Streaming Non-Autoregressive Model for Accent Conversion and Pronunciation Improvement

Tuan-Nam Nguyen[1], *Ngoc-Quan Pham*[1,2], *Şeymanur Akti*[1], *Alexander Waibel*[1,2]

[1]Karlsruhe Institute of Technology, Germany
[2]Carnegie Mellon University, USA

tuan.nguyen@kikt.edu

Abstract

We propose a first streaming accent conversion (AC) model that transforms non-native speech into a native-like accent while preserving speaker identity, prosody and improving pronunciation. Our approach enables stream processing by modifying a previous AC architecture with an Emformer encoder and an optimized inference mechanism. Additionally, we integrate a native text-to-speech (TTS) model to generate ideal ground-truth data for efficient training. Our streaming AC model achieves comparable performance to the top AC models while maintaining stable latency, making it the first AC system capable of streaming.

Index Terms: accent conversion, speech recognition, human-computer interaction

1. Introduction

Second-language L2 English learners often have accents and mispronunciations that impact communication. AC modifies speech to enhance intelligibility while preserving content, emotion, and speaker identity. Streaming AC enables streaming applications like video conferencing [1], where seamless interaction are crucial. Previous AC mprocess full utterances, leveraging long context for accurate accent conversion and pronunciation correction.

Streaming AC requires on-the-fly speech conversion, posing challenges for previous AC models, which process entire utterances at once. Unlike non-streaming models, streaming AC must operate incrementally on speech frames or chunks. In this work, we modify a state-of-the-art non-streaming AC model to support streaming by adjusting its architecture and training process while leveraging knowledge distillation. To our knowledge, this is the first streaming AC model capable of both accent conversion and pronunciation improvement.

Previous AC methods typically follow two approaches. The first involves mapping non-native accents to native accents, relying heavily on parallel data for training [2, 3, 4, 5, 6]. However, such data—utterances from the same speakers in different accents—is rare and difficult to obtain. To address this, it is possible to use TTS [3, 7] or VC [5] to generate ground-truth audio synthetically, but these often mismatch in duration and prosody. Seq2seq encoder-decoder [8, 9, 10]models with attention help align inputs and outputs but they still struggle with length mismatches, making them unsuitable for tasks requiring precise synchronization, like video dubbing and conferencing. Their slow autoregressive inference and unstable attention mechanism further limit their applicability to real-time streaming.

The second approach, disentangle-resynthesis, leverages non-parallel data by decomposing speech into components like speaker identity, content, prosody, and accent, then allowing controlled modification and synthesis [11, 12]. Content features are often represented by bottleneck features (BNFs) extracted from self-supervised models [13, 14, 15], ASR bottlenecks [11], or ASR logits [16]. However, BNFs inherently capture accent features and pronunciation errors. While adversarial training [12] and Pseudo-Siamese networks [11] attempt to remove accent influences, they struggle to improve pronunciation due to the lack of ground-truth references.

Recent AC advancements combine both disentangling and mapping approaches into a unified framework[17]. They hypothesize that a TTS system trained solely on native speech will produce accent-independent linguistic representations. Additionally, this native TTS system is able to generate ideal ground-truth data for non-native speakers, ensuring native pronunciation, same speaker identity, duration, prosody, and precise alignment with the original non-native audio. Their AC model leverages TTS text representations to learn accent-independent features and uses synthetic ground-truth to learn the mapping function from non-native accent speech to native-like speech. This approach enables both AC and pronunciation correction while maintaining fast inference due to its non-autoregressive architecture.

Although this model achieves fast inference, it still requires the entire input for prediction, making it unsuitable for streaming applications. We adopt their approach of using native TTS to generate ideal ground truth. Recognizing this model as one of the best for AC, we focus on architectural and training modifications to develop a streaming-capable AC framework while maintaining the performance of its non-streaming counterpart.

Our work proposes the streaming AC by adapting and enhancing existing non-streaming AC models to operate in a streaming context. Our major contributions include significant architectural and training modifications that enable the model to process speech incrementally while maintaining high performance comparable to non-streaming models. Furthermore, we provide the implementation source code for streaming inference. This marks a substantial step forward in the field, as our streaming AC model is the first to effectively perform AC and pronunciation improvement in streaming fashion.

2. Methodology

This section presents the step-by-step training process for our streaming AC model. We begin by training the Native TTS model and demonstrating how it generates ideal ground-truth data for non-native speech. Next, we outline the architectural modifications needed for streaming AC and describe the training process using synthetic ground truth. Finally, we detail the inference procedure for streaming.

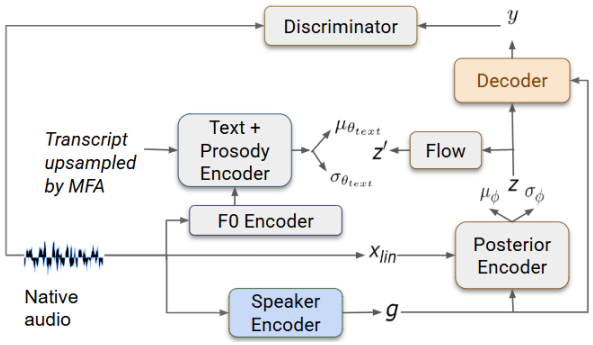

Figure 1: *Native TTS with prosody and speaker preserving*

2.1. Training Native TTS with prosody and speaker preserving

2.1.1. Training

We train the native TTS model using the VITS [18], a conditional variational autoencoder (CAVE) enhanced with normalizing flow. VITS consists of three key components: a posterior encoder define $q_\phi(z|x)$, a prior encoder define $p_\theta(z|c)$, and a waveform generator define $p_\psi(y|z)$, shown in Figure 1.

To align text and audio, VITS utilizes Monotonic Alignment Search (MAS). To speed up training and eliminate alignment learning, we employ the Montreal Forced Aligner (MFA) [19] to upsample text representations before training, ensuring they match the length of the audio. Our prior is conditioned on both the upsampled transcript and the F0 sequence.

The F0 Encoder extracts frame-level F0 embeddings, which are combined with text embeddings before being processed by a Transformer-based encoder. This Transformer-based encoder [20], enhanced with normalizing flow, generates the prior distribution $p_{\theta_{text}}(z|c)$ conditioned on text and prosody information. The posterior encoder uses linear spectrograms x_{lin} and speaker embeddings g to sample a latent variable $q_\phi(z|x)$, while the HiFi-GAN waveform generator [21] ψ reconstructs the waveform y from the latent variable z.

We use a downsampling rate of 320 for computing mel-spectrograms, F0, and linear spectrograms, while the HiFi-GAN decoder applies an upsampling rate of 320 accordingly. F0 sequences are extracted using the YAAPT algorithm [22] with the same downsampling rate, and the F0 encoder follows the approach in [23]. We utilize the pre-trained speaker encoder from [24]. Our training loss and hyperparameters is similar to the original paper [18].

2.1.2. Generate ground-truth from native TTS

The native TTS model trained earlier is used to generate ideal ground-truth audio for each non-native input. For each non-native audio, we use the MFA to find the alignment and upsample the phoneme transcript match the length of audio. Next, similar to inference process of VITS, we obtain the latent variable $z^{'}$ by sampling from $p_{\theta_{text}}(z^{'}|c_{upsample})$, which is then refined through an inverted normalizing flow f^{-1}. Finally, the ground-truth audio $\hat{y}_{\text{ground-truth}}$ is generated from $p_\psi(x|f^{-1}(z^{'}), g, F0)$ using the HiFi-GAN decoder, where speaker embedding g and $F0$ are extracted from original audio. In summary, the synthetic ground truth $y_{ground-truth}$ is generated from the transcripts with perfect native pronunciation, while utilizing the MFA alignment, $F0$ and g from the non-native audio to retain the original duration, prosody and speaker identity.

$$A = MFA(x_{non-native}, c_{text}) \tag{1}$$

$$c_{upsample} = upsampling(c_{text}, A) \tag{2}$$

$$z^{'} \sim p_{\theta_{text}}(z^{'}|c_{upsample}, F_0) \tag{3}$$

$$y_{ground-truth} = Hifigan(f^{-1}(z^{'}), g, F0) \tag{4}$$

2.2. Streaming non-autoregressive model

2.2.1. Architecture

CVAE architectures are also widely used and proved efficiency in speech-to-speech tasks such as voice conversion (VC) [25] and AC [17]. Unlike TTS, which uses text input for prior encoder, the AC or VC model takes content representation as input. This content representation can be derived from speech recognition models or self-supervised speech models. While the TTS process from text to audio is inherently a one-to-many problem, making the use of a conditional VAE to generate text-conditional prior distributions highly effective, AC or VC involves a one-to-one mapping from rich continuous content representation to audio. As a result, a CVAE may not be strictly necessary for this task. Therefore, to facilitate more straight forward training and inference for streaming AC, we simplified the model by using conventional autoencoder architecture, which include only four main components: a content encoder, a bottleneck extractor, a HiFi-GAN decoder, and a speaker encoder. Our content encoder is designed to extract both content representations and frame-based prosody features, eliminating the need for a separate prosody encoder like in the original non-streaming model. This content encoder is a pre-trained model that remains frozen during the training of the AC model. The bottleneck extractor eliminates accents and corrects pronunciation of content representation, while the HiFi-GAN decoder reconstructs native-like audio from the bottleneck extractor's output while injecting speaker embedding from the original speech.

The original AC content encoder utilized a Transformer-based ASR model to extract content representations from input audio. However, the conventional Transformer's global attention mechanism is not ideal for streaming applications. To overcome this limitation, we replaced it with Emformer [26], a Transformer variant optimized forlow latency streaming ASR [27, 28, 29] with lower latency. The Emformer is trained with 12 layers, a hidden size of 1024, a right look-ahead context of 8, a segment size of 4, and a left context of 30.

The bottleneck extractor was originally developed using a WaveNet-based [30] architecture with non-causal dilated convolutions, which demonstrated strong performance in its initial implementation. Although the bottleneck extractor and original HiFi-GAN decoder are not explicitly designed for streaming, their fully convolutional architecture inherently supports chunk-by-chunk processing, enabling streaming.

To leverage the high-quality audio generation capabilities of previous AC or VC models in a streaming setup, we retain both the WaveNet-based bottleneck extractor and HiFi-GAN decoder in our streaming model. To improve inference speed, we replace HiFi-GAN V1 from previous AC model with HiFi-GAN V2. Furthermore, we optimize the source code to facilitate efficient streaming inference for both components.

2.2.2. Training

The content encoder is trained on the Librispeech [31] and L2Arctic datasets using CMU-dict phoneme labels, optimizing

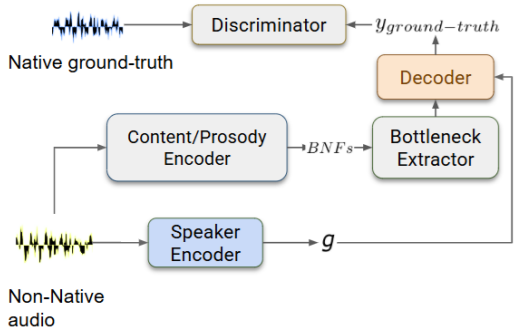

Figure 2: *Training streaming Accent Conversion Architecture with generated ideal ground-truth*

with CTC and prosody losses. This training allows the final layer's output to capture linguistic content and prosody features. The prosody loss is computed as the L1 loss between the predicted and ground truth log-F0 values. Based on our observations, the optimal coefficient for the prosody loss is set to 0.2. The loss function is described below:

$$L_{ASR} = (1 - \alpha) * L_{ctc} + \alpha * L_{prosody} \quad (5)$$

We implemented the Emformer based on the Torch Emformer implementation. We observed that the attention padding mask in the Torch Emformer exhibited an average sparsity of 70%, leading to substantial unnecessary computations in the multi-head scaled dot-product attention operation. To address this, we developed a custom implementation of the Torch Emformer using a Flex Attention function[32] , significantly optimizing the process. This improvement resulted in a threefold reduction in training time and enabled an eightfold increase in batch size compared to the original implementation. We also provide the source code for this enhanced implementation.

After training the content encoder, we pre-train the streaming AC model exclusively on native data, where both the input and output originate from the same native audio, allowing it to fully learn the distribution and characteristics of native speech. In the second stage, we fine-tune the pre-trained AC model using non-native input paired with the native generated ground-truth output. During this step, both native and non-native data are incorporated into the training process, following a 3:1 ratio between non-native and native audio. For native audio input, the output remains identical to the input, similar to the pre-training phase. This mixed-data approach ensures that the model effectively performs accent conversion and pronunciation correction for non-native speech while retaining the ability to recognize and preserve the accent and pronunciation of native speech, preventing unnecessary modifications. The loss function for both training stages is the HiFi-GAN loss, which consists of mel-spectrogram loss, feature loss, and discriminator loss.

2.2.3. Streaming Inference

It is noticeable that, during training the model has access to the future original frames to predict the current accent-converted frames. Being a fundamentally non-causal model, the look-ahead window is set at 0.64 seconds, accounting for the right context window of the Emformer content model, as well as the Hifi-GAN and WaveNet models. While we have access to the whole utterance during training, in inference the model receives chunk-based inputs, limited to 0.08 seconds (corresponding to the size of 4 segment lengths of the Emformer Model). Our streaming inference strategy is designed to ensure the output remains identical between streaming inference and full-segment inference, minimize the input latency, as well as the smoothness of the streaming output.

In practice, the model inference depends on a Voice Activity Detector (VAD) [33] to identify the range of speech within the input stream. Given the detected speech, in order to satisfy the desiderata above, the system waits for a total delay of 0.8 seconds (corresponding to 10 chunks of 0.08s) of audio before outputting the first two converted chunk and extracting the speaker embedding, which remains consistent throughout the entire speech segment. The player thread starts when the model outputs the second chunk, ensuring smooth and uninterrupted audio playback. Details of the streaming inference process are provided in Algorithm 0

Algorithm 1 Streaming AC Inference

Initialize: output_chunks ← [], input_chunks ← [], cache ← None
while start stream **do**
  chunk ← receive_new_chunk()
  input_chunks.append(chunk)
  if len(input_chunks) >= 10 **then**
    if cache == None **then**
      out, cache ← model.forward(input_chunks)
    else
      out, cache ← model.forward_with_cache(chunk, cache)
    end if
    output_chunks.append(out)
  end if
  if len(output_chunks) == 2 **then**
    player_thread.start_play()
  end if
end while

With a system-wide real-time factor for each chunk below 1 (achievable at around 0.25 on a GTX 1060 6GB), our streaming process ensures the chunks of output thread remains at least two chunks ahead of the player thread. As a result, the latency stays around 0.8s, regardless of the computational device, enabling smooth and efficient streaming. While latency could be reduced by shortening the look-ahead receptive field, this is beyond our scope. Instead, we focus on an efficient streaming framework that ensures high audio quality.

Notably, streaming inference is performed without any paddings, ensuring that the streaming output remains identical to full-segment processing. After each chunk is processed, the model caches (including Emformer, WaveNet-based component, and HiFi-GAN) are updated and stored, enabling efficient processing of subsequent chunks.

3. Experiment and Result

3.1. Data

We use the LJSpeech dataset [34], which contains recordings from a single native speaker with consistent pronunciation. To augment the dataset, we employ FreeVC [25], a voice conversion model, to generate multi-speaker native-accented utterances from the original voices. The augmented multi-speaker dataset used in our work consisted of approximately 65,000 utterances from 100 VCTK speakers. These were generated by augmenting the 13,000 utterances from the LJspeech dataset, with each utterance paired with five random speaker embed-

dings from the VCTK corpus [35]. This ensured a diverse representation of voices with one specific native-like pronunciation. This multi-speaker native dataset is then used to train the native TTS model in a multi-speaker setting and to pre-train the AC model in the first stage.

In the second stage, we fine-tune the AC model using the L2-ARCTIC dataset [36], which comprises speech from 24 accented speakers across six different accents. For each accent, we select three speakers for training, while the remaining speakers are used for testing. Each speaker's data is divided into a training set of 1,032 non-overlapping utterances, a validation set of 50 utterances, and a test set of 50 utterances. The test set is selected based on our competitive ASR model [37], focusing on utterances with an average Word Error Rate (WER) greater than 10 across all speakers, under the assumption that higher WERs indicate stronger accents and more pronunciation issues.

3.2. Evaluation metrics

3.2.1. Subjective tests

Nativeness and Speaker Similarity Tests: Ten participants performed two evaluations using a 5-point scale (1-bad, 2-poor, 3-fair, 4-good, 5-excellent). In the speaker similarity test, they rated how closely the voice identity of the converted audio matched the original input. In the nativeness test, they assessed how native-like the converted speech sounded.

3.2.2. Objective tests

Word error rate (WER): To assess pronunciation improvement, we measure WER using a competitive seq2seq Transformer ASR model [37], where lower WER indicates better intelligibility and pronunciation.

Accent classifier accuracy (ACC): We assess accent conversion effectiveness using an accent classifier trained to distinguish native from non-native speech, following the training setup of [5]. This metric is measured on both original non-native and converted native-like audio. A larger accuracy gap, where the classifier easily identifies the accent in original audio but struggles with the converted speech, indicates a stronger AC model in reducing non-native accent characteristics.

Speaker Embedding Cosine Similarity (SECS): Speaker similarity is evaluated using SECS, which computes cosine similarity between speaker embeddings of the original and converted speech. We extract embeddings using a state-of-the-art speaker verification model [13], where a similarity score above 0.85 suggests the audios are likely from the same speaker.

Mean Opinion Score Net (MOSNet): For output quality assessment, we use MOSNet [38] to predict scores highly correlated with human MOS ratings.

3.3. Experimental setup

Our streaming AC model is based on the fine-tuned version in the second stage. To assess its performance, we compare it with the non-streaming model introduced in [17].

Some causal streaming voice conversion models have been proposed [39]. However, in our accent conversion and pronunciation enhancement task, a look-ahead context is essential for accurate pronunciation improvement. To demonstrate this, we also train a causal streaming variant by replacing the non-causal CNN and transposed CNN of the Wavenet-based and Hifigan decoder with their causal counterparts. We analyze the audio quality of synthetic ground truth from the native TTS model. The training hyperparameters follow those of the original HiFi-GAN [21], with the system trained for 600,000 steps in the pre-training stage and 200,000 steps in the fine-tuning stage. In the both stages, the mel-spectrogram loss converges to approximately 0.3. Sample evaluation audios are available in [1].

3.4. Result

The objective metric evaluation in Table 2 shows that the streaming model performs comparably to the non-streaming model. Surprisingly, MOSNet assigns a higher score to our accent conversion output than to the original audio. This suggests that pre-trained MOSNet, which is primarily trained on non-accented speech, may give lower scores to non-native speech, even when it is real. A MOSNet score of approximately 4.1 indicates that our generated audio maintains good quality in terms of naturalness. Additionally, SECS remains stable at 0.85 and 0.84, verifying that the speaker identity is well-preserved in both settings. The ACC results are weaker compared to other metrics. Since our model aims to retain the original audio's prosody, which can sometimes reflect the speaker's accent, it occasionally classifies the converted audio as non-native. This prosody preservation may, in some cases, reduce the effectiveness of accent conversion. The causal streaming model performs worst across all metrics, highlighting the importance of look-ahead context. Subjective evaluation (Table 1) shows similar scores for streaming and non-streaming models, while strong WER performance indicates high-quality synthetic ground truth.

Table 1: *Subjective metrics*

Models	Nativeness	Sim-MOS
Original	1.12 ± 0.15	-
Synthetic ground-truth	3.93 ± 0.15	4.02 ± 0.17
Causal streaming Model	1.23 ± 0.12	3.75 ± 0.12
Streaming model	3.78 ± 0.18	3.92 ± 0.18
Non-Streaming model	3.87 ± 0.20	3.96 ± 0.19

Table 2: *Objective metrics*

Models	WER	ACC	SECS	MOSNet
Original	18.3	98.3	1.0	3.97
Synthetic ground-truth	4.7	11.5	0.83	4.18
Causal streaming Model	33.5	25.1	0.75	3.01
Streaming model	14.1	16.3	0.85	4.12
Non Streaming model	14.3	17.0	0.84	4.08

4. Conclusion

This work presents the first streaming accent conversion model, demonstrating its capability to synthesize high-quality audio with native-like pronunciation. Our results show that the streaming model achieves performance comparable to its non-streaming counterpart. We introduced an effective streaming inference method for the non-autoregressive accent conversion model. Although the current latency is not yet at a low-latency level (0.8s), it can be further reduced by decreasing the look-ahead receptive field of the entire model. Future work will focus on minimizing look-ahead latency while maintaining audio quality, as well as optimizing inference for low-power computing devices such as CPUs.

[1] https://accentconversion.github.io/streaming_demo

5. Acknowledgment

This research was supported by a grant from Zoom Video Communications , Inc. T; European Commission Project Meetween (101135798), the Federal Ministry of Education and Research (BMBF) of Germany under the number 01EF1803B (RELATER), and the pilot program Core-Informatics of the Helmholtz A and the HoreKa supercomputer funded by the Ministry of Science, Research and the Arts Baden-Württemberg and BMBF.

6. References

[1] A. Waibel and C. Fuegen, "Simultaneous translation of open domain lectures and speeches," Jan. 3 2012, uS Patent 8,090,570.

[2] 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*, 2022.

[3] T.-N. Nguyen, Q. Pham, and A. Waibel, "Accent conversion using discrete units with parallel data synthesized from controllable accented tts," in *Synthetic Data's Transformative Role in Foundational Speech Models*, 2024, pp. 51–55.

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

[5] T. N. Nguyen, N.-Q. Pham, and A. Waibel, "Accent Conversion using Pre-trained Model and Synthesized Data from Voice Conversion," in *Proc. Interspeech 2022*, 2022, pp. 2583–2587.

[6] Z. jia, H. Xue, X. Peng, and Y. Lu, "Convert and speak: Zero-shot accent conversion with minimum supervision," in *ACM Multimedia 2024*, 2024.

[7] T.-N. Nguyen, N.-Q. Pham, and A. Waibel, "Syntacc : Synthesizing multi-accent speech by weight factorization," in *ICASSP 2023*.

[8] T. Schultz, I. Rogina, and A. Waibel, "Lvcsr-based language identification," in *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*.

[9] C. Huber, E. Y. Ugan, and A. Waibel, "Code-switching without switching: Language agnostic end-to-end speech translation," *arXiv preprint arXiv:2210.01512*, 2022.

[10] T. Schultz and A. Waibel, "Experiments on cross-language acoustic modeling." in *INTERSPEECH*, 2001, pp. 2721–2724.

[11] D. Jia, Q. Tian, K. Peng, J. Li, Y. Chen, M. Ma, Y. Wang, and Y. Wang, "Zero-Shot Accent Conversion using Pseudo Siamese Disentanglement Network," in *Proc. INTERSPEECH 2023*, 2023.

[12] M. Jin, P. Serai, J. Wu, A. Tjandra, V. Manohar, and Q. He, "Voice-preserving zero-shot multiple accent conversion," in *ICASSP 2023*, 2023, pp. 1–5.

[13] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao, J. Wu, L. Zhou, S. Ren, Y. Qian, Y. Qian, J. Wu, M. Zeng, X. Yu, and F. Wei, "Wavlm: Large-scale self-supervised pre-training for full stack speech processing," *IEEE Journal of Selected Topics in Signal Processing*.

[14] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," in *NEURIPS 2020*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33, 2020.

[15] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, "Hubert: Self-supervised speech representation learning by masked prediction of hidden units," *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, 2021.

[16] M. Jin, P. Serai, J. Wu, A. Tjandra, V. Manohar, and Q. He, "Voice-preserving zero-shot multiple accent conversion," in *ICASSP 2023*, 2023, pp. 1–5.

[17] T. N. Nguyen, S. Akti, N. Q. Pham, and A. Waibel, "Improving pronunciation and accent conversion through knowledge distillation and synthetic ground-truth from native tts," in *ICASSP 2025*.

[18] J. Kim, J. Kong, and J. Son, "Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech," in *International Conference on Machifi-ne Learning*, 2021.

[19] M. McAuliffe, M. Socolof, S. Mihuc, M. Wagner, and M. Sonderegger, "Montreal forced aligner: Trainable text-speech alignment using kaldi," in *Interspeech 2017*, 2017, pp. 498–502.

[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., 2017.

[21] J. Kong, J. Kim, and J. Bae, "Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., 2020.

[22] A. Kroon, "Comparing conventional pitch detection algorithms with a neural network approach," 2022.

[23] 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 *Interspeech 2021*, 2021, pp. 1344–1348.

[24] S. Liu, Y. Cao, D. Wang, X. Wu, X. Liu, and H. Meng, "Any-to-many voice conversion with location-relative sequence-to-sequence modeling," *TASLP 2021*, 2021.

[25] J. Li, W. Tu, and L. Xiao, "Freevc: Towards hifigagh-quality text-free one-shot voice conversion," in *ICASSP 2023*, 2023, pp. 1–5.

[26] Z. Hao, H. Quan, and Y. Lu, " EMF-former: An Efficient and Memory-Friendly Transformer for Medical Image Segmentation ," in *proceedings of Medical Image Computing and Computer Assisted Intervention – MICCAI 2024*, 2024.

[27] T.-S. Nguyen, S. Stüker, and A. Waibel, "Super-human performance in online low-latency recognition of conversational speech," in *Interspeech 2021*, 2021, pp. 1762–1766.

[28] J. Niehues, N.-Q. Pham, T.-L. Ha, M. Sperber, and A. Waibel, "Low-latency neural speech translation," 2018.

[29] J. Niehues, T. S. Nguyen, E. Cho, T.-L. Ha, K. Kilgour, M. Müller, M. Sperber, S. Stüker, and A. Waibel, "Dynamic transcription for low-latency speech translation." in *Interspeech*, 2016.

[30] A. van den 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," 2016. [Online]. Available: https://arxiv.org/abs/1609.03499

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

[32] J. Dong, B. Feng, D. Guessous, Y. Liang, and H. He, "Flex attention: A programming model for generating optimized attention kernels," 2024.

[33] S. S. Sarfjoo, S. Madikeri, and P. Motlicek, "Speech activity detection based on multilingual speech recognition system," in *Interspeech 2021*, 2021, pp. 4369–4373.

[34] K. Ito and L. Johnson, "The lj speech dataset," https://keithito.com/LJ-Speech-Dataset/, 2017.

[35] J. Yamagishi and C. Veaux, "CSTR VCTK Corpus: English multi-speaker corpus for CSTR voice cloning toolkit," 2019.

[36] 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 *Interspeech 2018*, 2018.

[37] N.-Q. Pham, T.-L. Ha, T.-N. Nguyen, T.-S. Nguyen, S. Stüker, J. Niehues, and A. Waibel, "Relative positional encoding for speech recognition and direct translation," in *Interspeech 2020*.

[38] T. Rosenbaum, I. Cohen, E. Winebrand, and O. Gabso, "Differentiable mean opinion score regularization for perceptual speech enhancement," *Pattern Recognition Letters*, vol. 166, 2023.

[39] Y. Yang, Y. Kartynnik, Y. Li, J. Tang, X. Li, G. Sung, and M. Grundmann, "Streamvc: Real-time low-latency voice conversion," in *ICASSP 2024*, 2024, pp. 11 016–11 020.