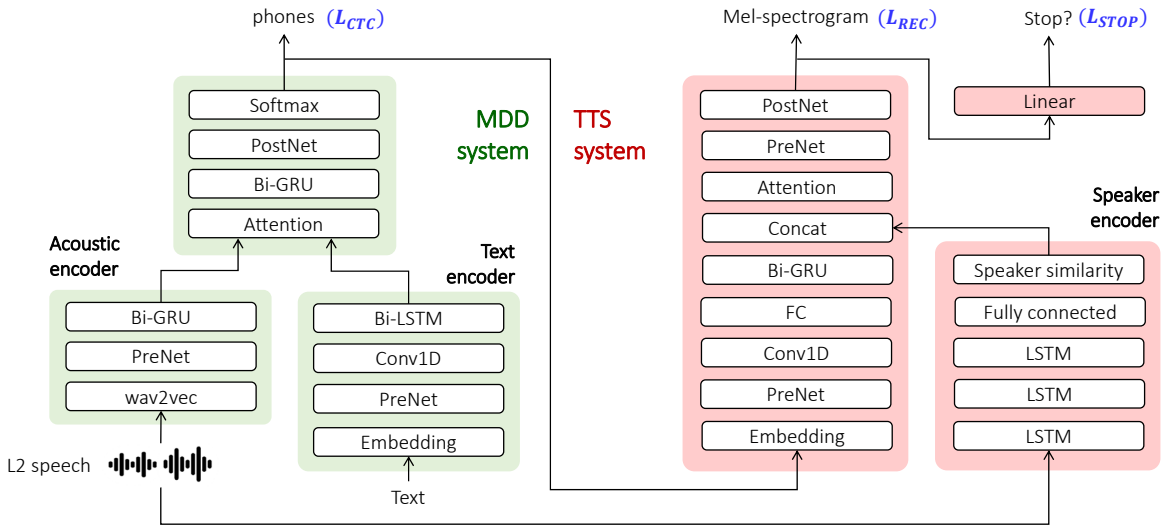

Fig. 1. Block diagram of the proposed sequential model comprising of an MDD system (left, in green) and a TTS system (right, in red). The MDD system predicts the phone sequence from an input utterance and its text, and passes the predicted phone sequence to the TTS system to reconstruct the input utterance. The MDD system is trained to minimize the phone-recognition loss (L_{CTC}) and the speech reconstruction loss (L_{REC}) jointly.

proved ASV accuracy compared to a single-task ASV baseline. Ding *et al.* [42] showed that adding a generative adversarial network (GAN) improved ASV performance compared to an i-vector baseline and a triplet loss baseline. Sigtia *et al.* [43] showed that training a model to perform ASV and voice trigger detection achieves similar performance as dedicated baseline models for each task.

III. Methods

The workflow of the proposed sequential system is shown in Figure 1. The model consists of five modules: (1) a wav2vec feature extractor that generates a latent representation for the input utterance, (2) a speaker encoder that generates an embedding that captures the voice characteristics of the speaker for ASV purposes, (3) a text encoder that generates an embedding of the input text sequence, (4) a seq2seq model that consumes the latent speech representation, the text embedding and the speaker embedding and produces the phones contained in the speech signal, and (5) a TTS module that combines the predicted phone sequence and an embedding of the target speaker to synthesize the speech waveform. In this fashion, the seq2seq system is forced to produce the phone sequence that matches the one produced by the source speaker.

Before training the sequential model, we fine-tune a wav2vec-based acoustic model on the L2-ARCTIC corpus [7], and pre-train the speaker encoder and TTS system. Then, we extract features from the acoustic model and the speaker encoder and use them to train the sequential model. At this point, we freeze the weights of the TTS system, and only learn the weights of the seq2seq model, guided by the TTS loss and the MDD loss.

A. wav2vec 2.0 model

The wav2vec 2.0 model consists of CNN and Transformer layers. Raw audio is first sent to a CNN to obtain a latent representation Z, which is then discretized via a learnable codebook to obtain Q. In parallel, random segments from Z are masked and fed to the Transformer layers to generate a contextualized representations C. Let us denote the values of Z and C at time t by z_t and c_t, respectively, and the corresponding Q vector at time t by q^+ and all other time steps by q^-. The model is trained with a contrastive loss to maximize the similarity between c_t and q^+, and minimize the similarity between c_t and q^-. The total loss is the weighted sum of the contrastive loss and the codebook loss.

B. Speaker Encoding

Our speaker encoder is based on the ASV model in [57] and [58], which prior work on accent conversion has shown to generalize to unseen speakers [59]. The workflow of the speaker encoder is illustrated in Figure 1. It takes 40-dimensional Mel-spectrograms with a 25ms window and a 10ms step as input, and passes it to a 3-layer long-short-term memory (LSTM) with 256 hidden units and a fully connected (FC) layer with 256 nodes. The output of the FC layer is passed through a rectified linear unit (ReLU) activation to generate a sparse embedding. The hyperparameters of the speaker encoder are summarized in Table III. For each enrolled speaker, we create a template by computing speaker embeddings of a few utterances. At runtime, we extract a speaker embedding from the input utterance and compare it against the templates using generalized end-to-end loss (GE2E) [57]. If the similarity between the embedding and the template is above a threshold, the speaker is verified. During training, the model computes embeddings $\boldsymbol{e_{ij}} (1 \leq i \leq N, 1 \leq j \leq M)$ of M utterances from N speakers. A speaker embedding is derived for each speaker: $\boldsymbol{c_i} = \boldsymbol{1/M} \sum_{j=1}^{M} \boldsymbol{e_{ij}}$. The similarity matrix $\boldsymbol{S_{ij,k}}$ is obtained by comparing all embeddings $\boldsymbol{e_{ij}}$ against every

speaker embedding $\boldsymbol{c_k}$, $1 \leq k \leq N$ in the batch using scaled cosine similarity:

$$\boldsymbol{S_{ij,k}} = w.cos(\boldsymbol{e_{ij}}, \boldsymbol{c_k}) + b = w.\boldsymbol{e_{ij}}.||\boldsymbol{c_k}||_2 + b \quad (1)$$

where w and b are parameters learned by the model, and the cosine similarity is the dot product between the two embeddings. The model outputs high similarity when utterance matches the speaker ($i = k$) and low similarity ($i \neq k$) otherwise. In our experiments, we remove the similarity matrix computation and directly use the 256-dimensional feature as the speaker embedding. To obtain an utterance-level embedding for an unknown speaker, we split a test utterance into 1.6-sec segments with a 50% overlap, and feed each segment to the encoder. Finally, we average resulting outputs and normalize them to produce the final embedding for the utterance.

C. Text Encoding

As shown in Figure 1, the text encoder takes text sequences as inputs and generates a text representation that is then fed to the seq2seq model. Following [5], we convert a text sequence into a sequence of integer IDs corresponding to characters in the text, and then pass the latter to a Pre-Net consisting of two FC layers with ReLU activation and 0.5 dropout. The Pre-Net acts as an information bottleneck, which helps the model converge and generalize [5], [11]. Following [11], we set the number of hidden layers in the second layer of the PreNet to half of those in the previous layer. The PreNet output is consumed by a CNN-RNN module that produces the final output of the text encoder. The CNN-RNN module consists of five CNN layers with ReLU and batch normalization [60] and a bidirectional LSTM layer. The rationale behind using CNN layers is that they model longer-term context (e.g. N-grams) in the input text sequence.

D. Mispronunciation Detection and Diagnosis

Our MDD system is based on a seq2seq model proposed by Feng *et al.* [5]. The seq2seq model consumes wav2vec embeddings from the input utterance along with the text sequence, and produces the corresponding phone sequence as the output. We pass wav2vec embeddings to an acoustic encoder consisting of an embedding layer, followed by a PreNet layer with two FC layers with 0.5 dropout rate and ReLU activation, followed by two bidirectional GRUs layers with 256 units each.

We align the outputs of the audio and text encoders using a Bahdanau attention mechanism [61]. Denoting the last hidden layer of the text encoder as $(\boldsymbol{t_1}, \boldsymbol{t_2}, \ldots, \boldsymbol{t_{T_S}})$ and the last hidden state of the audio encoder output as $(\boldsymbol{a_1}, \boldsymbol{a_2}, \ldots, \boldsymbol{a_{T_A}})$, we compute attention scores as:

$$score(\boldsymbol{t_i}, \boldsymbol{a_j}) = v \times tanh(W_1 \boldsymbol{t_i} + W_2 \boldsymbol{a_j}) \quad (2)$$

$$\boldsymbol{\alpha_{ij}} = softmax(score(\boldsymbol{t_i}, \boldsymbol{a_j})) \quad (3)$$

$$\boldsymbol{c_j} = \sum_{i=1}^{T_s} \boldsymbol{\alpha_{ij}} \boldsymbol{t_i} \quad (4)$$

$$\boldsymbol{a'_j} = concat(\boldsymbol{c_j}, \boldsymbol{a_j}) \quad (5)$$

where v, W_1, and W_2 are model parameters. We normalize the cross-attention scores between the acoustic encoder output and the text encoder using softmax and scale them with the text encoder's output to create a context vector $\boldsymbol{c_j}$. We then concatenate $\boldsymbol{c_j}$ with the acoustic encoder's GRU output $\boldsymbol{a_j}$ and pass them to the decoder GRU followed by a post-processing net with similar CNN-RNN architecture as the Text encoder except it has 512 units instead of 1,024 units. Finally, we use a softmax layer to classify the output of the post-processing net into phone sequences.

E. Text-to-speech

The last component of our sequential model is a seq2seq TTS system that takes phones produced by the MDD system and the speaker's embedding as input and generates Mel spectrograms. The TTS architecture follows Wang *et al.* [11], which has been shown to generate natural-sounding speech. A detailed list of hyperparameters is shown in Table III.

The encoder of the TTS system converts an input phone sequence into a hidden representation. We pass the phone sequence predicted by the MDD to an Embedding layer followed by a PreNet, as in the Text Encoder. Then, we pass the PreNet output through five 1D convolutional layers with Max pooling and batch normalization followed by four FC layers and a bidirectional GRU. To capture the voice characteristics of the target speaker, we concatenate the encoder output (which only contains text information) with the target speaker's embedding from the pre-trained speaker encoder (see Section III-B) and feed it to the decoder.

The decoder uses a location-sensitive attention mechanism [62] to generate Mel-spectrogram frames in an auto-regressive manner. At each decoding step, we pass the predicted Mel-spectrogram from the previous time step to a PreNet consisting of two FC layers. Then, we apply an attention mechanism between the PreNet output and the hidden representation from the Encoder to generate a context vector. We then concatenate the PreNet output with the context vector and feed it to a 2-layer LSTM . Finally, we pass the output of the second LSTM layer to an 80-dim linear layer to generate Mel-spectrograms. An additional layer predicts the stop token to decide the end of decoding using binary cross entropy.

To improve synthesis quality, we pass the Mel-spectrogram through a post-processing net that predicts the residual. Following [11], the post-processing net consists of five 1D convolution layers, four FC layers, and a bidirectional GRU. In a final step, we add the residual back into the predicted Mel-spectrogram to obtain the final prediction.

We train the sequential model to minimize two losses: the Euclidean distance between predicted and ground truth Mel-spectrograms (L_{REC}) (TTS loss) and the CTC (L_{CTC}) loss between predicted and ground truth phones (MDD loss).

$$\begin{aligned} L_{REC} &= ||\hat{\boldsymbol{y}}_{\boldsymbol{post}} - \boldsymbol{y}||_2^2 + ||\hat{\boldsymbol{y}} - \boldsymbol{y}||_2^2 \\ L_{STOP} &= CrossEntropy(\boldsymbol{t}, \hat{\boldsymbol{t}}) \\ L_{CTC} &= CTC(\hat{\boldsymbol{p}}, \boldsymbol{p}) \\ L &= \alpha(L_{REC} + \lambda \times L_{STOP}) + (1 - \alpha) L_{CTC} \end{aligned} \quad (6)$$

TABLE I
DISTRIBUTION OF TRAIN, TEST, AND VALIDATION SENTENCES FOR SEEN AND UNSEEN CONDITIONS

		Train		Test	Validation
		L2-ARCTIC	TIMIT	L2-ARCTIC	L2-ARCTIC
Seen	Speakers	18	630	6	6
	Utterances	2,520	6,300	60	60
	Duration	2.6 hr	4.5 hr	3.4 min	3.7 min
Unseen	Speakers	18	630	6	6
	Utterances	2,340	6,300	60	60
	Duration	2.4 hr	4.5 hr	3.4 min	3.7 min

where $\hat{\boldsymbol{y}}_{\boldsymbol{post}}$ is the output of the post-processing net, $\hat{\boldsymbol{y}}$ is the predicted Mel-spectrogram before the post-processing net, $\hat{\boldsymbol{t}}$ and $\boldsymbol{t}$ are the predicted and ground-truth stop tokens, respectively, $\hat{\boldsymbol{p}}$ and $\boldsymbol{p}$ are the predicted and ground-truth phone sequences, respectively.

IV. EXPERIMENTAL SETUP

A. Speech corpora and training-test splits

We used two publicly available datasets for our experiments: TIMIT [63], and L2-ARCTIC [7]. TIMIT is a native English corpus consisting of 6,300 audio recordings from 630 speakers, with a total duration of 4.5 hours of speech. L2-ARCTIC is a non-native corpus consisting of 24 speakers, four speakers from six different first languages (Hindi, Korean, Mandarin, Spanish, Arabic and Vietnamese). Both corpora include annotated phone transcriptions that can be used for MDD. However, TIMIT uses 61 phones whereas L2-ARCTIC uses 48 phones. To unify the two phone sets, we map TIMIT's 61 phones and L2-ARCTIC's 48 phones into 40 phones (including silence) following [14]. We also resample the L2-ARCTIC recordings (48KHz) down to 16KHz to match TIMIT's sampling rate.

We train our system on the entire TIMIT corpus and 18 speakers from L2-ARCTIC, leaving one speaker from each accent for testing (NJS: Spanish; TXHC: Mandarin; SVBI: Hindi, YKWK: Korean; ZHAA: Arabic; HQTV: Vietnamese). We consider two testing conditions: seen and unseen. In the seen condition, test sentences are also used for training (test speakers are always unseen). In the unseen condition, we exclude test sentences from the training set (i.e., unseen speakers *and* sentences). Validation sentences are *never* part of the training and test sets. Table I shows the breakdown of the train, test and validation sentences for each condition. For each test speaker, we used 10 sentences for testing, 10 sentences for validation, and the remaining 130 sentences for training. To reduce variance, we use the same test and validation sentences for all speakers. Following [64]. we select sentences using a greedy forward selection method that maximizes the entropy of the phonetic transcriptions (i.e., phonetic balance). First, we select the sentence with the highest entropy for training. Conditioned on this sentence, we select the sentence with the highest entropy out of the remaining ones for validation. Finally, we select the next most phonetically balanced sentence (conditioned on the previously selected sentences) for testing. We repeat this procedure until all sentences are covered.

B. wav2vec 2.0 model

We used a wav2vec 2.0 model that had been pre-trained for phone recognition on 54.2 thousand hours from the Librispeech, and LibriVox corpora [65], and fine-tuned it on the training set from L2-ARCTIC following the procedure described in [27][1]. The fined-tuned wav2vec 2.0 model generates a 768-dimensional embedding, which we then reduced down to 300 dimensions using two linear layers.

C. Speaker encoder

We use a pre-trained speaker encoder trained on 1.2k speakers from VoxCeleb1, 6k speakers from VoxCeleb2 [66], and 2.4k speakers from LibriSpeech, or approximately 3,000 hours of speech. It takes 40 channel Mel-spectrograms as input and outputs a 256-dimensional speaker embedding. We obtain Mel-spectrograms using 25 ms analysis windows with a step of 10 ms. As previously described in section III-B, we split speech samples into 1.6-sec windows with 50% overlap and pass them to the encoder. To obtain the final speaker embedding, we L2-normalize each individual window and then compute their average. The model achieves an equal error rate (EER) of 4.5% on the combined test sets of Librispeech, VoxCeleb1, and VoxCeleb2 [58].

D. Multi-task and single-task MDD systems

We train our multi-task MDD model (MT-MDD) and an equivalent single-task MDD model without TTS (ST-MDD) using a batch size of 96 and an Adam optimizer [67] with a learning rate of 10^{-3}. We empirically set the hyperparameter α for the MT-MDD system (see Eq. 6) to 0.2. For both systems, we used an early stopping of 15 epochs. We implemented both systems in PyTorch and trained them on an NVIDIA GeForce RTX 3090 GPU.

E. Text-to-speech

We train the Tacotron-based TTS system on L2-ARCTIC on the same train-test split described in section IV-D to be consistent with the MDD system. We set the batch size to 32 and the initial learning rate to 10^{-3} annealed to 10^{-5} using an exponential decay. We implement the model using PyTorch and stop training after 300k steps on an NVIDIA GeForce RTX 3090 GPU. The TTS system generates Mel-spectrograms.

F. Vocoder

To generate speech waveforms from TTS Mel-spectrograms, we use a WaveRNN vocoder that had been pre-trained on Librispeech [68]. WaveRNN generates speech waveforms with audio quality comparable to WaveNet but with a significantly shorter inference time (13x) [69].

[1] https://github.com/Mu-Y/mpl-mdd

V. RESULTS

We compared the performance of the proposed system (MT-MDD) against its single-task counterpart (ST-MDD) and a state-of-the-art MDD baseline [27]. The baseline model uses wav2vec as a self-supervised model and then fine-tunes using pseudo-labels obtained from 17,403 unannotated sentences from all train speakers in L2-ARCTIC and annotated sentences using the configuration in Table I. Unlike in the original implementation of the baseline model, we do not remove duplicate silences during pre-processing –for consistency with the pre-processing steps in the MT-MDD and ST-MDD models.

We evaluated the three MDD models on mispronunciation *detection* and mispronunciation *diagnosis* tasks, first when test sentences were seen during training, and then when test sentences were unseen. This allowed us to assess their relative generalization properties. We also evaluated speech reconstruction performance of the MT-MDD system using objective and subjective evaluations. For objective evaluation, we examined speaker embeddings for recorded and reconstructed speech. For subjective evaluations, we conducted perceptual listening test of accentedness and intelligibility.

A. MDD performance on seen sentences

In a first experiment, we evaluated the three MDD systems on test sentences that were seen during training. In mispronunciation *detection*, the goal is to identify if a phone has been mispronounced (i.e., a binary classification task). In mispronunciation *diagnosis* the goal is to identify which phone was actually produced (a multi-class task).

Mispronunciation detection. To compute a mispronunciation detection score, we align the annotated and predicted phone sequences using the Needleman-Wunsch algorithm [70]. Then, we count the number of insertions, deletions, and substitution errors. Following prior work [5], we use accuracy and correctness as *detection* metrics:

$$correctness = \frac{N - S - D}{N} \tag{7}$$

$$accuracy = \frac{N - S - D - I}{N} \tag{8}$$

where N is the number of phones, S is the number of substitutions, D is the number of deletions, and I is the number of insertions. We aggregate results across the six test subjects from L2-ARCTIC, and report error bars. Figure 2 shows the accuracy and correctness scores of the three systems. MT-MDD achieves an average correctness of 0.901, while ST-MDD and baseline achieve an average correctness of 0.876 and 0.880, respectively, both of which are significantly lower than that of the MT-MDD system (one-tail t-test $p < 0.05$). We observe a similar trend for accuracy. Notice that accuracy is lower than correctness due to the inclusion of insertion errors. MT-MDD achieves an average accuracy of 0.879, whereas ST-MDD and baseline achieve average accuracy of 0.856 ($p < 0.05$) and 0.860 ($p < 0.05$), respectively. Because MT-MDD outperforms ST-MDD and baseline in correctness and accuracy, this indicates that adding TTS as a secondary task is beneficial.

TABLE II
METRICS FOR MISPRONUNCIATION DIAGNOSIS. CANONICAL TRANSCRIPTION FROM A LEXICON. HUMAN ANNOTATIONS IN L2-ARCTIC. MACHINE ANNOTATION FROM MDD

		Canonical	Human	Machine
Correct pronunciation	True accept (TA)	α	α	α
	False reject (FR)	α	α	β
Incorrect pronunciation	False accept (FA)	α	β	α
	Correct diagnosis (CD)	α	β	β
	Diagnosis error (DE)	α	β	γ

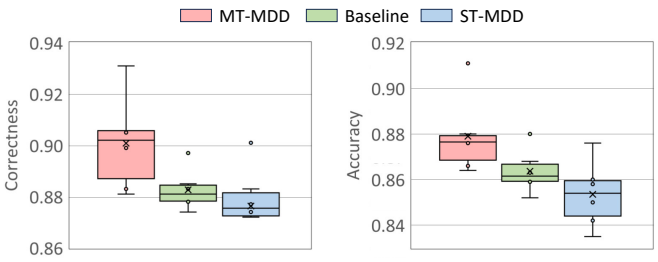

Fig. 2. Mispronunciation *detection* performance (correctness, accuracy) of the three systems on seen sentences

Mispronunciation detection. Following [26], we computed diagnosis metrics by comparing MDD phone predictions against canonical transcriptions from a lexicon and human annotations in L2-ARCTIC –see Table II for definitions. When the human-annotation and canonical phone match, we denote it as a *correct pronunciation*. In this case, whenever the machine annotation (MDD prediction) matches the canonical and human annotation, we refer to it as true accept (TA); otherwise, we refer to it a false reject (FR). Likewise, when the canonical and human annotations do not match, we denote it as a *mispronunciation*. In this case, if the machine and human annotation match, we refer to it as a true reject (TR), and false accept (FA) otherwise. We further divide TR into correct diagnosis (CD) and diagnosis error (DE). From these, we compute Precision (P), recall (R), and $F1$ score as:

$$P = \frac{TR}{TR + FR} \tag{9}$$

$$R = \frac{TR}{FA + TR} \tag{10}$$

$$F1 = \frac{2PR}{P + R} \tag{11}$$

We also compute false rejection rate (FRR), false acceptance rate (FAR), and detection error rate (DER) as follows:

$$FRR = \frac{FR}{TA + FR} \tag{12}$$

$$FAR = \frac{FA}{FA + TR} \tag{13}$$

$$DER = \frac{DE}{CD + DE} \tag{14}$$

Conceptually, FRR measures the number of correct phones that MDD flags as mispronunciations, FAR measures the

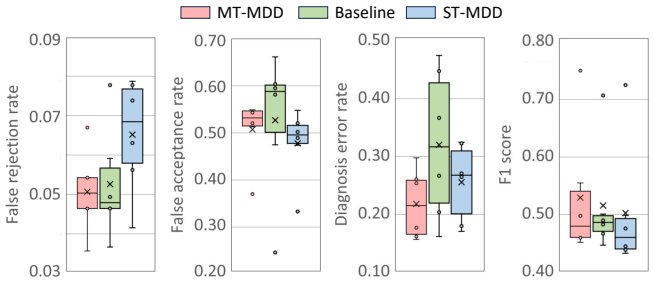

Fig. 3. Mispronunciation *diagnosis* performance of the three systems on seen sentences

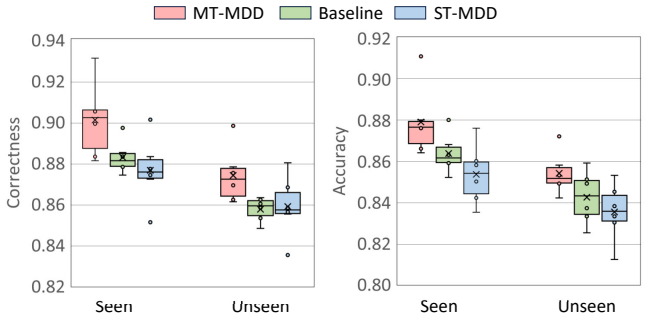

Fig. 4. Mispronunciation *detection* performance of the three systems on *seen* and *unseen* sentences

number of mispronounced phones that MDD flags as a correct phone, and DER measures the number of MDD predictions that do not match either the canonical or human-annotated phones. Results are summarized in Figure 3. MT-MDD achieves an average FRR of 0.050, whereas ST-MDD and baseline achieve average FRR of 0.065 ($p < 0.05$) and 0.052 ($p = 0.40$), respectively. MT-MDD achieves an average FAR of 0.505, whereas ST-MDD and baseline achieve average FAR of 0.475 ($p = 0.25$) and 0.525 ($p = 0.39$), respectively. In terms of DER, MT-MDD achieves an average score of 0.214, whereas ST-MDD and baseline achieve DER of 0.252 ($p = 0.16$) and 0.316 ($p < 0.05$), respectively. For F1 scores, MT-MDD achieves an average of 0.525, whereas ST-MDD and baseline achieve averages of 0.498 ($p = 0.35$) and 0.511 ($p = 0.41$), respectively. In summary, MT-MDD outperforms ST-MDD and baseline in terms of mispronunciation *detection* (i.e., correctness and accuracy), when considering mispronunciation *diagnosis* (FRR, FAR, DER, and F1 scores), MT-MDD outperforms ST-MDD and performs comparably to baseline in FRR. It also outperforms baseline and performs comparably to ST-MDD in DER, and trends favorably for F1.

B. MDD performance on unseen sentences

Evaluating MDD performance on test sentences that were not seen during training represents a more flexible scenario for pronunciation training where L2 learners practice on a broad range of sentences than those in the training set. This condition is also more challenging, but allows us to evaluate the generalization properties of the MDD system.

Mispronunciation detection. The correctness and accuracy of the three systems are summarized in Figure 4. MT-MDD achieves average correctness of 0.874, whereas ST-MDD and baseline achieve average correctness of 0.859 ($p < 0.05$) and 0.860 ($p < 0.05$), respectively. MT-MDD achieves an average accuracy of 0.854, whereas ST-MDD and baseline achieve an average accuracy of 0.835 ($p < 0.05$) and 0.842 ($p = 0.06$), respectively. In summary, MT-MDD has higher correctness than ST-MDD and baseline. It also has a higher accuracy, but differences are only significant with respect to ST-MDD .

To examine the relative contributions of type of model (MT-MDD, ST-MDD, baseline) and type of sentence (seen, unseen), we conducted a two-way ANOVA on accuracy and correctness with model and sentence as independent factors. For correctness, we found a main effect for model ($p < 0.05$) and sentence ($p < 0.05$) but no interaction effects ($p = 0.67$). For accuracy, we also found a main effect for model ($p < 0.05$) and sentence ($p < 0.05$) but no interaction effects ($p = 0.83$). Thus, model and sentence factors provide independent contributions to correctness and accuracy, Finally, to examine generalization performance, we conducted post-hoc tests between MT-MMD on *unseen* sentences and ST-MDD and baseline on *seen* sentences. A one-tailed t-test reveals no statistically significant differences in correctness between MT-MDD on *unseen* sentences, and ST-MDD and baseline correctness on *seen* sentences ($p = 0.38$ and $p = 0.10$, respectively). Likewise, we find no statistically-significant differences in accuracy between MT-MDD on *unseen* sentences, and ST-MDD and baseline on *seen* sentences ($p = 0.47$, and $p = 0.06$, respectively). In sum, adding TTS as a secondary task to an MDD task results in correctness and accuracy on *unseen* sentences that are comparable to those of single-task MDD on *seen* sentences, indicating that MTL improves generalization in MDD tasks.

Mispronunciation diagnosis. Figure 5 shows the average FRR, FAR, DER and F1 scores for the three systems. MT-MDD system achieves an average FRR of 0.072, whereas ST-MDD and baseline achieve average FRR of 0.078 ($p = 0.19$) and 0.079 ($p = 0.20$), respectively. MT-MDD system achieves an average FAR of 0.469, whereas ST-MDD and baseline achieve average FAR of 0.460 ($p = 0.43$) and 0.485 ($p = 0.41$), respectively. MT-MDD system achieves an average DER of 0.288, whereas ST-MDD and baseline achieve average DER of 0.365 ($p = 0.29$) and 0.313 ($p = 0.06$), respectively. Finally, MT-MDD system achieves an average F1 score of 0.494, whereas ST-MDD and baseline achieve average F1 scores of 0.475 ($p = 0.38$) and 0.493 ($p = 0.49$), respectively. Thus, MT-MDD performs comparably to ST-MDD and baseline in mispronunciation *diagnosis*, although results show a favorable trend towards MT-MDD. As in the case of mispronunciation detection, we examined the relative contributions of the type of model and the type of sentence on mispronunciation diagnosis. For FRR, we found a main effect for sentence ($p < 0.05$) but none for model ($p = 0.18$) and interaction ($p = 0.42$). For FAR, we did not find any effects for model ($p = 0.70$), sentence ($p = 0.40$), or interaction ($p = 0.96$). Likewise, for DER, we found a main effect for model ($p < 0.05$) and sentence ($p < 0.05$) but no interaction

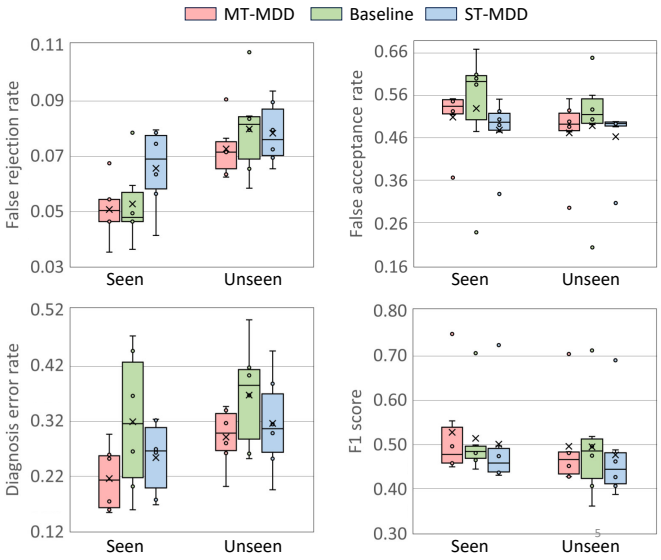

Fig. 5. Mispronunciation *diagnosis* for the three systems on *seen* and *unseen* sentences

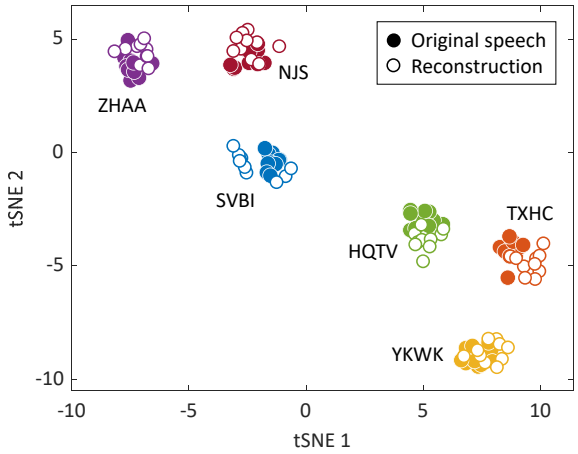

Fig. 6. Speaker embeddings of original (color filling) and reconstructed (white filling) speech for six *unseen* speakers. Original speech and its reconstruction cluster together, indicating the TTS model is able to capture the speakers' identity.

($p = 0.93$). Finally, for F1 scores we found no effects for model ($p = 0.87$), sentence ($p = 0.50$) or interaction ($p = 0.99$). Thus, model and sentence have independent contributions in DER, but not for the other three measures of mispronunciation *diagnosis*. Finally, we examined generalization performance on mispronunciation *diagnosis*. For FRR, we found no statistically significant differences between MT-MDD on *unseen* sentences and ST-MDD ($p = 0.18$) on *seen* sentences, and significant differences between MT-MDD and baseline on *seen* sentences ($p = 0.01$). Similarly, we found no statistically significant differences in FAR between MT-MDD on *unseen* sentences, and ST-MDD and baseline systems on *seen* sentences ($p = 0.45$, and $p = 0.23$, respectively). A similar trend was observed in differences in DER between MT-MDD on *unseen* sentences and ST-MDD and baseline on *seen* sentences ($p = 0.16$, and $p = 0.32$, respectively). Finally, we found no significant differences in F1 between MT-MDD on *unseen* sentences and ST-MDD and baseline systems ($p = 0.47$, and $p = 0.38$, respectively) on *seen* sentences. Therefore, with the exception of FRR, MT-MDD performs comparably to the ST-MDD and baseline systems in diagnosis metrics, again indicating that adding TTS as secondary task improves generalization.

C. TTS performance

Our final set of evaluations focuses on the auxiliary TTS task of our proposed MT-MDD system –note that ST-MDD and baseline do not have a TTS task. We evaluate TTS performance using both objective and subjective tests. For the objective evaluation, we examined the voice quality of the reconstructed speech using t-distributed stochastic neighbor embedding (t-SNE) [71]. For the subjective evaluation, we performed perceptual listening test of accentedness and intelligibility for two types of synthetic speech: (1) using canonical annotations, and (2) using manually-annotated transcriptions. We hypothesized that using canonical transcriptions would synthesize speech that was more native-accented and intelligible than when using manually-annotated transcriptions (i.e., the actual phones L2 learners produced). Following prior work [72] [73], we conduted listening tests of intelligibility and accentedness on Amazon Mechanical Turk (AMT).

Objective evaluation: speaker embeddings To examine whether the TTS system can capture voice quality, we visualize the speaker embeddings of the original speech and speech reconstructions from MT-MDD using t-SNE scatterplots. Results are shown in Figure 6 for the 10 test sentences of the 6 test speakers. Each utterance is color coded by the speaker identity, color filling and white filling indicating original and reconstructed utterances, respectively. The t-SNE plots reveal a strong clustering according to speakers, indicating that the TTS system is able to capture the voice characteristics of each speaker. More importantly, original and reconstructed utterances are in close proximity within each speaker cluster, indicating that the speech reconstructions have similar voice quality as the original utterances.

Subjective evaluation of foreign accentedness. To participate in the listening tests, participants were required to pass a qualification test that required them to discriminate different accents of American English including Northeast, Southern, and General American [74]. Further, all participants resided in the United States and identified as native English speakers. Prior to performing the evaluations, participants were provided calibration samples. Then, they were asked to rate synthesized utterances that had been randomly selected from test sentences. All experiments were approved by the Institutional Review Board at Texas A&M University.

We synthesized two types of utterances for the listening tests: using manual annotations of test sentences as inputs to the TTS block, and using canonical phone sequences generated from the original sentence by an English grapheme-to-phoneme (g2p) [2]. Since manual annotations capture speakers'

[2]We use the g2p system reported in: https://github.com/Kyubyong/g2p

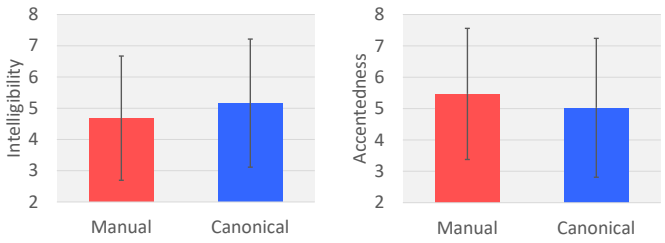

Fig. 7. Intelligibility and accentedness ratings of speech reconstructions from canonical g2p transcriptions and manually-annotated transcriptions with segmental mispronunciations.

mispronunciations, we expected the corresponding reconstructions would be rated as more accented than those from the English g2p phone sequence.

Listeners (N=20) were asked to rate the degree of accentedness of an utterance on a nine-point Likert scale (1: no foreign accent, 9: heavy foreign accent). Listeners were told that the native accent in the task was General American, and were instructed to ignore voice quality/timbre, and only pay attention to the accent. Participants (N=20) rated 72 randomly selected utterances, 36 generated from canonical and manually-annotated phones each, 6 sentences per test speaker. Results are summarized in Figure 6. Utterances synthesized from canonical transcriptions were rated as being less accented (5.03) than those synthesized from manually-annotated phones (5.47). A two-tailed t-test revealed that these differences are statistically significant ($p < 0.001$). These results indicate that the TTS system can capture segmental differences in the phonetic sequences of (canonical) native speech and non-native speech (with its unique mispronunciations), which is essential for the TTS reconstruction loss to have any positive effect on the accuracy of the MDD system upstream.

Subjective evaluation of intelligibility. Following the same recruitment and qualification criteria of the accentedness test, we recruited a different group of participants (N=20) to rate the degree of intelligibility of an utterance using a nine-point Likert scale (1: not intelligible; 9: highly intelligible). Participants rated the same 72 randomly selected utterances used in the accentedness test. Results are summarized in Figure 6. Utterances synthesized from canonical transcriptions were rated as being significantly more intelligible (5.16) than those synthesized from manually-annotated phones (4.68) ($p < 0.001$). These results corroborate those on the accentedness test, indicating that TTS reconstructions have lower intelligibility if the phonetic transcriptions reflect the mispronunciations that are present in speech productions from L2 learners.

VI. Discussion

Prior work on mispronunciation detection shows that adding text (that non-native learners are asked to produce) as an additional input improves MDD performance [5]. This article examined whether resynthesizing the non-native learner's production from the predicted MDD phone sequences would provide further improvements in MDD performance. To test this hypothesis, we proposed a multi-task learning method that combines MDD and speech reconstruction tasks. Results from our experiments indicate that adding a speech reconstruction block does indeed improve mispronunciation detection performance, when compared to single-task MDD models.

We analyzed the mispronunciation *detection* performance of the three systems on test sentences that had not been used for training (unseen condition). Our proposed MT-MDD model achieves significantly higher correctness and accuracy scores than the ST-MDD system. Since MT-MDD and ST-MDD models share the same seq2seq backbone network, improvements in correctness and accuracy can only be attributed to the secondary TTS task. The proposed MT-MDD model also achieves higher correctness and accuracy than a baseline system, which is remarkable considering the baseline system had been fine-tuned on the pseudo-labels.

We also evaluated the three systems on mispronunciation *diagnosis* measures of FRR, FAR, DER, and F1 score. We found no significant differences in diagnosis scores between MT-MDD and ST-MDD and baseline. Though not statistically-significant, MT-MDD has higher a FAR than ST-MDD, which indicates that MT-MDD has a higher tendency to flag mispronunciations as correct pronunciations. This result may be due to the large class imbalance between correct productions and mispronunciations in the dataset, and could be traced back to human annotations in L2-ARCTIC, which tended to focus on the most severe mispronunciations in each sentence. As a result, MT-MDD tends to be conservative, which leads to opposite trends in FRR and FAR. Similar findings have been reported in the literature [28] as a trade-off between FRR and FAR, which are complementary measures. In general, pronunciation training experts tend to favor low FRR scores at the cost of higher FAR, since flagging correct pronunciations as mispronunciations can be detrimental to the L2 learner [75], [76]. The same trade-off is also seen in the baseline system, which has a low FRR and a higher FAR compared to the ST-MDD system. These results indicate that MT-MDD and baseline have a tendency to predict pronunciation errors as correct pronunciations, compared to ST-MDD.

We repeated the above evaluations on sentences that had not been used for training (unseen condition). In this case, MT-MDD outperformed ST-MDD and baseline on mispronunciation *detection* performance. More importantly, MT-MDD performance on unseen sentences was similar to that of ST-MDD and baseline on seen sentences, a clear indication that adding TTS as a secondary task improves the generalization properties of the primary task (MDD), thus validating the main hypothesis of this study.

We also evaluated the performance of the proposed MT-MDD model on the secondary task (speech reconstruction). For this purpose, we used t-SNE to analyze speaker embeddings from original utterances and their TTS reconstructions. We observed that speaker embeddings from reconstructed speech are close to those of the original utterances, with a clear clustering effect between speakers. Finally, we examined if speech reconstructions from text, with and without mispronunciations, led to differnces in perceived accentedness and intelligibility. When we feed canonical phone sequences to the TTS system, listeners rate reconstructions as less accented

and more intelligible than those produced when we feed human annotations. This result indicates the TTS system is sufficiently sensitive to phone sequences that contain actual mispronunciations, as is needed for the reconstruction loss to guide the MDD system towards higher performance in mispronunciation detection and diagnosis.

A. Future Work

Though our proposed MT-MDD model outperforms the ST-MDD and baseline models, it is important to note some of its limitations. First, our speech reconstructions do not faithfully capture the duration of the original utterance, which otherwise would provide additional cues to non-native accents. A potential solution is to include a prosody encoder in the TTS system to condition the duration of speech reconstructions and better match the duration of the original utterance [77]. This would be important when a pronunciation training tool aims to improve the *prosody* of L2 speakers. However, when the objective of pronunciation training is to improve *segmental* productions (i.e., reduce substitutions, deletions and additions), it is not that clear that a prosody encoder would improve mispronunciation *detection* and *diagnosis*.

A major issue for research in MDD systems is the lack of large speech corpora with manually-annotated phonetic transcriptions. Our results suggest a potential venue to address this problem through data augmentation. Given that our TTS system can resynthesize accented speech (if the phonetic sequences include mispronunciations), it may be possible to generate an arbitrarily large corpus of L2 speech by manually introducing segmental errors into the phonetic transcriptions. This idea has been previously suggested in the literature. For example, Korzewka *et al.* [78] introduced mispronunciations (insertions, substitutions, and deletions) into phone sequences, and then synthesized the corresponding speech using a pre-trained TTS system. Then, they used the synthetic speech containing manually-inserted phone errors to train an MDD model. The authors reported an increase in MDD accuracy at the *word level*, which opens the possibility for improvements in MDD at the *phone level*, which is generally the main target in pronunciation training.

The MT-MDD model has nearly twice the number of parameters of the ST-MDD model due to the TTS block, which can make it more challenging for deployment. However, current workstations are well equipped to handle much larger models, so this issue is not a primary concern at this point. In fact, with proper regularization techniques (e.g., pruning, dropout), the increased model size often leads to better generalization performance, a far more important criterion than storage requirements. Further, note that the TTS block is only required for training, but not once the MDD system is deployed. If MDD model size became an issue (i.e., on mobile devices), lighter TTS models such as Deep Voice [34] may be used.

VII. Appendix

Table III shows the hyperparameters of the proposed MT-MDD and ST-MDD models. Note that TTS parameters only apply to the MT-MDD model.

TABLE III
HYPERPARAMETERS OF THE MT-MDD MODEL

	Block	Component	Parameters
MDD	Acoustic encoder	PreNet	2 x FC layers 512 units; ReLU
		1 x bidirectional GRU	256 cells
	Text encoder	Embedding layer	64 embeddings each 1024 dim
		PreNet	2 x FC layers 1024 and 512 units; ReLU
		3 x Conv layers	5 x 1 kernels with 1 x 1 stride ReLU; batch norm
		1 x bidirectional LSTM	512 cells
	Attention	Bahdanau attention	512 dim attention context
	Decoder	1 x GRU	512 cells
		PostNet	3 x Conv layer 5 x 1 kernel with 1 x 1 stride ReLU; batch norm 1 bidirectional LSTM 512 cells 1 x linear layer 42 units; softmax
TTS	Encoder	Embedding	42 embeddings each 512 dim
		PreNet	2 x FC layers 512 units; ReLU
		5 x 1 Conv layers	5 x 1 kernels with 1 x 1 stride MaxPooling 2 x 1 kernel
		Highway	4 x FC layers 512 units; ReLU
		1 x bidirectional GRU	256 cells
		Projection layer	1 x FC layer
	Attention	Location sensitive attention	512 dim attention context
	Decoder	PreNet	2 x FC layers 256 units; ReLU
		2 x LSTM	1024 cells
		Linear (Mel)	1 x FC layer; 80 units
		Linear (stop token)	1 x FC layer; 1 unit
		PostNet	5 x 1 Conv layers MaxPooling 2 x 1 kernel 4 x FC layers 512 units; ReLU 1 x bidirectional GRU; 256 cells
	Speaker encoder	3 x LSTM Linear	256 cells 1 x FC layer 80 units; ReLU

VIII. Acknowledgments

This work was funded by NSF award 2016959. We would like to acknowledge Profs. John Levis and Evgeny Chuckarev for initial discussions of the proposed MT-MDD system.

References

[1] A. M. Harrison, W. Y. Lau, H. M. Meng, and L. Wang, "Improving mispronunciation detection and diagnosis of learners' speech with context-sensitive phonological rules based on language transfer," in *Proc Interspeech*, 2008.

[2] W.-K. Lo, S. Zhang, and H. Meng, "Automatic derivation of phonological rules for mispronunciation detection in a computer-assisted pronunciation training system," in *Proc Interspeech*, 2010.

[3] W.-K. Leung, X. Liu, and H. Meng, "Cnn-rnn-ctc based end-to-end mispronunciation detection and diagnosis," in *Proc Intl Conf AcousticsSpeech, Signal Processing (ICASSP)*. IEEE, 2019, pp. 8132–8136.

[4] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proc Intl Conf Machine Learning (ICML)*, 2006, pp. 369–376.

[5] Y. Feng, G. Fu, Q. Chen, and K. Chen, "Sed-mdd: Towards sentence dependent end-to-end mispronunciation detection and diagnosis," in *Proc Intl Conf Acoustics Speech Signal Processing (ICASSP)*. IEEE, 2020, pp. 3492–3496.

[6] W. Ye, S. Mao, F. Soong, W. Wu, Y. Xia, J. Tien, and Z. Wu, "An approach to mispronunciation detection and diagnosis with acoustic, phonetic and linguistic (apl) embeddings," in *Proc Intl Conf Acoustics Speech Signal Processing (ICASSP)*. IEEE, 2022, pp. 6827–6831.

[7] G. Zhao, S. Sonsaat, A. Silpachai, I. Lucic, E. Chukharev-Hudilainen, J. Levis, and R. Gutierrez-Osuna, "L2-arctic: A non-native english speech corpus." in *Proc Interspeech*, 2018, pp. 2783–2787.

[8] G. Pironkov, S. Dupont, and T. Dutoit, "Speaker-aware multi-task learning for automatic speech recognition," in *Proc Intl Conf Pattern Recognition (ICPR)*. IEEE, 2016, pp. 2900–2905.

[9] S. Karita, S. Watanabe, T. Iwata, M. Delcroix, A. Ogawa, and T. Nakatani, "Semi-supervised end-to-end speech recognition using text-to-speech and autoencoders," in *Proc Intl Conf Acoustics Speech Signal Processing (ICASSP)*. IEEE, 2019, pp. 6166–6170.

[10] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Adv Neural Information Processing Systems*, vol. 33, pp. 12 449–12 460, 2020.

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

[12] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc Intl Conf Computer Vision (ICCV)*, 2017, pp. 2223–2232.

[13] S. M. Witt and S. J. Young, "Phone-level pronunciation scoring and assessment for interactive language learning," *Speech Communication*, vol. 30, no. 2-3, pp. 95–108, 2000.

[14] F. Zhang, C. Huang, F. K. Soong, M. Chu, and R. Wang, "Automatic mispronunciation detection for mandarin," in *Proc Intl Conf Acoustics Speech Signal Processing (ICASSP)*. IEEE, 2008, pp. 5077–5080.

[15] S. Sudhakara, M. K. Ramanathi, C. Yarra, and P. K. Ghosh, "An improved goodness of pronunciation (gop) measure for pronunciation evaluation with dnn-hmm system considering hmm transition probabilities." in *Proc Interspeech*, vol. 2, 2019, pp. 954–958.

[16] W. Hu, Y. Qian, F. K. Soong, and Y. Wang, "Improved mispronunciation detection with deep neural network trained acoustic models and transfer learning based logistic regression classifiers," *Speech Communication*, vol. 67, pp. 154–166, 2015.

[17] H. Strik, K. Truong, F. De Wet, and C. Cucchiarini, "Comparing different approaches for automatic pronunciation error detection," *Speech Communication*, vol. 51, no. 10, pp. 845–852, 2009.

[18] A. Ito, Y.-L. Lim, M. Suzuki, and S. Makino, "Pronunciation error detection method based on error rule clustering using a decision tree," in *European Conf Speech Communication Tech*, 2005.

[19] M. Shahin and B. Ahmed, "Anomaly detection based pronunciation verification approach using speech attribute features," *Speech Communication*, vol. 111, pp. 29–43, 2019.

[20] H. Meng, Y. Y. Lo, L. Wang, and W. Y. Lau, "Deriving salient learners' mispronunciations from cross-language phonological comparisons," in *Proc Workshop Automatic Speech Recognition & Understanding (ASRU)*. IEEE, 2007, pp. 437–442.

[21] X. Qian, H. Meng, and F. Soong, "A two-pass framework of mispronunciation detection and diagnosis for computer-aided pronunciation training," *IEEE/ACM Trans Audio Speech Language Processing*, vol. 24, no. 6, pp. 1020–1028, 2016.

[22] K. Li, X. Qian, and H. Meng, "Mispronunciation detection and diagnosis in l2 english speech using multidistribution deep neural networks," *IEEE/ACM Trans Audio Speech Language Processing*, vol. 25, no. 1, pp. 193–207, 2016.

[23] S. Mao, Z. Wu, R. Li, X. Li, H. Meng, and L. Cai, "Applying multitask learning to acoustic-phonemic model for mispronunciation detection and diagnosis in l2 english speech," in *Proc Intl Conf Acoustics Speech Signal Processing (ICASSP)*. IEEE, 2018, pp. 6254–6258.

[24] K. Fu, J. Lin, D. Ke, Y. Xie, J. Zhang, and B. Lin, "A full text-dependent end to end mispronunciation detection and diagnosis with easy data augmentation techniques," *arXiv preprint arXiv:2104.08428*, 2021.

[25] X. Xu, Y. Kang, S. Cao, B. Lin, and L. Ma, "Explore wav2vec 2.0 for mispronunciation detection." in *Proc Interspeech*, 2021, pp. 4428–4432.

[26] L. Peng, K. Fu, B. Lin, D. Ke, and J. Zhang, "A study on fine-tuning wav2vec2. 0 model for the task of mispronunciation detection and diagnosis." in *Proc Interspeech*, 2021, pp. 4448–4452.

[27] M. Yang, K. Hirschi, S. D. Looney, O. Kang, and J. H. Hansen, "Improving mispronunciation detection with wav2vec2-based momentum pseudo-labeling for accentedness and intelligibility assessment," *arXiv preprint arXiv:2203.15937*, 2022.

[28] M. Wu, K. Li, W.-K. Leung, and H. Meng, "Transformer based end-to-end mispronunciation detection and diagnosis." in *Proc Interspeech*, 2021, pp. 3954–3958.

[29] A. J. Hunt and A. W. Black, "Unit selection in a concatenative speech synthesis system using a large speech database," in *Proc Intl Conf Acoustics Speech Signal Orocessing (ICASSP)*, vol. 1. IEEE, 1996, pp. 373–376.

[30] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, "Speech parameter generation algorithms for hmm-based speech synthesis," in *Proc Intl Conf Acoustics Speech Signal Orocessing (ICASSP)*, vol. 3. IEEE, 2000, pp. 1315–1318.

[31] K. Tokuda, Y. Nankaku, T. Toda, H. Zen, J. Yamagishi, and K. Oura, "Speech synthesis based on hidden markov models," *Proc. IEEE*, vol. 101, no. 5, pp. 1234–1252, 2013.

[32] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[33] A. Gibiansky, S. Arik, G. Diamos, J. Miller, K. Peng, W. Ping, J. Raiman, and Y. Zhou, "Deep voice 2: Multi-speaker neural text-to-speech," *Adv Neural Information Processing Systems*, vol. 30, 2017.

[34] W. Ping, K. Peng, A. Gibiansky, S. O. Arik, A. Kannan, S. Narang, J. Raiman, and J. Miller, "Deep voice 3: Scaling text-to-speech with convolutional sequence learning," *arXiv preprint arXiv:1710.07654*, 2017.

[35] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan *et al.*, "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions," in *Proc Intl Conf Acoustics Speech Signal Processing (ICASSP)*. IEEE, 2018, pp. 4779–4783.

[36] Y. Ren, Y. Ruan, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "Fastspeech: Fast, robust and controllable text to speech," *Adv Neural Information Processing Systems*, vol. 32, 2019.

[37] I. Elias, H. Zen, J. Shen, Y. Zhang, Y. Jia, R. J. Weiss, and Y. Wu, "Parallel tacotron: Non-autoregressive and controllable tts," in *Proc Intl Conf Acoustics Speech Signal Processing (ICASSP)*. IEEE, 2021, pp. 5709–5713.

[38] J. Stadermann, W. Koska, and G. Rigoll, "Multi-task learning strategies for a recurrent neural net in a hybrid tied-gs acoustic mode," in *Proc Interspeech*, 2005.

[39] M. L. Seltzer and J. Droppo, "Multi-task learning in deep neural networks for improved phoneme recognition," in *Proc Intl Conf Acoustics Speech Signal Processing (ICASSP)*. IEEE, 2013, pp. 6965–6969.

[40] Q. Hu, Z. Wu, K. Richmond, J. Yamagishi, Y. Stylianou, and R. Maia, "Fusion of multiple parameterisations for dnn-based sinusoidal speech synthesis with multi-task learning," in *Proc Interspeech*, 2015.

[41] M. S. Ribeiro, O. Watts, J. Yamagishi, and R. A. Clark, "Wavelet-based decomposition of f0 as a secondary task for dnn-based speech synthesis with multi-task learning," in *Proc Intl Conf Acoustics Speech Signal Processing (ICASSP)*. IEEE, 2016, pp. 5525–5529.

[42] W. Ding and L. He, "Mtgan: Speaker verification through multitasking triplet generative adversarial networks," *arXiv preprint arXiv:1803.09059*, 2018.

[43] S. Sigtia, E. Marchi, S. Kajarekar, D. Naik, and J. Bridle, "Multi-task learning for speaker verification and voice trigger detection," in *Proc Intl Conf Acoustics Speech Signal Processing (ICASSP)*. IEEE, 2020, pp. 6844–6848.

[44] S. Parthasarathy and C. Busso, "Jointly predicting arousal, valence and dominance with multi-task learning." in *Proc Interspeech*, vol. 2017, 2017, pp. 1103–1107.

[45] Y. Li, T. Zhao, T. Kawahara *et al.*, "Improved end-to-end speech emotion recognition using self attention mechanism and multitask learning." in *Proc Interspeech*, 2019, pp. 2803–2807.

[46] Y. Lu, F. Lu, S. Sehgal, S. Gupta, J. Du, C. H. Tham, P. Green, and V. Wan, "Multitask learning in connectionist speech recognition," in *Proc Australian Intl Conf Speech Science Technology*, 2004.

[47] V. Arora, A. Lahiri, and H. Reetz, "Phonological feature-based speech recognition system for pronunciation training in non-native language learning," *J Acoustical Society America*, vol. 143, no. 1, pp. 98–108, 2018.

[48] K. Krishna, S. Toshniwal, and K. Livescu, "Hierarchical multitask learning for ctc-based speech recognition," *arXiv preprint arXiv:1807.06234*, 2018.

[49] J. J. Godfrey, E. C. Holliman, and J. McDaniel, "Switchboard: Telephone speech corpus for research and development," in *Proc Intl Conf*

Acoustics Speech Signal Processing (ICASSP)), vol. 1. IEEE Computer Society, 1992, pp. 517–520.
[50] K.-F. Lee and H.-W. Hon, "Speaker-independent phone recognition using hidden markov models," *IEEE/ACM Trans Acoustics Speech Signal Processing*, vol. 37, no. 11, pp. 1641–1648, 1989.
[51] J.-T. Huang, J. Li, D. Yu, L. Deng, and Y. Gong, "Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers," in *Proc Intl Conf Acoustics Speech Signal Processing (ICASSP)*. IEEE, 2013, pp. 7304–7308.
[52] W. Hou, Y. Dong, B. Zhuang, L. Yang, J. Shi, and T. Shinozaki, "Large-scale end-to-end multilingual speech recognition and language identification with multi-task learning," *Babel*, vol. 37, no. 4k, p. 10k, 2020.
[53] Z. Wu, C. Valentini-Botinhao, O. Watts, and S. King, "Deep neural networks employing multi-task learning and stacked bottleneck features for speech synthesis," in *Proc Intl Conf Acoustics Speech Signal Processing (ICASSP)*. IEEE, 2015, pp. 4460–4464.
[54] L. Tóth, G. Gosztolya, T. Grósz, A. Markó, and T. G. Csapó, "Multi-task learning of speech recognition and speech synthesis parameters for ultrasound-based silent speech interfaces." in *Proc Interspeech*, 2018, pp. 3172–3176.
[55] X. Cai, J. Yuan, R. Zheng, L. Huang, and K. Church, "Speech emotion recognition with multi-task learning." in *Proc Interspeech*, vol. 2021, 2021, pp. 4508–4512.
[56] N. Chen, Y. Qian, and K. Yu, "Multi-task learning for text-dependent speaker verification," in *Proc Interspeech*, 2015.
[57] L. Wan, Q. Wang, A. Papir, and I. L. Moreno, "Generalized end-to-end loss for speaker verification," in *Proc Intl Conf Acoustics Speech Signal Orocessing (ICASSP)*. IEEE, 2018, pp. 4879–4883.
[58] C. Jemine, "Real-time-voice-cloning," *University of Liége, Liége, Belgium*, p. 3, 2019.
[59] W. Quamer, A. Das, J. Levis, E. Chukharev-Hudilainen, and R. Gutierrez-Osuna, "Zero-shot foreign accent conversion without a native reference," in *Proc Interspeech*, 2022, pp. 4920–4924.
[60] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc Intl Conf Machine Learning (ICML)*. pmlr, 2015, pp. 448–456.
[61] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.
[62] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," *Adv Neural Information Processing Systems*, vol. 28, 2015.
[63] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, and D. S. Pallett, "Darpa timit acoustic-phonetic continous speech corpus cd-rom. nist speech disc 1-1.1," *NASA STI/Recon technical report*, vol. 93, p. 27403, 1993.
[64] C. Liberatore, S. Aryal, Z. Wang, S. Polsley, and R. Gutierrez-Osuna, "Sabr: sparse, anchor-based representation of the speech signal," in *Proc Interspeech*, 2015.
[65] J. Kearns, "Librivox: Free public domain audiobooks," *Reference Reviews*, vol. 28, no. 1, pp. 7–8, 2014.
[66] A. Nagrani, J. S. Chung, W. Xie, and A. Zisserman, "Voxceleb: Large-scale speaker verification in the wild," *Computer Speech & Language*, vol. 60, p. 101027, 2020.
[67] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
[68] T. Okamoto, T. Toda, Y. Shiga, and H. Kawai, "Real-time neural text-to-speech with sequence-to-sequence acoustic model and waveglow or single gaussian wavernn vocoders." in *Proc Interspeech*, 2019, pp. 1308–1312.
[69] N. Kalchbrenner, E. Elsen, K. Simonyan, S. Noury, N. Casagrande, E. Lockhart, F. Stimberg, A. Oord, S. Dieleman, and K. Kavukcuoglu, "Efficient neural audio synthesis," in *Proc Intl Conf Machine Learning (ICML), pages=2410–2419, year=2018, organization=PMLR*.
[70] V. Likic, "The needleman-wunsch algorithm for sequence alignment," *Lecture 7th Melbourne Bioinformatics Course (Bi021)*, pp. 1–46, 2008.
[71] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *J Machine Learning Research*, vol. 9, no. 11, 2008.
[72] G. Zhao, S. Sonsaat, J. Levis, E. Chukharev-Hudilainen, and R. Gutierrez-Osuna, "Accent conversion using phonetic posteriorgrams," in *Proc Intl Conf Acoustics Speech Signal Processing (ICASSP)*. IEEE, 2018, pp. 5314–5318.
[73] G. Zhao, S. Ding, and R. Gutierrez-Osuna, "Converting foreign accent speech without a reference," *IEEE/ACM Trans Audio Speech Language Processing*, vol. 29, pp. 2367–2381, 2021.
[74] S. Aryal and R. Gutierrez-Osuna, "Articulatory-based conversion of foreign accents with deep neural networks," in *Proc Interspeech*, 2015.
[75] L. F. Bachman, *Fundamental considerations in language testing*. Oxford Univ Pr, 1990.
[76] M. Eskenazi, "An overview of spoken language technology for education," *Speech Communication*, vol. 51, no. 10, pp. 832–844, 2009.
[77] T. Raitio, J. Li, and S. Seshadri, "Hierarchical prosody modeling and control in non-autoregressive parallel neural tts," in *Proc Intl Conf Acoustics Speech Signal Processing (ICASSP)*. IEEE, 2022, pp. 7587–7591.
[78] D. Korzekwa, J. Lorenzo-Trueba, T. Drugman, and B. Kostek, "Computer-assisted pronunciation training—speech synthesis is almost all you need," *Speech Communication*, vol. 142, pp. 22–33, 2022.

Anurag Das received his BTech degree in Electronics and Communication Engineering from National Institute of Technology, Silchar in 2016. He is pursuing his PhD in Computer Science at Texas A&M University College Station, TX, USA. His research interests include speech recognition and synthesis, mispronunciation detection in second-language speech, and developing machine learning solutions for nutrition and diabetes management.

Ricardo Gutierrez-Osuna (Senior Member, IEEE) received a B.S. degree in electrical engineering from the Polytechnic University of Madrid in 1992 and M.S. and Ph.D. degrees in computer engineering from North Carolina State University, in 1995 and 1998, respectively. He is a Professor in the Department of Computer Science and Engineering at Texas A&M University. His research interests include chemometrics, wearable sensors, and speech processing.