2024 IEEE 14th International Symposium on Chinese Spoken Language Processing (ISCSLP) | 979-8-3315-1682-6/24/$31.00 ©2024 IEEE | DOI: 10.1109/ISCSLP63861.2024.10800531

LLM-based Expressive Text-to-Speech Synthesizer with Style and Timbre disentanglement

Yuanyuan Zhu[1], *Jiaxu He*[1], *Ruihao Jing*[1,2], *Yaodong Song*[1], *Jie Lian*[1], *Xiao-lei Zhang*[1,2], *Jie Li*[1*]

[1]Institute of Artificial Intelligence (TeleAI), China Telecom, Beijing
[2]School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an

zhuyy17@chinatelecom.cn, hejx15@chinatelecom.cn, songyd@chinatelecom.cn,
lianj1@chinatelecom.cn, lij86@chinatelecom.cn, {ruihaojing, xiaolei.zhang}@nwpu.edu.cn

Abstract

The ICACG challenge Track 1 requires to generate the target speaker audios with high naturalness under extremely limited speaker dataset. To achieve this goal, we introduce a novel text-to-speech synthesizer which disentangles the style and timbre information in cascade approach. Firstly, an auto-regressive large language model (LLM) is applied to complete the text-to-token generation, which can capture the style information from audio prompt in zero-shot mode. Subsequently, a variational generator is used to reconstruct the mel-spectrogram in corresponding to the target speaker timbre conditioned on speaker embedding. Therefore, the final synthesized audio can not only contain the timbre of the target speaker, but also achieve a high degree of expressiveness by utilizing the capabilities of LLM. Since large and diverse data is necessary for training, a novel data processing pipeline is also proposed to process the collected data. As a result, our system achieved great performance in terms of expressive speech synthesis and ranked the first place in ICAGC 2024 Track 1 over all five evaluation metrics: 3.89 Quality, 3.83 Similarity, 3.85 Emotion, 3.89 MOS(avg) and 0.22 MOS (std).

Index Terms: text-to-speech, timbre disentanglement, data processsing

1. Introduction

The inspirational and convincing audio generation challenge 2024 (ICAGC 2024) [1] aims to enhance the persuasiveness and acceptability of synthesized audio, focusing on human alignment convincing and inspirational audio generation. The objective of Track 1 in challenge is to synthesize the audio with the target speaker timbre and convincing emotions for different text theme. The total number of target speakers is ten, and the total audio length per speaker is less than 14.5 minutes. The given test texts encompass themes from various domains, including novel chapters, ancient Chinese poems, etc.

Due to the data limitation, the traditional text-to-speech (TTS) systems [2, 3, 4] fail to support this task, because most of them are trained on limited datasets recorded in studios and rely on the speaker adaptation for unseen speakers. Recent advancements based on LLM models [5, 6, 7, 8, 9] have shown remarkable performance in zero-shot TTS, which can clone a timbre and prosody with just a few seconds of audio prompt. As shown in Fig.1, speech is usually tokenized into discrete tokens [10, 11, 12, 13], which makes model more robust to the noise and data quality. In such TTS system, the LLM model is applied to accomplish the next-token prediction task. By leveraging large and diverse data as much as possible, LLM models are empowered of the strong in-context learning capabilities. Besides, the diversity of the generated acoustic tokens can be improved by using different sampling strategies during inference stage [5].

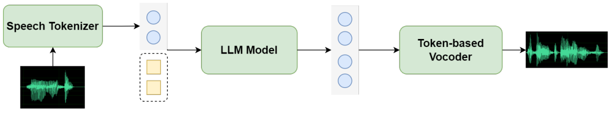

Figure 1: *An overview of LLM-based TTS model inference pipeline. The blue circles represent speech tokens and yellow squares represent text tokens. (1) The audio prompt is converted to prompt tokens via speech tokenizer. (2) The LLM model autogressively generates the speech tokens with condition of text and speech. (3) The token-based vocoder reconstructs the waveform given the intermediate token.*

The first language model based TTS framework is VALLE [5], which synthesizes the personalized speech with 3-second enrolled recording as audio prompt. Then, VALLE-X [14] is proposed for zero-shot cross-lingual TTS and zero-shot speech-to-speech translation tasks. However, the monotonic association between phoneme sequences and audio is aligned by self-attention mechanism, which brings the pose robustness issues such as typos, omissions and repetition. To solve this problem, VALL-E R [15] introduces a phoneme monotonic alignment strategy to strengthen the generation stability. The series of VALLE-related works is completed based the audio codes extracted from the direct waveforms. Differently, XTTS [16] and Single-Codec [17] propose to employ the Vector Quantised-Variational AutoEncoder (VQ-VAE) to encode the mel-spectrogram into latent codes. BASETTS [18] discretizes the features extracted from a WavLM Self-Supervised Learning (SSL) [19] model to reconstruct the mel spectrogram. Specifically, BASETTS [18] also reveals that the large language models begin to demonstrate the "emergent abilities", more natural prosody on complex sentences, when trained on increasing volume of data.

Inspired by the success of LLM-based TTS models, we introduce an expressive TTS synthesizer to solve the Track 1 challenge. The whole framework is a cascaded system. Firstly, the LLM model predicts the speech tokens which are constructed by multi-lingual Wav2Vec [20] with K-means. A variantional autoencoder [21] as a voice conversion model is applied to generate the mel spectrogram from the speech tokens. Finally, the HifiGAN vocoder [22] is used to synthesize a waveform with the generated mel as input. To encourage the disentangement between the timbre and style, our strategy is to limit the responsibilities of the auto-regressive model to captures the phoneme

[*] Corresponding author

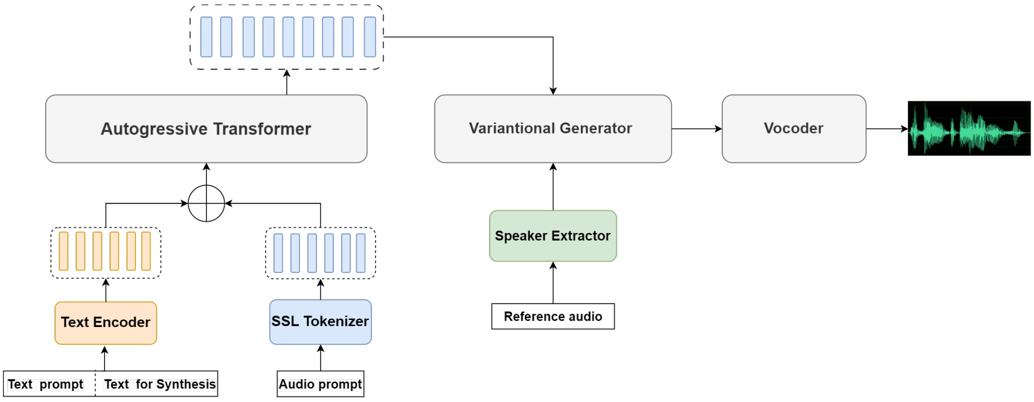

Figure 2: *An overview of the proposed LLM-based TTS model*

contents, duration and prosody, while designating a separate token-to-mel decoder with the reconstruction of speaker identity. Additionally, it's worth noting that the performance of the current LLM-based TTS models significantly degrades with the reduction of training dataset volume. Thus, we also propose a data processing pipeline to transform in-the-wild speech data into high-quality training data with annotations for speech generation. In a conclusion, our contributions are mainly follows:

- A high effective data processing pipeline is designed, which can process one hour of raw speech data ready for model training in a few minutes.
- A LLM-based model is designed based on discreted SSL features. In order to further improve the style migration and high expressiveness of the synthesized audios, we propose to sum the text tokens and speech tokens for prompt part and design two loss functions for training.
- A variational generator is to applied to construct the mel-spectrogram from speech tokens. To reconstruct the speaker timbre, we incorporate the speaker embedding into the VAE model. Actually, the variational model can be directly used as a voice conversion model to convert the speech tokens from other speaker audio to the mel-spectrogram containing the target speaker timbre.

Our experimental results demonstrate the superiority of the proposed LLM-based TTS model in expressive speech synthesis. The rest of the paper details the whole system and experiments.

2. Methods

Since the large and diverse dataset is foundational to the success of LLM-based TTS model, we design a data processing pipeline to process the in-the-wild speech audios. The proposed model consists of three main stages, as shown in Fig. 2. Firstly the auto-regressive transformer model generates the speech tokens conditioned on text prompt and audio prompt, which aims to control the phonetic and prosodic information. Secondly, an variantioal generator reconstructs the Mel spectrogram with target speaker timbre. Finally, a vocoder synthesizes a waveform.

2.1. Data processing

As illustrated in Fig.3, the whole data processing pipeline includes seven steps:

1. Standarization: Since the collected data vary in encoding formats and contain some bad data, all data are converted to mono channel, 16bit and 16kHz audios as WAV files. A unique name is generated for each audio based on its source, style and so on. Bad data like empty audio, pure noise are filtered.
2. Speech Enhancement: A speech enhancement model in waveform domain [23] is used to extract the cleaner human vocals and reduce the impact of noise. Then, the enhanced speech is used to calculate the signal-to-noise ratio (SNR). In addition, we employ the speech super-resolution technique [24] to generate high-frequency components for severely damaged audios.
3. VAD Segment: Given the enhanced speech data, we use an open-sourced voice active detection[1] (VAD) tool to segment the audios. By merging the shorter segments and splitting the long one, the final speech segments range from 3 to 20 seconds.
4. Speech Diarization: To ensure the speaker label of each speech segment, we employ an open-source speaker diarization toolkit[2] to determine both the number of speakers and the speaker assignments.
5. Quality Filter: To ensure the quality of final speech, we further employ data filtering methods. The speech segments with SNR lower than 15db and more than one speaker are filterd.
6. Audio Labeling: To understand audios, we apply the emotion2vec [25] to determine the utterance-level emotion. By finetuning the wavLM model [19], we classify the gender and ages of voice in each audio. The variables like speaking rate, energy and pitch are estimated following the open-source toolkit[3].
7. Transcriptions Normalization: Finally, we employ the neural automatic speech recognition (ASR) model to process the audios without transcriptions. In order to obtain the accurate text transcriptions, two ASR models are used. The audio files with similarity of two sets of transcriptions greater than 80% are ultimately retrained. One ASR system is FunASR [26], another is specialized in-house developed system. Additionally, the punctuations are further restored based on the recognized text results and silence in the audio.

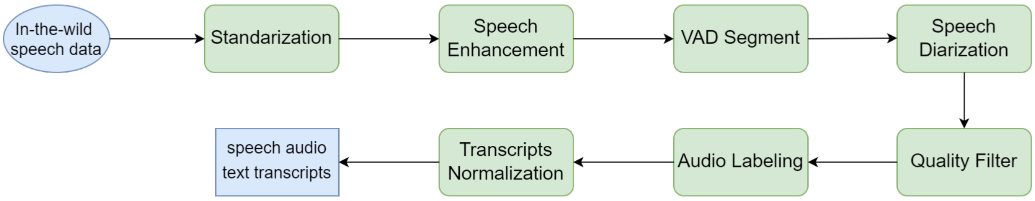

Figure 3: *The diagram of the proposed data preprocessing pipeline*

In general, the resulting pipeline can process 1.0 hours of raw speech data in one minute using an independent server with eight NVIDIA RTX A100 GPUs.

2.2. Discrete speech representations

In LLM-based model, it is foundational to seek an appropriate discrete representation for speech tokenization and reconstruction. In this work, we aim to develop the speech tokens that contain the phonetic and style information, which are disentangled from speaker identity.

Thus, the waveform are passed into the multi-lingual Wav2Vec [20] to extract the hidden states of the fifteenth layer. Then, a K-Means model is trained in these features to obtain a discrete speech tokens. The representations are compressed (550 bits/s) to allow more efficient auto-regressive modeling compared to popular audio codes (e.g. 6k bits/s in [12]]). With this level of compression, we aim to remove the speaker-related information from speech codes that can be reconstructed during decoding to ensure that the capacity in speech codes is primarily dedicated to encoding phonetic and style information.

2.3. Auto-regressive speech modeling

In LLM-based model, we formulate the TTS task as an auto-regressive speech token generation problem. The auto-regressive model structure is largely identical to the GPT-2 language model [27], which is called as ”GPT-Speech” in this work. In the training stage, we randomly select a audio clip at the beginning of a audio segment as the audio prompt. The text tokens and speech tokens corresponding to the prompt part are summed together instead of stacking into an embedding, which can improve the expressiveness of synthesized speech. Both the prompt tokens are encoded to a fixed size.

The GPT-Speech model is trained from scratch, without pretraining on text. Furthermore, we design two loss functions for training . First, the GPT-Speech model is trained to generate the speech tokens corresponding to the remaining part of audio segment. Besides, in order to retrain the textual information to guide prosody, we also train the GPT-Speech model with an objective to predict the text tokens for the whole segment.

2.4. Waveform generation

We first use a VAE generator [21] to reconstruct the mel-spectrogram from the speech tokens. To ensure the speaker identity, speaker embedding is extracted by the open-source tool Resemblyzer[4] so that the mel-spectrogram obtain the timbre of target speaker. Actually, the VAE generator can be considers as a separate voice conversion model. It primarily consists of two parts. First, the speech tokens are passed through an encoder composed of transformers, which aims to convert the original discrete representations into continuous, high-dimensional, context-aware representations. Simultaneously, the speaker information is used as an input to guide the timbre. Then, the obtained representations are processed by a Flow-VAE decoder (FVAE).

The FVAE consists of three components: the encoder, the decoder, and a flow model. Both the encoder and decoder are mainly built on convolutional layers. Meanwhile, the flow model incorporates multiple invertible transformations, notably Coupling Layers, to achieve precise density estimation and sampling within the latent space. During training, the model is optimized to minimize reconstruction loss and Kullback-Leibler (KL) divergence, ensuring that the generated features closely align with labels and maintain structured latent representations.

HifiGAN [22] comprises a generator, multi-period discriminators, and multi-scale discriminators, which is used to produce high-quality speech waveforms. The training process of generator involves a combination of adversarial loss and reconstruction loss. The discriminators are optimized using a binary cross-entropy loss to improve the classification accuracy.

3. Experiments

3.1. Dataset

Leveraging the proposed data processing pipeline, we construct a multilingual dataset of a collection of speech data from a wide range of video platforms and podcasts, containing diverse speaking styles of real human speech. The total collected dataset contains over 95k hours of speech data at 16kHz and mainly covers over two languages: Chinese and English.

3.2. Training and hyperparameters

The three main modules, auto-regressive transformer, VAE generator and vocoder, are trained separately. A pre-trained checkpoint[5] is used for Wav2vec model. The codebook size is 2048 for the speech tokens and a vocabulary size is 14319. We train a 24-layer decoder-only transformer on our internal datasets for 90k steps with 8 A100-40G GPUs. The ScaledAdam optimization is used with max learning rate of 0.025 and warmup steps of 200.

The encoder and decoder of VAE generator is built on 8 convolutions and 4 convolutions respectively, which is trained with the learning rate of 2.0e-4. The HiFiGAN is trained with adjusted dilation sizes within the residual blocks set to [[1, 3, 5],[1,3,5],[1,3,5]] and a convolutional kernel size of [3,7,11].

[1] https://github.com/wiseman/py-webrtcvad

[2] https://huggingface.co/pyannote/speaker-diarization-3.1

[3] https://github.com/huggingface/dataspeech

[4] https://github.com/resemble-ai/Resemblyzer

[5] https://huggingface.co/facebook/wav2vec2-large-xlsr-53

Table 1: *The evaluation results of top-5 systems on the blind test-set in Track 1 challenge*

Rank	**Team**	**Quality**	**Similarity**	**Emotion**	**MOS(avg)**	**MOS(std)**
1	**Our team**	3.89	3.83	3.85	3.86	0.22
2	NPU	3.84	3.48	3.58	3.63	0.30
3	zyzx_ai	3.67	3.50	3.50	3.56	0.42
4	PeitangTTSer	3.64	3.52	3.28	3.48	0.39
5	Happy Happy	3.33	3.45	3.33	3.37	0.36

3.3. Results

3.3.1. Evaluations on Generation Quality

We compare the proposed model with other three industry TTS system, iFlytek[6], Unisound[7] and fastpitch [28]. The audios generated by iFlytek is from the ”vcn.x4_lingfeizhe_oral” voice from iFlytekSpark model. The voice for Unisound is ”chenyang-normal-plus”. For our proposed method, the audios is generated by using a 38 seconds reference clip of a Chinese male speaker.

Two sets of test cases are constructed: one is 15 sentences with 1-50 words and another is 15 sentences within 50-100 words. The total duration of synthesized audios per system ranges 10 minutes or so. For each test set, we organized 30 professional evaluators to score the audios synthesized by three systems according to the rules shown in Table 2. The average score of total evaluation results are show in Fig.4.

Table 2: *Metric for evaluating the TTS system*

Metric	**Score**	**Weight**
Pronunciation Accuracy	10	20%
Naturalness and Fluency	10	60%
Audio Quality	10	20%

Compared with other TTS models, the proposed framework achieves comparable performance in terms of pronunciation accuracy, naturalness and audio quality. The long text transcription which exceeds 50 words makes little degradation in the performance of audio synthesis for the proposed model. Additionally, we observe that the scores of Unisound and Fastpitch have a significant difference compared to iFlytek and the proposed model. It is also demonstrates that the LLM-based TTS model is able to achieve better performance than the traditional TTS algorithms.

3.3.2. Evaluation on in-context Learning Synthesis

In terms of in-context learning synthesis, we compare the proposed model with other TTS teams on the test set in Track 1 of ICAGC challenge. Through the experiments, we have found that the GPT-Speech model can capture the style information for unseen speaker without fine-tuning, but the variantional generator can extract the better timbre information by speaker adaption. Thus, we adopt a data augment strategy to enlarge the datasets.

First of all, all the provided data are classified by the corresponding style. Notably, only speaker1 and speaker2 have four and three distinct styles respectively, while the other speakers have one single style. Subsequently, the degraded speech audios from speaker4 and speaker6 are processed by speech super-resolution model [24] to restore the high-frequency components. Then, all data are used to fine-tune the open-source model[8] to generate the audios for each speaker. However, although the synthesized audios maintain the timbre information, there are many bad cases such as repetitions, omissions or even pure noise. We have filterd out the low-quality audios, and finally audios per speaker amounts to approximately 1 hour. The obtained audios are used to fine-tune the variantional generator in the proposed algorithm.

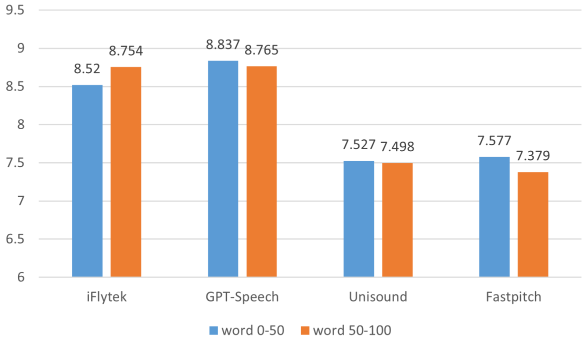

Figure 4: *Comparison results with other TTS models in terms of pronunciation accuracy, naturalness and audio quality*

Notably, due to the disentanglement between style and timbre in the proposed model, we can use different reference audio in LLM model and variantional generator. In detail, the audio prompt used in LLM model can be selected by audio style instead of timbre, while the reference audio for VAE model must be selected from the target speaker audios. After obtaining all the 16kHz generated audios from the proposed model, we also use the super-resolution model [24] to get the 48k audios to further improve the speech quality.

In the evaluation of the ICAGC challenge Track 1, we achieved great speaker similarity with 3.83 score and high emotional expressiveness with 3.85 score, which ranked the first place. The Table 1 showed the metrics of top-5 teams on the test-set in five evaluation metrics.

4. Conclusion

In this work, we introduced an expressive speech synthesizer for ICAGC challenge Track 1. Experimental results have demonstrated the state-of-the-art performance in generation quality and in-context leanring for the proposed system.

[6] https://console.xfyun.cn/services/medd90fec
[7] https://ai.unisound.com/doc/
[8] https://github.com/RVC-Boss/GPT-SoVITS

5. References

[1] R. Fu, R. Liu, C. Qiang, Y. Gao, T. W. Yi Lu, Y. Li, Z. Wen, C. Zhang, H. Bu, Y. Liu, S. Shi, X. Qi, and G. Li, "Inspirational and Convincing Audio Generation Challenge 2024 ICAGC 2024," in *The 14th International Symposium on Chinese Spoken Language Processing (ISCSLP 2024)*, 2024, http://www.iscslp2024.com/Icagc.

[2] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, Q. Le, Y. Agiomyrgiannakis, R. Clark, and R. A. Saurous, "Tacotron: Towards end-to-end speech synthesis," 2017. [Online]. Available: https://arxiv.org/abs/1703.10135

[3] X. T. T. Q. S. Z. Z. Z. Yi Ren, Yangjun Ruan and T.-Y. Liu, "Fastspeech: Fast, robust and controllable text to speech," 2019.

[4] R. J. W. M. S. N. J. Z. Y. Z. C. Y. Z. Y. W. R. S.-R. e. a. Jonathan Shen, Ruoming Pang, "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions," 2018, pp. 4779–4783.

[5] C. Wang, S. Chen, Y. Wu, Z. Zhang, L. Zhou, S. Liu, Z. Chen, Y. Liu, H. Wang, J. Li, L. He, S. Zhao, and F. Wei, "Neural codec language models are zero-shot text to speech synthesizers," 2023.

[6] P. Peng, S.-W. Li, P.-Y. Huang, A. Mohamed, and D. Harwath, "Voicecraft: Zero-shot speech editing and text-to-speech in the wild," *ACL*, 2024.

[7] S. Team, "Seed-tts: A family of high-quality versatile speech generation models," 2024.

[8] Z. Borsos, M. Sharifi, D. Vincent, E. Kharitonov, N. Zeghidour, and M. Tagliasacchi, "Soundstorm: Efficient parallel audio generation," 2023.

[9] C. Du, Y. Guo, F. Shen, Z. Liu, Z. Liang, X. Chen, S. Wang, H. Zhang, and K. Yu, "Unicats: A unified context-aware text-to-speech framework with contextual vq-diffusion and vocoding," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 16, p. 17924–17932, Mar. 2024. [Online]. Available: http://dx.doi.org/10.1609/aaai.v38i16.29747

[10] A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, "High fidelity neural audio compression," 2022.

[11] R. Kumar, P. Seetharaman, A. Luebs, I. Kumar, and K. Kumar, "High-fidelity audio compression with improved rvqgan," 2023.

[12] N. Zeghidour, A. Luebs, A. Omran, J. Skoglund, and M. Tagliasacchi, "Soundstream: An end-to-end neural audio codec," 2021.

[13] 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," 2021.

[14] Z. Zhang, L. Zhou, C. Wang, S. Chen, Y. Wu, S. Liu, Z. Chen, Y. Liu, H. Wang, J. Li, L. He, S. Zhao, and F. Wei, "Speak foreign languages with your own voice: Cross-lingual neural codec language modeling," 2023.

[15] B. Han, L. Zhou, S. Liu, S. Chen, L. Meng, Y. Qian, Y. Liu, S. Zhao, J. Li, and F. Wei, "Vall-e r: Robust and efficient zero-shot text-to-speech synthesis via monotonic alignment," 2024.

[16] E. Casanova, K. Davis, E. Gölge, G. Göknar, I. Gulea, L. Hart, A. Aljafari, J. Meyer, R. Morais, S. Olayemi, and J. Weber, "Xtts: a massively multilingual zero-shot text-to-speech model," 2024.

[17] H. Li, L. Xue, H. Guo, X. Zhu, Y. Lv, L. Xie, Y. Chen, H. Yin, and Z. Li, "Single-codec: Single-codebook speech codec towards high-performance speech generation," 2024.

[18] M. Łajszczak, G. Cámbara, Y. Li, F. Beyhan, A. van Korlaar, F. Yang, A. Joly, Álvaro Martín-Cortinas, A. Abbas, A. Michalski, A. Moinet, S. Karlapati, E. Muszyńska, H. Guo, B. Putrycz, S. L. Gambino, K. Yoo, E. Sokolova, and T. Drugman, "Base tts: Lessons from building a billion-parameter text-to-speech model on 100k hours of data," 2024.

[19] 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, and F. Wei, "Wavlm: Large-scale self-supervised pre-training for full stack speech processing," 2021.

[20] A. Conneau, A. Baevski, R. Collobert, A. Mohamed, and M. Auli, "Unsupervised cross-lingual representation learning for speech recognition," 2020. [Online]. Available: https://arxiv.org/abs/2006.13979

[21] Y. Ren, J. Liu, and Z. Zhao, "Portaspeech: Portable and high-quality generative text-to-speech," 2022. [Online]. Available: https://arxiv.org/abs/2109.15166

[22] J. Kong, J. Kim, and J. Bae, "Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis," 2020. [Online]. Available: https://arxiv.org/abs/2010.05646

[23] A. Defossez, G. Synnaeve, and Y. Adi, "Real time speech enhancement in the waveform domain," in *Interspeech*, 2020.

[24] S.-H. Lee, H.-Y. Choi, S.-B. Kim, and S.-W. Lee, "Hierspeech++: Bridging the gap between semantic and acoustic representation of speech by hierarchical variational inference for zero-shot speech synthesis," 2023. [Online]. Available: https://arxiv.org/abs/2311.12454

[25] Z. Ma, Z. Zheng, J. Ye, J. Li, Z. Gao, S. Zhang, and X. Chen, "emotion2vec: Self-supervised pre-training for speech emotion representation," 2023.

[26] Z. Gao, Z. Li, J. Wang, H. Luo, X. Shi, M. Chen, Y. Li, L. Zuo, Z. Du, Z. Xiao, and S. Zhang, "Funasr: A fundamental end-to-end speech recognition toolkit," in *INTERSPEECH*, 2023.

[27] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," 2019.

[28] A. Łańcucki, "Fastpitch: Parallel text-to-speech with pitch prediction," 2021. [Online]. Available: https://arxiv.org/abs/2006.06873