Text-to-Speech Integration With Voice Cloning

Yuhao Zhao 36920241153284[1] **Yulei Gong 36920241153205**[1] **Qianyun Wu 20520240156948**[2]

[1]Class AI [2]Class Informatics

Abstract

Text-to-speech (TTS) systems have made significant advancements with the application of deep learning, producing speech that is increasingly natural and expressive. However, challenges persist in the areas of personalization and improving synthesis clarity, particularly in few-shot learning scenarios. This study focuses on GPT-SoVITS, a novel model designed for few-shot text-to-speech and voice cloning, GPT-SoVITS allows fine-tuning with just one minute of training data, significantly enhancing the realism and speaker similarity of the generated speech. Despite these advancements, the model faces limitations when synthesizing longer sentences, with issues such as audio loss and low clarity. This study explores methods to address these issues by improving the model architecture and training processes, with the goal of significantly enhancing overall performance.

Introduction

Text-to-speech (TTS) aims to synthesize intelligible and natural speech (Tan et al. 2021). The emergence of deep learning has revolutionized the field of speech synthesis, leading to the development of highly sophisticated TTS systems. Traditional TTS approaches often struggled with naturalness and expressiveness, limiting their application in human-computer interactions (van den Oord et al. 2016). Recent innovations, particularly in voice cloning technologies, have paved the way for more personalized and context-aware speech synthesis (Chen et al. 2021) (Yan et al. 2021). Voice cloning allows for the reproduction of a specific speaker's voice, capturing unique characteristics such as tone, pitch, and emotional inflections.

The GPT-SoVITS model addresses these challenges through its unique features: few-shot text-to-speech (TTS) and cross-lingual support. Few-shot TTS allows for fine-tuning the model with just one minute of training data, significantly enhancing voice similarity and realism. This is particularly beneficial in scenarios where obtaining extensive voice samples is impractical (Wang et al. 2023). However, while GPT-SoVITS demonstrates great potential, it still faces challenges with longer sentences, where issues such as audio loss and low clarity may occur, affecting the quality of the output. If this problem can be effectively solved, it will greatly increase the ability of GPT-SoVITS to solve complex sentences.

In this paper, we first review the relevant literature on text-to-speech (TTS) and voice cloning technologies, highlighting the limitations of existing systems. We then outline our proposed method, detailing the architecture of the GPT-SoVITS model and the training process used. Next, we present feasible solutions to some current issues and conduct experiments aimed at generating more personalized and naturally sounding speech.The findings of this research will contribute to the ongoing development of TTS systems that can seamlessly adapt to individual user preferences.

Related Work

Traditional and Neural Network-Based TTS

Early text-to-speech (TTS) systems relied on concatenative methods or statistical parametric models. Concatenative TTS, such as the system proposed in Hunt and Black (Hunt and Black 1996), involved piecing together pre-recorded speech units to form words and sentences, which often resulted in synthesized speech lacking in naturalness, especially when generating prosody and emotional expressions. Statistical parametric models, including Hidden Markov Models (Zen, Tokuda, and Black 2009), provided more flexibility in generating speech but suffered from a "muffled" or "robotic" sound.

The emergence of deep learning has significantly improved speech synthesis, especially in terms of naturalness and expressiveness(Kong, Kim, and Bae 2020). Neural network-based models, such as Tacotron (Wang et al. 2017) and Tacotron 2 (Shen et al. 2018), have replaced traditional methods by learning the mapping from text to mel-spectrograms, which are then converted into audio waveforms using neural vocoders like WaveNet (van den Oord et al. 2016). These models allow for more flexible and natural-sounding speech synthesis, capable of dynamically adjusting pitch, duration, and intensity.

Few-shot learning has gained traction in TTS due to the difficulty of obtaining large amounts of labeled speech data for individual speakers. VITS combines variational autoencoders and normalizing flows to generate high-quality speech from limited data. Similarly, AdaSpeech (Chen et al. 2021)introduces adaptive mechanisms to enhance speech

naturalness and consistency when trained on few-shot data. These approaches are particularly useful in applications such as personalized voice assistants, where large amounts of speech data for each user are impractical.

Voice Cloning with Deep Learning

Voice cloning technology has advanced significantly with the use of deep learning(Arik et al. 2018). A neural network-based system for TTS synthesis (Jia et al. 2019) introduced a speaker encoder capable of generating a voice embedding from a small amount of data, enabling high-quality voice cloning with minimal input. Other systems, such as AdaSpeech (Chen et al. 2021) (Yan et al. 2021), have pushed the boundaries of cross-lingual voice cloning, allowing models to generalize across languages that were not present in the training data. This is essential in multilingual contexts where TTS systems need to support multiple languages.

VALL-E

VALL-E is a TTS system based on Neural Codec Language Modeling (Wang et al. 2023). Its design consists of three main modules: Phoneme Conversion, Audio Codec Encoder and Neural Codec Language Modeling. The system takes two inputs: a text prompt and an acoustic prompt (a 3-second audio recording of the target voice). The phoneme conversion module processes the text input into phoneme representations, while the audio codec encoder encodes the acoustic prompt into discrete acoustic tokens. These inputs are then passed into the core module—Neural Codec Language Modeling, which employs a pre-trained Transformer model to learn the semantic relationship between the speech and text inputs, generating personalized acoustic token sequences. Finally, the audio codec decoder converts the generated acoustic tokens into high-quality personalized speech.

VALL-E has been pre-trained on a large-scale speech corpus, allowing it to generalize well to a wide range of voices and languages with minimal data.The system excels not only in voice cloning but also in emotional speech synthesis and style adaptation, making it a versatile tool for a variety of speech tasks. Furthermore, VALL-E achieves near real-time synthesis, providing high-quality results with low latency. This positions VALL-E as a competitive option for applications such as virtual assistants, audiobooks, and accessibility tools, where high-fidelity, personalized speech is required.

Figure 1 demonstrates how VALL-E integrates text and voice prompts to achieve efficient and personalized speech synthesis.

GPT-SoVITS

Building on these advancements, the GPT-SoVITS model introduces a few-shot learning paradigm combined with cross-lingual capabilities. Few-shot learning enables the model to be fine-tuned with just one minute of speaker data, significantly improving voice cloning efficiency and realism. GPT-SoVITS also expands cross-lingual support, covering English, Japanese, Korean, Cantonese, and Chinese, making it a versatile solution for multilingual applications. However, challenges remain in synthesizing longer sentences, where the model can suffer from audio dropouts and reduced intelligibility.

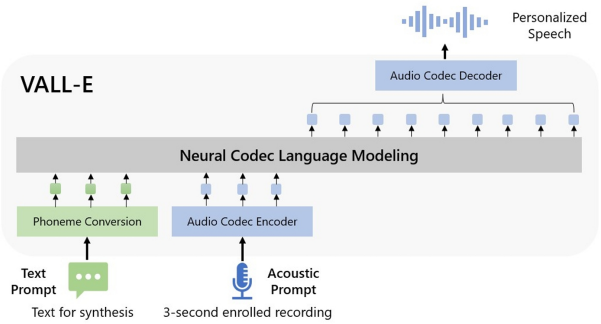

Figure 1: The architecture of VALL-E, illustrating its main components: Phoneme Conversion, Audio Codec Encoder, Neural Codec Language Modeling, and Audio Codec Decoder

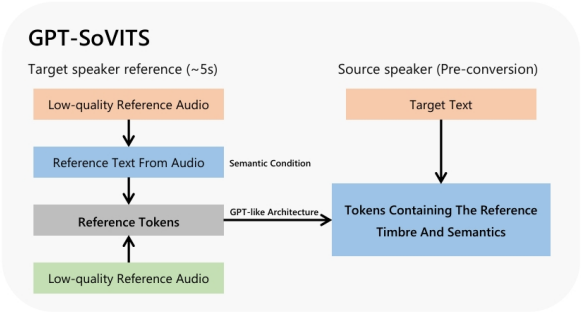

Figure 2: The workflow of GPT-SoVITS

Figure 2 demonstrates the GPT-SoVITS model for TTS utilizes a dual-input approach. A low-quality reference audio clip (around 5 seconds) is used to extract the target speaker's timbre through ASR or manual transcription, generating reference text and tokens. Simultaneously, source speaker input, such as target text, is combined with reference tokens containing timbre and semantic information via a GPT-like architecture. This process outputs speech that matches the target speaker's characteristics while preserving the semantic content of the input.

Proposed Solution

Preprocessing Models with the Initial Dataset

To enhance the quality and generalization ability of the model, multiple preprocessing strategies are employed on the same dataset.

Denoising Models To improve the quality of the speech data and ensure that the model learns from clean, noise-free inputs, several denoising strategies are employed during the preprocessing phase.UVR5 is used as the initial step for preprocessing the audio materials. It enhances the quality by

effectively removing various types of unwanted noise and artifacts, preparing the speech signals for further processing.

To handle reverberation, the MDX-Net model is employed. It is particularly effective for dual-channel reverberation removal, offering superior performance in eliminating room echoes and environmental reverberation from multi-microphone recordings. However, it is worth noting that MDX-Net is not suitable for single-channel reverberation removal.

Two models DeEcho-Aggressive and DeEcho-DeReverb are utilized to remove echo.The DeEcho-Aggressive model is designed to aggressively suppress echoes, making it ideal for environments with significant echo. On the other hand, DeEcho-DeReverb performs additional reverberation removal, capable of handling single-channel echoes effectively. While DeEcho-DeReverb excels at eliminating single-channel reverberation, it struggles to fully clean reverberation caused by high-frequency plate-like reflections, commonly present in challenging acoustic conditions.

Slice Selection Strategy In order to enhance the quality and relevance of the training data, we applied two distinct slicing strategies to the initial data. The first strategy involves directly slicing the denoised audio data without further processing. This method ensures a straightforward approach to segmenting the audio into manageable pieces, allowing for easy use in model training.

The second strategy, however, takes a more refined approach. After denoising the audio, we perform an additional filtering and selection process to retain only the highest quality segments. This step involves evaluating the audio slices for clarity and relevance, and approximately 20% of the original dataset is kept for training(Chen et al. 2019). This filtered subset represents the most informative and noise-free portions of the audio, ensuring that the model is trained on high-quality data, which is expected to improve performance and reduce the impact of less useful or noisy samples.

Additionally, all other slicing parameters, such as the threshold, minimum length, minimum interval, hop size, and maximum mute retention, remain consistent across both strategies. This ensures that the only difference between the two approaches is the quality control applied in the second strategy. By maintaining uniformity in all other parameters, we can attribute any improvements in model performance directly to the quality of the selected slices rather than differences in preprocessing configurations.

Data Augmentation with the Same Processing Method For data augmentation, we prepared six initial speech segments with durations of 10, 20, 30, 40, 50, and 60 minutes. Each segment was constructed by incrementally adding 10 minutes of new raw audio to the preceding one, ensuring that all samples are based on a continuously expanding dataset(Lakhotia et al. 2021). With different slicing strategies applied to these segments, a total of 12 distinct datasets were generated for comparative analysis. This approach allows for evaluating the impact of varying data sizes and slicing methods on model performance(Chung et al. 2019), while maintaining consistent processing techniques across all datasets.

Hyperparameter Tuning

In the hyperparameter tuning process, different configurations were explored to facilitate comparison between models. For the SoVITS model, the learning rate weight for the text module was tested to assess the impact of the parameter on performance. Similarly, for the GPT model, adjustments were made to the total training duration and the activation of DPO(Direct Preference Optimization) training, in order to evaluate their effects on the model's output. These hyperparameter adjustments were made to compare the models' effectiveness under different settings, providing insights into the optimal configurations for each approach.

Fusion Strategy

The various strategies outlined above lead to the creation of several distinct SoVITS and GPT models. The ultimate output of the inference process is generated through the fusion of these models. The purpose of exploring different fusion strategies is to investigate how the combination of these models can enhance overall performance. By experimenting with various fusion methods, we aim to identify the optimal approach that maximizes the quality and effectiveness of the final model, ensuring improved synthesis results.

Model Evaluation

We evaluate all the different models generated in the final phase. Each model generates X sentences, with each sentence having Y variations, resulting in a total of $X \times Y$ speech samples. These samples are randomly shuffled and scored. Let $s_{i,j}$ denote the score for the j-th variation of the i-th sentence. The evaluation score for each model is computed as follows:

$$Q = \frac{1}{X \times Y} \sum_{i=1}^{X} \sum_{j=1}^{Y} s_{i,j}$$

where Q is the overall evaluation metric, representing the model's performance, and $s_{i,j}$ are the individual scores for each sample. The average score across all generated samples is used to determine the final performance metric of the model.

After computing the scores for all models, we rank them based on their Q_k values. A higher score indicates better performance, encompassing clarity, completeness, and voice similarity(Adigwe et al. 2018). The models are sorted in descending order, and the rank of the k-th model is determined as follows:

$$\text{Rank}(k) = \text{argsort}\left(Q_1, Q_2, \ldots, Q_K\right)$$

where Q_k is the evaluation score of the k-th model, and the models are ranked in decreasing order of their scores. The model with the highest Q_k will have rank 1, and the model with the lowest Q_k will have rank K.

Process Design

The overall Proposed Solution process design is shown in Figure 3.

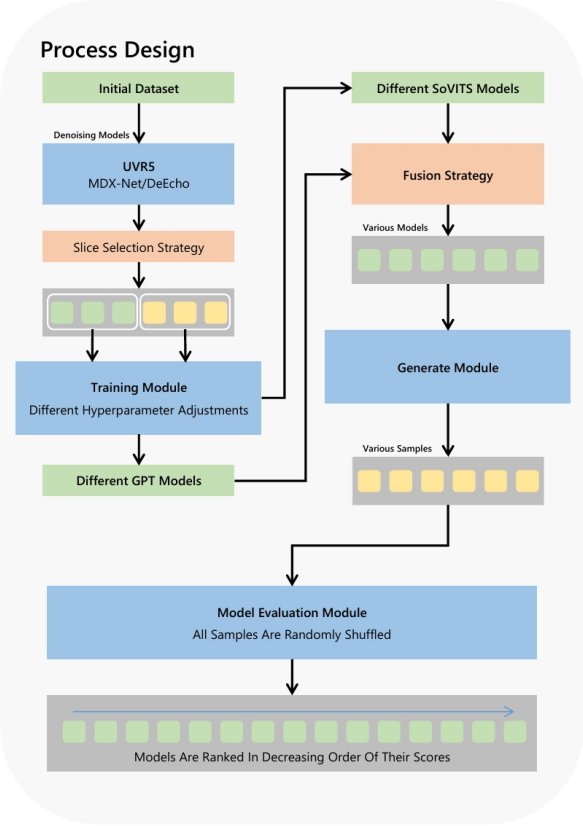

Figure 3: Proposed solution process design

Experiments

In real-world scenarios, the collected raw voice data is often imperfect and contains various imperfections. To ensure a fair comparison in our experiments, we selected voice materials that included noise, distortion, echoes, and fluctuations in volume. This approach ensures that the experiment reflects more realistic conditions and prevents high-quality data from skewing the comparative evaluation of the models.

In this section, we present the results of our experiments designed to evaluate the effectiveness of the GPT-SoVITS model under various configurations and preprocessing methods.

Experiment Setup

To increase the contrast in our experiments, we selected six initial speech recordings (01 to 06) and combined them into two datasets: **01-02** and **01-06**, with durations of 20 minutes and 60 minutes, respectively. These two datasets were processed using two slicing strategies: **Direct Slicing** and **Selective Slicing**, where only approximately 20% of the original length was retained in the selective slicing strategy. This resulted in four sliced datasets:

- **01-02 (20min) Direct Slicing**
- **01-02 (20min) Selective Slicing**
- **01-06 (60min) Direct Slicing**
- **01-06 (60min) Selective Slicing**

The SoVITS model training phase was conducted for 8 epochs, with the text module learning rate weights set to 0.4 and 0.28, resulting in 8 different SoVITS model parameter files.

For the GPT model training phase, a total of 15 epochs were performed with DPO(Direct Preference Optimization) either enabled or disabled, generating 8 GPT model parameter files.

Experimental results and analysis

In the fusion phase, the SoVITS and GPT models were combined. For each model combination, 5 sentences were generated, and each sentence was generated twice, yielding a total of 640 speech samples.These samples were scored on a scale of 60-100, in integer values.The scores should follow a normal distribution centered around 80 to better compare experimental results.

The final evaluation of these 64 models involved averaging the 640 scores for each model, resulting in a quality score (Q value) for each model. The models were then ranked based on their Q values.

The experimental results and analysis are as follows:

- The SoVITS model trained on **01-02** (20min) using the selective slicing strategy and combined with the GPT model trained on **01-06** (60min) using the selective slicing strategy with both DPO disabled, received the highest scores.
- The SoVITS model trained on **01-06** (60min) using the direct slicing strategy, combined with the GPT model trained on **01-02** (20min) using the direct slicing strategy, received the lowest scores.
- For other combinations, models trained with the selective slicing strategy generally received higher scores than those trained with the direct slicing strategy. Even though only about 20% of the original dataset remains, selective slicing strategy effectively improves the performance of the model, whether in training the SoVITS model or the GPT model.
- When the training dataset was **01-02** (20min), enabling DPO training significantly improved the model performance regardless of the slicing strategy. However, when trained on **01-06** (60min), DPO training introduced more noise in the model outputs, indicating that DPO training improved scores for models trained on shorter datasets but performed poorly on longer datasets.
- The text learning rate weight had little to no impact on the overall model scores.
- Increasing the dataset length during SoVITS model training did not improve model performance. In fact, as the dataset length increased, the model learned more noise features from the dataset, leading to worse output results.
- Increasing the dataset length during GPT model training effectively improved model performance. GPT models

trained on longer datasets performed better on these sentences, reducing issues like missing or unclear speech in the output.

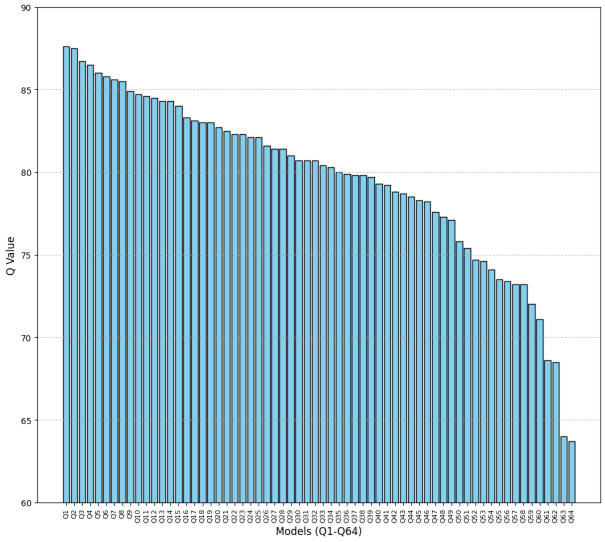

Figure 4: Average Q Values for 64 Combinations

Figure 4 displays the bar chart of the average Q values for each model (Q1-Q64), ranked from highest to lowest. The chart highlights the performance variation across models, with the y-axis limited to a range of 60 to 90 for clarity.

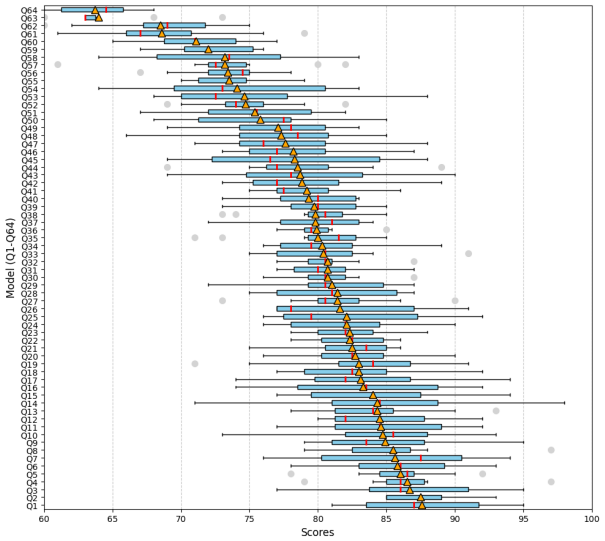

Figure 5: Box Plot for 64 Model Combinations

Figure 5 illustrates the distribution of scores for each model combination in the form of a box plot. Each box plot represents the score distribution of a specific model, showing the median (red line), mean score(orange triangle), quartiles, and potential outliers (light grey points). The scores are displayed on the x-axis, while the model labels (Q1-Q64) are shown on the y-axis. The x-axis range is set from 60 to 100 to focus on the score range of interest, and the box plots are designed with a sky blue fill and black borders. Outliers are defined as data points that fall outside the range:

$$\text{range}(k) = [S_k - 1.5 \times IQR_k, T_k + 1.5 \times IQR_k]$$

where S is the first quartile (lower quartile), T is the third quartile (upper quartile), and $IQR = T - S$ is the interquartile range. These outliers represent data points that are significantly different from the main distribution of the scores, indicating either exceptionally high or low performance in certain model combinations.

Conclusion

In this study, we introduced the GPT-SoVITS model, which addresses key challenges in text to speech synthesis and voice cloning, particularly in few shot learning scenarios. Through a series of experiments, we demonstrated that refining the model architecture and improving the preprocessing strategies can mitigate some of these issues. The results highlight the importance of the quality of the original dataset. Discarding or modifying some of the low-quality or invalid content not only reduces unnecessary training time but also significantly improves model performance and demonstrate that GPT-SoVITS can be a valuable tool for personalized voice cloning in challenging conditions. Thereby demonstrating that GPT-SoVITS can be a valuable tool for personalized voice cloning in challenging conditions. Further analysis revealed that DPO training led to improved outcomes for models trained on smaller datasets. These insights could facilitate the synthesis of longer sentences via the GPT-SoVITS model.

References

Adigwe, A.; Tits, N.; Haddad, K. E.; Ostadabbas, S.; and Dutoit, T. 2018. The Emotional Voices Database: Towards Controlling the Emotion Dimension in Voice Generation Systems. arXiv:1806.09514.

Arik, S.; Chen, J.; Peng, K.; Ping, W.; and Zhou, Y. 2018. Neural Voice Cloning with a Few Samples. In Bengio, S.; Wallach, H.; Larochelle, H.; Grauman, K.; Cesa-Bianchi, N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Chen, M.; Tan, X.; Li, B.; Liu, Y.; Qin, T.; Zhao, S.; and Liu, T.-Y. 2021. AdaSpeech: Adaptive Text to Speech for Custom Voice. arXiv:2103.00993.

Chen, Y.; Assael, Y.; Shillingford, B.; Budden, D.; Reed, S.; Zen, H.; Wang, Q.; Cobo, L. C.; Trask, A.; Laurie, B.; Gulcehre, C.; van den Oord, A.; Vinyals, O.; and de Freitas, N. 2019. Sample Efficient Adaptive Text-to-Speech. arXiv:1809.10460.

Chung, Y.-A.; Wang, Y.; Hsu, W.-N.; Zhang, Y.; and Skerry-Ryan, R. 2019. Semi-supervised Training for Improving Data Efficiency in End-to-end Speech Synthesis. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 6940–6944.

Hunt, A.; and Black, A. 1996. Unit selection in a concatenative speech synthesis system using a large speech database. In *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, volume 1, 373–376 vol. 1.

Jia, Y.; Zhang, Y.; Weiss, R. J.; Wang, Q.; Shen, J.; Ren, F.; Chen, Z.; Nguyen, P.; Pang, R.; Moreno, I. L.; and Wu, Y. 2019. Transfer Learning from Speaker Verification to Multispeaker Text-To-Speech Synthesis. arXiv:1806.04558.

Kong, J.; Kim, J.; and Bae, J. 2020. HiFi-GAN: Generative Adversarial Networks for Efficient and High Fidelity Speech Synthesis. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 17022–17033. Curran Associates, Inc.

Lakhotia, K.; Kharitonov, E.; Hsu, W.-N.; Adi, Y.; Polyak, A.; Bolte, B.; Nguyen, T.-A.; Copet, J.; Baevski, A.; Mohamed, A.; and Dupoux, E. 2021. On Generative Spoken Language Modeling from Raw Audio. *Transactions of the Association for Computational Linguistics*, 9: 1336–1354.

Shen, J.; Pang, R.; Weiss, R. J.; Schuster, M.; Jaitly, N.; Yang, Z.; Chen, Z.; Zhang, Y.; Wang, Y.; Skerry-Ryan, R.; Saurous, R. A.; Agiomyrgiannakis, Y.; and Wu, Y. 2018. Natural TTS Synthesis by Conditioning WaveNet on Mel Spectrogram Predictions. arXiv:1712.05884.

Tan, X.; Qin, T.; Soong, F.; and Liu, T.-Y. 2021. A Survey on Neural Speech Synthesis. arXiv:2106.15561.

van den Oord, A.; Dieleman, S.; Zen, H.; Simonyan, K.; Vinyals, O.; Graves, A.; Kalchbrenner, N.; Senior, A.; and Kavukcuoglu, K. 2016. WaveNet: A Generative Model for Raw Audio. arXiv:1609.03499.

Wang, C.; Chen, S.; Wu, Y.; Zhang, Z.; Zhou, L.; Liu, S.; Chen, Z.; Liu, Y.; Wang, H.; Li, J.; He, L.; Zhao, S.; and Wei, F. 2023. Neural Codec Language Models are Zero-Shot Text to Speech Synthesizers. arXiv:2301.02111.

Wang, Y.; Skerry-Ryan, R.; Stanton, D.; Wu, Y.; Weiss, R. J.; Jaitly, N.; Yang, Z.; Xiao, Y.; Chen, Z.; Bengio, S.; Le, Q.; Agiomyrgiannakis, Y.; Clark, R.; and Saurous, R. A. 2017. Tacotron: Towards End-to-End Speech Synthesis. arXiv:1703.10135.

Yan, Y.; Tan, X.; Li, B.; Zhang, G.; Qin, T.; Zhao, S.; Shen, Y.; Zhang, W.-Q.; and Liu, T.-Y. 2021. AdaSpeech 3: Adaptive Text to Speech for Spontaneous Style. arXiv:2107.02530.

Zen, H.; Tokuda, K.; and Black, A. W. 2009. Statistical parametric speech synthesis. *Speech Communication*, 51(11): 1039–1064.