Decoupling Segmental and Prosodic Cues of Non-native Speech through Vector Quantization

Conference Paper · August 2023

DOI: 10.21437/Interspeech.2023-2202

3 authors:

Waris Quamer
Texas A&M University
10 PUBLICATIONS **88** CITATIONS

SEE PROFILE

Anurag Das
Heritage Institute of Technology
9 PUBLICATIONS **27** CITATIONS

SEE PROFILE

Ricardo Gutierrez-Osuna
Texas A&M University
199 PUBLICATIONS **5,670** CITATIONS

SEE PROFILE

INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

Decoupling segmental and prosodic cues of non-native speech through vector quantization

Waris Quamer[1], *Anurag Das*[1], *Ricardo Gutierrez-Osuna*[1]

[1]Department of Computer Science and Engineering, Texas A&M University, United States
{quamer.waris, adas, rgutier}@tamu.edu

Abstract

Accent conversion (AC) seeks to transform utterances from a non-native speaker to appear native-like. Compared to voice conversion, which generally treats accent and voice quality as one, AC provides a finer-grained decomposition of speech. This paper presents an AC system that further decomposes an accent into its segmental and prosodic characteristics, and provides independent control of both channels. The system uses conventional modules (acoustic model, speaker/prosody encoders, seq2seq model) to generate accent conversions that combine (1) the segmental characteristics from a source utterance, (2) the voice characteristics from a target utterance, and (3) the prosody of a reference utterance. However, naive application of this idea prevents the system from learning and transferring prosody. We show that vector quantization and removal of repeated codewords allows the system to transfer prosody and improve voice similarity, as verified by objective and perceptual measures.

Index Terms: accent conversion, voice conversion, prosody modeling, vector quantization, non-native speech

1. Introduction

Older learners of a second language (L2) often speak with a so-called "foreign accent." While many other aspects of communicating in an L2 can be acquired well into adulthood (e.g., vocabulary, grammar, writing), achieving native-like pronunciation is difficult past a critical period because of the neuro-musculatory basis of speech production [1]. While a native accent is not required to be intelligible, improving pronunciation can reduce listening effort [2] as well as negative social evaluations [3].

To improve pronunciation, several studies have suggested that L2 learners would benefit from imitating a model voice that is close to their own voice [4], if not their own voice transformed to sound native-like [5]. In fact, several techniques have been proposed for this purpose, borrowing models from the voice conversion and speech synthesis literature [6]. These "accent conversion" techniques provide a finer-grained separation of speaker characteristics than voice conversion [7], since they treat accent and voice quality as independent factors to be disentangled. However, accent conversion techniques do not attempt to disentangle the two main sources of non-native accent: segmental and prosodic characteristics. Being able to manipulate an L2 speaker's segmental and/or prosodic characteristics independently is critical to quantify how these two channels contribute to speech comprehensibility and social attitudes. This information would further be used to improve self-imitation tools in computer assisted pronunciation training.

As a first step to address this issue, we present a model that allows an utterance U1 from any source speaker to be resynthesized to match the voice quality in any target utterance U2 (as in voice cloning [8]) and the prosody from a reference utterance U3 (as in expressive text-to-speech synthesis [9]). Our model passes U1 through an acoustic model to generate a speaker-independent phonetic posteriorgram (PPG), and then to a sequence-to-sequence (seq2seq) model that combines the PPG with a speaker embedding from U2 and a prosody embedding from U3. However, naïve application of this strategy leads the seq2seq model to preserve the prosodic content in U1, which is readily available in the PPG (e.g., duration), instead of those in U2, which are heavily encoded.

Key contributions. To solve this problem, we propose a technique that reduces prosodic information in the PPG, bringing it close to the information available in a phonetic transcription. Namely, we apply vector quantization to the PPGs, and then remove consecutive duplicates in the resulting VQ-PPG. This simple trick forces the seq2seq model to use the prosodic embedding to reconstruct the speech signal and, additionally, makes the converted speech significantly closer to that of the target speaker. We evaluate the approach using objective and subjective measures of acoustic quality, speaker transfer and prosody transfer, and compare it against a baseline system that does not use vector quantization. Our results show that the proposed system achieves significantly better transfer of prosody characteristics and, as a side benefit, improved transfer of voice characteristics.

2. Related work

2.1. Prosody modeling for speech synthesis

Research on prosody modeling for speech synthesis can be broadly divided into two categories: text-to-speech (TTS) and voice conversion (VC). In TTS, Skerry-Ryan et al. [9] developed a Tacotron-based speech synthesizer with an encoder module that separates prosody information from the original speech. They showed that conditioning the synthesizer on this learnt embedding can be used to synthesize audio that matches the prosody of the reference signal. Wang et al. [10] trained "global style tokens", a bank of embeddings with a Tacotron-based seq2seq model without any explicit labels. The model is trained in a self-supervised manner and the learnt embeddings can be used to alter the speed of the speech signal, control, and transfer the speaking style, independently from the text content.

To control prosody in VC, AutoVC [11] uses an auto-encoder based network with an information bottleneck to disentangle speaker information form linguistic content while reconstructing the original speech. SpeechFlow [12] extended AutoVC by using three encoder channels with different information bottleneck designs and adding randomly sampled noise to disentangle content, pitch, rhythm, and speaker identity. However, bottlenecks need to be carefully designed to effectively separate the three prosodic features.

10.21437/Interspeech.2023-2202

2.2. Vector quantization

Vector quantization (VQ) reduces the speech signal into a number of discrete clusters. Baevski [13] used VQ to learn discrete representations of audio segments and predict future speech segments as in wav2vec [14]. They showed that pre-training a BERT model with the learnt representations improved phoneme classification on the TIMIT dataset and speech recognition performance on the WSJ dataset. In the context of VC, Wu et al. [15] used VQ to develop VQVC and disentangled speaker and content representations and trained the model to reconstruct the speech signal. VQVC+ [16] improved VC performance using a U-net architecture and an auto-encoder based system to generate audio of high quality. However, the content and speaker representations in these systems may still be entangled. To avoid this, Wang et al. [17] proposed VQMI, a model that combined VQ with mutual information to decorrelate the individual representations as much as possible. Their model only retained linguistic and intonation variations from the source speaker while capturing target speaker characteristics, and achieved state-of-the-art performance for one-shot VC.

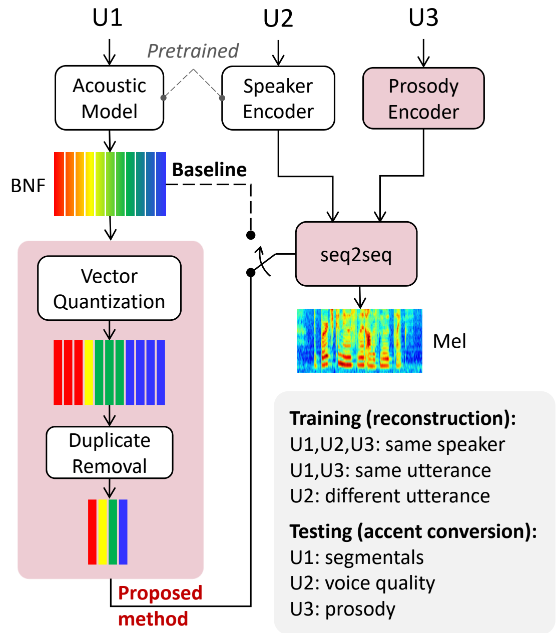
3. Methods

The proposed model is illustrated in Figure 1. An acoustic model (AM) converts an input utterance (U1) into a bottleneck feature (BNF) matrix that captures the phonetic content of the utterance. The BNF matrix is then passed to (1) a vector quantization (VQ) module that discretizes each column (i.e., frame) into one of N codewords (i.e., cluster centers), and then (2) a duplicate removal (DR) stage that eliminates consecutive duplicates of each codeword. The resulting short sequence codewords can be viewed as a sequence of phonemic codes (for N=39) or sub-phonetic codes (for larger N). Thus, it is akin to a phonemic transcription, except phonemes are not represented by symbols but by their corresponding BNFs.

A seq2seq model consumes (i) a short sequence of codewords from utterance U1, (ii) a speaker embedding representing the voice quality in utterance U2 from a target speaker, and (iii) a prosody embedding from a reference utterance U3. From these three information bottlenecks, the seq2seq attempts to reconstruct the original Mel spectrogram. The prosody encoder and seq2seq model are trained simultaneously in an unsupervised fashion (i.e., as an auto-encoder) while the speaker encoder and acoustic model are pre-trained in advance. During training, the acoustic model and prosody encoder are fed the same utterance from the same speaker, whereas the speaker embedding is fed a different utterance U2 from the same speaker. This trick ensures the prosody encoder learns a different mapping than the speaker encoder, and the seq2seq model does not attempt to infer prosody from the speaker embedding.

3.1. Acoustic Model

The acoustic model generates a phonetic posteriorgram (PPGs) containing the posterior probability that each speech frame belongs to a predefined set of phonetic units (phonemes or triphones/senones). The model is a TDNN-F network with 5 hidden layers and ReLU activation, with 256 neurons in the last hidden layer [18]. Following [19], we train the TDNN-F on the Librispeech corpus [20]. Following [21], we use the 256-dim output of the last hidden layer as bottleneck features (BNF). Compared to the PPG generated in the final softmax layer, BNFs have much lower dimensionality (256 vs 6024 for senone-PPGs), which makes training the seq2seq easier.



Training (reconstruction):
U1,U2,U3: same speaker
U1,U3: same utterance
U2: different utterance

Testing (accent conversion):
U1: segmentals
U2: voice quality
U3: prosody

Figure 1: *Block diagram of the proposed system. The prosody encoder and seq2seq model are trained jointly as an auto-encoder.* ***For accent conversion, segmentals come from U1 and prosody from U3, thus providing independent control of both channels.***

3.2. Vector Quantization and Duplicate Removal

Though the BNF matrix captures primarily segmental information in U1, it also preserves significant prosodic characteristics (e.g., phone duration, speaking rate). As such, if the BNF matrix is used as an input, the seq2seq model must learn to ignore its prosody content (which is that of U1) and instead focus on the prosody embedding from utterance U3. However, prosody in U1 is trivially available (i.e., the number of columns in the BNF matrix equals the duration), whereas prosody in U3 is encoded into a compact vector. As such, the seq2seq generally converges to a local minimum that ignores the prosody encoding and instead preserves the prosody in the BNF matrix.

To avoid this local minimum, we propose to remove prosodic content in the BNF matrix using vector quantization (VQ). Namely, we pre-train a k-means clustering model to learn a set of codewords (i.e., cluster centers) from the L2-ARCTIC corpus [22] (20 speakers, 1000 utterances each). Once the codebook has been learned, we replace each column in the BNF matrix with its corresponding codeword, and finally eliminate any duplicate codewords that are adjacent in the sequence, as depicted in Figure 1. In this fashion, timing information is removed from the BNF matrix, which is reduced to a short sequence of codewords that only preserves key segmental information in U1.

Key insight. When we use this short codebook sequence to jointly train the prosody encoder and the seq2seq model, the two modules are forced to learn complementary tasks. First, the prosody encoder is forced to learn to generate an embedding that summarizes the prosody in U3. Second, the seq2seq model is forced to learn to combine the prosody embedding with the short codebook sequence to reconstruct the original Mel spectrogram.

Though not our main focus, a second major advantage of vector quantizing BNFs is that it can lead to significant improvements in voice conversion performance, as shown in prior studies [15, 16]. It is important to note that this secondary benefit is due to the VQ step alone, not the subsequent DR step.

3.3. Sequence-to-Sequence Model

Our seq2seq model is derived from the Voice Transformer Network [23] which is a combination of Transformer [24] and Tacotron2 [25]. As suggested in [26], the Transformer architecture is adapted to the VC task by adding pre-nets to the decoder. An extra linear layer was added to predict the stop token, along with a weighted binary cross-entropy loss to train the model to learn when to stop decoding. Similar to recent TTS models [25, 27], a five-layer CNN postnet was used to predict a residual to refine the final prediction.

The seq2seq model takes in a BNF matrix and outputs a converted log-mel spectrogram. The high time resolution of both the input and output acoustic features in VC makes attention learning difficult and increases the training memory footprint. While training our baseline model, we use a reduction factor r_e and r_d on both encoder and decoder side respectively so that it can stack multiple frames to reduce the time axis. This not only improves attention alignment but also reduces the training memory footprint by half and the number of required gradient accumulation steps [26]. When using duplicate removal, the time resolution of the input vector quantized BNFs are comparable to text inputs in TTS systems, and much lower than those of original acoustic features. So, in the latter case, we employ the reduction factor r_d only on the decoder side.

4. Experimental setup

We conducted our experiments using the ARCTIC [28] and L2-ARCTIC [22] corpora. The combined dataset consists of 28 speakers (1,132 utterance each), out of which four speakers (NJS, YKWK, TXHC and ZHAA) were excluded from the training set so they could be used as unseen speakers during testing. We use speaker BDL from ARCTIC as the reference L1 speaker for all experiments. For each utterance, we extracted 80-dim Mel-spectrograms with 25ms window and 10ms shift. The seq2seq model consisted of 4 encoding layers and 4 decoding layers, and both had reduction factors r_e and r_d of 2. We set the batch size to 16, and used the Lamb optimizer with a learning rate of 10^{-3} annealed down to 10^{-5} by exponential scheduling. To convert Mel-spectrograms to waveforms, we used a pre-trained HiFiGAN vocoder [29]. All our models were trained using two NVIDIA Tesla V100 GPUs.

5. Results

We evaluated the proposed model on a series of objective and subjective measures of synthesis quality, speaker transfer, and prosody transfer using the four speakers in L2-ARCTIC [22] that were held out when training the prosody embedding and seq2seq model.

5.1. Synthesis quality

We evaluated synthesis quality using objective and subjective measures. As an objective measure, we examined how the size of the codebook impacted Mel Cepstral Distortion (MCD). For this purpose, we used the system as an auto-encoder: to reconstruct at the output the same utterance fed to the acoustic model, prosody encoder and speaker encoding (i.e., U1=U2=U3). Results are shown in Figure 2 for different codebook sizes; no-VQ is equivalent to having an infinite number of codewords ($vq\infty$). As shown, MCD decreases significantly as the codebook size increases up to 128 codewords, after which the MCD stabilizes. One-way ANOVA shows that the effect of codebook size is statistically significant $F(7, 16) = 14.23, p \ll 0.001$. Further, paired t-test shows a significant difference between $vq128$ and $vq64$ ($p = 0.007$, one-tailed), and between $vq128$ and $vq\infty$ ($p = 0.005$, one-tailed). Thus, while the lowest MCD is achieved when VQ is not used ($vq\infty$), the lowest MCD among all the VQ models is for 128 codewords. As such, all subsequent models in this study are based on $vq128$.

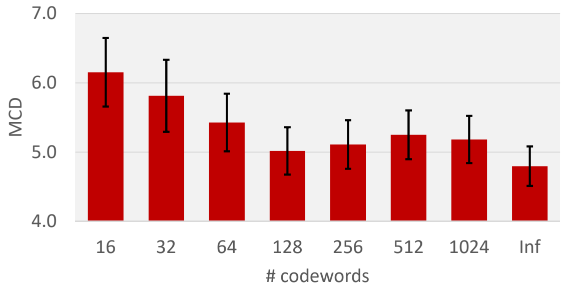


Figure 2: *Mel Cepstral Distortion (MCD) vs. codebook size.*

Table 1: *MOS for baseline ($vq\infty$) and proposed ($vq128$)*

	Target	Baseline	Proposed	p value
MOS	4.32 ± 0.82	3.89 ± 0.83	3.69 ± 0.79	$\ll$ 0.001

To verify these results perceptually, we conducted a listening test on Amazon Mechanical Turk (AMT), where listeners (N=20) were asked to rate the acoustic quality of utterances using a standard 5-point scale mean opinion score (MOS) as follows [rating, speech quality, level of distortion]: [5, excellent, imperceptible] — [4, good, just perceptible but not annoying] — [3, fair, perceptible but slightly annoying] — [2, poor, annoying but not objectionable] — [1, bad, very annoying and objectionable]. Each listener rated 20 utterances from the $vq128$ model (proposed) and the $vq\infty$ model (which served as a baseline), as well as original L2 utterances. Results are shown in Table 1. As expected, L2 utterances received the highest MOS ratings (4.32). Speech quality dropped by 0.43 MOS points (p$\ll$ 0.001) for the baseline system, and an additional 0.20 points (p$\ll$ 0.001) for the proposed system. While this result was also expected (and consistent with the objective results in Figure 2), it is noteworthy that discretizing the speech spectrum down to 128 codewords achieves nearly the same synthesis quality as using the full range of spectral variability in the speech corpus.

5.2. Speaker identity transfer

As we had done for synthesis quality, we used objective and subjective measures to evaluate speaker transfer in models $vq\infty$ (baseline) and $vq128$. As an objective measure, we visualized the embeddings produced by the speaker encoder for the source speaker (BDL), three target speakers (NJS, TXHC, ZHAA), and voice conversions from both systems, 10 utterances per voice. Results are shown in Figure 3. In this t-SNE plot, speaker transfer is inversely proportional to the distance from each voice conversion ($vq\infty$, $vq128$) to the corresponding target speaker. As shown, voice conversions from $vq128$ are significantly closer to their target than those from $vq\infty$, indicating that vector quantizing improves transfer from source to target speaker when compared to not using vector quantization.

To corroborate these results, we conducted an ABX listening test on AMT, where participants were presented with two audio samples, one from $vq\infty$ and one from $vq128$ (in a coun-

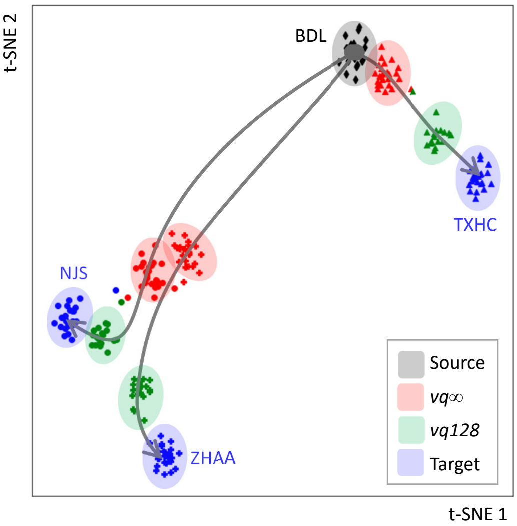


Figure 3: *t-SNE of speaker embeddings for source (black), target (blue), $vq\infty$ (red) and $vq128$ (green). The arrows represent a path connecting source and target utterances, passing through conversions from the two systems. Conversions from $vq128$ are much closer to the target than those from $vq\infty$, indicating that the $vq128$ system provides better transfer of speaker identity.*

Table 2: *Perceptual ratings of* ***speaker transfer*** *in an ABX test*

Rating	**Baseline** ($vq\infty$)	**Proposed** ($vq128$)
Closest to the L2 speaker	29.75%	70.25%
Average rater confidence	1.22	5.12

terbalanced fashion), followed by the original L2 utterance. Then, participants had to decide which audio sample ($vq\infty$ or $vq128$) was more similar to the L2 utterance, and rate the confidence in their decision using a 7-point scale (7: extremely confident; 5: quite a bit confident; 3: somewhat confident; 1: not confident at all). Following [30], the decision and confidence level were then collapsed to form a 14-point VSS (Voice Similarity Score) scale: -7 (definitely $vq\infty$) to +7 (definitely $vq128$). Each listener rated 10 ABX triplets per L2 speaker and system. As shown in Table 2, $vq128$ outputs were chosen as the closest to the L2 speaker 70.25% of the times, and with a high confidence level (5.12: quite a bit confident it is $vq128$), whereas the baseline ($vq\infty$) was selected only 29.75% of the times, and with a low confidence level (1:22: not confident at all it is $vq\infty$.) This result further corroborates the qualitative results in the t-SNE plot in Figure 3.

5.3. Prosody transfer

In the final set of tests, we examined how well the $vq\infty$ (baseline) and $vq128$ (proposed) models were able to transfer the prosodic characteristics of utterance U3. For this purpose, utterance U1 was from an L1 speaker, whereas utterances U2=U3 were from an L2 speaker. As such, the system was expected to generate an utterance with L1 segmentals and L2 prosody.

In a first experiment, we measured differences in duration, average F0 and F0 range between conversions from both systems and utterances U1/L1 (i.e., whose prosody should be ignored) and U2/L2 (i.e., whose prosody should be transferred). If prosody transfer was successful, we would hypothesize that the duration, F0 average and F0 range for the voice conversions would be closer to those of the L2 utterance than to those in the L1 utterance. Results in Table 3 confirm this hypothesis for the $vq128$ system, but the reverse hypothesis for the $vq\infty$ (baseline) system. Namely, the three measures of prosody for $vq\infty$ syntheses are closer to the L1 utterance (a negative result), whereas for the $vq128$ system the three measures are closer to the L2 utterance (a positive result). These results indicate that only the $vq128$ system is able to transfer the prosody characteristics present in the prosody embedding.

Table 3: *Differences in prosodic characteristics between original utterances (L1, L2) and accent conversions ($vq\infty$, $vq128$)*

	∆ duration		**∆ F0 avg**		**∆ F0 range**	
	(ms)		(Hz)		(Hz)	
	L1	L2	L1	L2	L1	L2
$vq\infty$	**16.89**	413.37	**36.83**	40.43	**48.53**	35.93
$vq128$	395.42	**5.89**	82.36	**7.96**	20.07	**11.42**

Table 4: *Perceptual ratings of* ***prosody transfer*** *in an ABX test*

Rating	**Baseline** ($vq\infty$)	**Proposed** ($vq128$)
Closest to the L2 speaker	31.12%	68.88%
Average rater confidence	1.31	3.2

To corroborate these findings, we conducted a second ABX test on AMT, where participants listened to audio samples from both systems ($vq\infty$ or $vq128$) in a counterbalanced fashion, followed by the original L2 utterance. As before, participants had to decide which audio sample ($vq\infty$ or $vq128$) was more similar to the L2 utterance, and then rate their confidence. Results are shown in Table 4. Listeners rated utterances from the proposed system ($vq128$) as the closest to the original L2 utterance 69% of the times with somewhat confidence (3.2), whereas the $vq\infty$ was selected the remaining 31.12% of the times with no confidence at all (1.2). This result is remarkable considering that listeners had to ignore differences in segmental content between the two accent conversions (L1 segmentals) and the L2 utterances (L2 segmentals), and instead focus on prosody.

6. Discussion

Conventional methods for accent conversion allow an L2 utterance to be resynthesized with both the segmental and the prosodic characteristics of native speech, but not one or the other. To address this limitation, we have proposed a model [1] that provides independent control of both channels. The trick is to discretize speech (subsampling in time, vector quantizing phonetic content) down to a handful of codewords, so that a seq2seq synthesizer learns to reconstruct speech using the prosody of a reference utterance. Through a series of objective and subjective experiments, we have shown that the discretization step in the time domain (duplicate removal) is **key** to achieve prosody transfer. Additionally, we show that vector quantizing the speech corpus leads to significantly better transfer of speaker identity.

The ability to control segmental and prosody characteristics independently enables future studies to quantify their relative effect in comprehensibility and social evaluations of non-native speech. Findings from these future studies could then be used to develop targeted interventions in pronunciation training.

7. Acknowledgements

This work was funded by NSF award 1619212 and 1623750.

[1] Audio samples from the two sytems can be found at: https://anonymousis23.github.io/demos/prosody-accent-conversion/

8. References

[1] J. P. Lantolf, "A time to speak: A psycholinguistic inquiry into the critical period for human speech," *Studies in Second Language Acquisition*, vol. 12, no. 1, pp. 84–85, 1990.

[2] K. J. Van Engen and J. E. Peelle, "Listening effort and accented speech," *Frontiers in human neuroscience*, vol. 8, p. 577, 2014.

[3] M. J. Munro, "A primer on accent discrimination in the Canadian context," *TESL Canada Journal*, vol. 20, no. 2, pp. 38–51, 2003.

[4] K. Probst, Y. Ke, and M. Eskenazi, "Enhancing foreign language tutors–in search of the golden speaker," *Speech Communication*, vol. 37, no. 3-4, pp. 161–173, 2002.

[5] K. Hirose, F. Gendrin, and N. Minematsu, "A pronunciation training system for Japanese lexical accents with corrective feedback in learner's voice," in *Proc. 8th European Conf Speech Communication Technology*, 2003, pp. 3149–3152.

[6] Z. Wang, W. Ge, X. Wang, S. Yang, W. Gan, H. Chen, H. Li, L. Xie, and X. Li, "Accent and speaker disentanglement in many-to-many voice conversion," in *Proc. ISCSLP*, 2021, pp. 1–5.

[7] B. Sisman, J. Yamagishi, S. King, and H. Li, "An overview of voice conversion and its challenges: From statistical modeling to deep learning," *IEEE/ACM Trans Audio Speech Language Process*, vol. 29, pp. 132–157, 2020.

[8] S. Arik, J. Chen, K. Peng, W. Ping, and Y. Zhou, "Neural voice cloning with a few samples," in *Proc. 32nd NIPS*, 2018, pp. 10 040–10 050.

[9] R. Skerry-Ryan, E. Battenberg, Y. Xiao, Y. Wang, D. Stanton, J. Shor, R. Weiss, R. Clark, and R. A. Saurous, "Towards end-to-end prosody transfer for expressive speech synthesis with tacotron," in *Proc. ICML*, 2018, pp. 4693–4702.

[10] Y. Wang, D. Stanton, Y. Zhang, R.-S. Ryan, E. Battenberg, J. Shor, Y. Xiao, Y. Jia, F. Ren, and R. A. Saurous, "Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis," in *Proc. ICML*, 2018, pp. 5180–5189.

[11] K. Qian, Y. Zhang, S. Chang, X. Yang, and M. Hasegawa-Johnson, "AUTOVC: Zero-shot voice style transfer with only autoencoder loss," in *Proc. ICML*, 2019, pp. 5210–5219.

[12] K. Qian, Y. Zhang, S. Chang, M. Hasegawa-Johnson, and D. Cox, "Unsupervised speech decomposition via triple information bottleneck," in *Proc. ICML*, 2020, pp. 7836–7846.

[13] A. Baevski, S. Schneider, and M. Auli, "VQ-wav2vec: Self-Supervised Learning of Discrete Speech Representations," in *Proc. 8th ICLR*, 2020.

[14] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," in *Proc. 34th NIPS*, vol. 33, 2020, pp. 12 449–12 460.

[15] D.-Y. Wu and H.-y. Lee, "One-shot voice conversion by vector quantization," in *Proc. ICASSP*, 2020, pp. 7734–7738.

[16] D.-Y. Wu, Y.-H. Chen, and H.-y. Lee, "VQVC+: One-Shot Voice Conversion by Vector Quantization and U-Net Architecture," in *Proc. Interspeech*, 2020, pp. 4691–4695.

[17] D. Wang, L. Deng, Y. T. Yeung, X. Chen, X. Liu, and H. Meng, "VQMIVC: Vector Quantization and Mutual Information-Based Unsupervised Speech Representation Disentanglement for One-Shot Voice Conversion," in *Proc. Interspeech*, 2021, pp. 1344–1348.

[18] X. Zhang, J. Trmal, D. Povey, and S. Khudanpur, "Improving deep neural network acoustic models using generalized maxout networks," in *Proc. ICASSP*, 2014, pp. 215–219.

[19] 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.

[20] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an ASR corpus based on public domain audio books," in *Proc. ICASSP*, 2015, pp. 5206–5210.

[21] G. Zhao, S. Ding, and R. Gutierrez-Osuna, "Converting foreign accent speech without a reference," *IEEE/ACM Trans Audio, Speech and Language Process*, vol. 29, pp. 2367–2381, 2021.

[22] 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.

[23] N. Li, S. Liu, Y. Liu, S. Zhao, and M. Liu, "Neural speech synthesis with transformer network," in *Proc AAAI*, vol. 33, no. 01, 2019, pp. 6706–6713.

[24] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. 31st NIPS*, 2017, p. 6000–6010.

[25] 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. ICASSP*, 2018, pp. 4779–4783.

[26] W.-C. Huang, T. Hayashi, Y.-C. Wu, H. Kameoka, and T. Toda, "Voice Transformer Network: Sequence-to-Sequence Voice Conversion Using Transformer with Text-to-Speech Pretraining," in *Proc. Interspeech*, 2020, pp. 4676–4680.

[27] S. Li, B. Ouyang, L. Li, and Q. Hong, "Light-TTS: Lightweight multi-speaker multi-lingual text-to-speech," in *Proc. ICASSP*, 2021, pp. 8383–8387.

[28] J. Kominek and A. W. Black, "The CMU Arctic speech databases," in *Proc. 5th ISCA workshop on speech synthesis*, 2004, pp. 223–224.

[29] J. Kong, J. Kim, and J. Bae, "Hifi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis," in *Proc. NIPS*, vol. 33, 2020, pp. 17 022–17 033.

[30] D. Felps, H. Bortfeld, and R. Gutierrez-Osuna, "Foreign accent conversion in computer assisted pronunciation training," *Speech Communication*, vol. 51, no. 10, pp. 920–932, 2009.