AdaptVITS: A Parameter-Efficient Fine-Tuning Approach for Limited Target Speaker Data in End-to-End Voice Cloning

by

Adil Ahmed Chowdhury

A dissertation submitted to the
Department of Computer Science
in conformity with the requirements for
the degree of Master of Science

Bishop's University
Canada
July 2026

Abstract

End-to-end voice cloning models like VITS require large amounts of data to produce high-quality speech. This makes it difficult to adapt models when target speaker data is highly limited. To address this problem, this thesis introduces AdaptVITS, a parameter-efficient fine-tuning approach optimized for scenarios with limited target speaker data. The strategy freezes the pre-trained text and posterior encoders to keep the internal language alignments stable. Meanwhile, the downstream decoder and other synthesis modules remain active. This allows the active 82.50% of the parameter capacity to focus entirely on capturing the unique vocal traits of the target speaker. Empirical evaluations across independent 5-hour training subsets show that AdaptVITS closely matches the quality of full fine-tuning. The model achieved a competitive cross-run mean of 3.87 MOS, 8.59% CER, 15.65% WER, and 0.5188 SECS. It completely outperforms training from scratch on the same 5-hour dataset, which suffers a total structural collapse. Furthermore, because this selective optimization happens only during the backward training pass, the final model layout remains unchanged. This introduces zero runtime latency or extra storage overhead during deployment. These results position AdaptVITS as a simple, stable, and resource-efficient solution for voice cloning.

Acknowledgments

I want to truly thank my parents, my wife, and my family. Their constant support, encouragement, and sacrifices made this entire journey possible.

I am deeply grateful to my supervisors, Dr. Madjid Allili and Dr. Mohammed Ayoub Alaoui Mhamdi, for their constant guidance, helpful feedback, and support throughout my research. I also want to thank the chair of my examining committee, Dr. Rachid Hedjam, and the committee members, Dr. Russell Butler and Dr. Yasir Malik. Thank you for carefully reviewing my work and sharing great suggestions that made this thesis much better and clearer. Special thanks to our graduate coordinator, Dr. Stefan Bruda, whose administrative help kept everything running smoothly. I also want to thank my undergraduate mentor, Dr. Arif Ahmad, who helped me understand this subject deeply and inspired me to pursue this work in the first place.

Finally, I am grateful to Bishop's University for believing in my potential and choosing me as one of the 28 international students to receive the International Tuition Waiver for the 2023–2024 academic year.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Introduction

In today's digital world, speech-based interaction is a foundational component of human-computer communication. Natural language interfaces, such as virtual assistants and personalized audiobooks, are more important than ever. These systems rely on high-quality speech synthesis to bridge the gap between machines and humans. However, if these systems lack naturalness or speaker identity, user engagement and accessibility suffer. This makes the accurate reproduction of human voices crucial for the next generation of interactive technologies.

The demand for high-fidelity reproduction has led to the emergence of voice cloning. This specialized domain of speech synthesis aims to replicate the unique vocal identity of a specific target speaker. Effective voice cloning is essential for personalized AI and medical rehabilitation [19]. By recreating a specific voice, organizations can provide assistive tools for individuals with speech impairments. Accurate cloning also maintains user trust in large-scale communication platforms and reduces the costs of professional voice recording. Achieving high-fidelity results ensures more human-centric operations in modern digital services.

Historically, modular architectures were the most reliable way to perform voice cloning. These systems use a multi-step configuration that separates the acoustic model from the neural vocoder [11]. While effective, managing these separate stages is complex. As deep learning evolved, the field shifted toward integrated end-to-end (E2E) architectures. These models simplify the pipeline by generating audio directly from text in a single process.

State-of-the-art E2E architectures, specifically VITS, unify text-to-waveform mapping into a single framework. VITS utilizes variational autoencoders, normalizing flows, and adversarial training to produce high-quality speech [14]. This design eliminates the need for intermediate feature extraction. Consequently, it yields higher naturalness than older modular pipelines. Like other deep learning models, VITS learns to replicate human speech by processing thousands of acoustic patterns.

When trained properly, the model output closely matches the natural prosody and timbre of a human speaker. However, a major challenge arises with this model when the available target data is limited. Successfully adapting the architecture to a specific individual remains the primary hurdle for voice cloning applications.

1.2 Problem Statement

Modern end-to-end (E2E) voice cloning faces a critical challenge in resource efficiency. Models like VITS provide high-quality speech but are inherently complex and data-hungry. The standard solution for adapting these models to a new speaker is full fine-tuning. This process involves updating every single parameter in the architecture.

Updating every parameter for every new speaker creates significant operational bottlenecks. First, full fine-tuning requires high Peak GPU memory. This limits the ability to train models on standard or consumer-grade hardware. Second, many components within an E2E system are speaker-independent. These modules learn the universal rules of language and phonetics during high-resource pretraining on large datasets. Forcing the model to re-learn these universal traits for every new speaker can lead to inefficient resource utilization and may yield diminishing returns in synthesis quality.

Furthermore, previous attempts to solve this efficiency gap have focused on changing the model's architecture to make it smaller, such as AdaVITS [26]. Others have introduced complex unlabeled data frameworks [15]. There is a lack of research into how the existing VITS architecture can be optimized simply by freezing its universal components. This research addresses this gap by developing a deep-learning-based, parameter-efficient approach. We achieve quality synthesis without modifying the core architecture.

We propose a method that keeps the original VITS architecture unchanged. We fine-tune the model while freezing speaker-independent modules. This strategy relies on the pretrained knowledge of these modules while training only the speaker-dependent ones. The text and posterior encoders are the primary modules we freeze. These encoders generalize well because they handle the universal rules of language, such as phonetics and word structure. Since these rules do not change between speakers, the encoders do not need to be fine-tuned.

In addition, this selective freezing improves stability and convergence when target data is limited. Selective freezing is designed to help the model maintain stable phonetic and syntactic representations while the remaining modules learn the new speaker's unique sound. To verify this, we conducted an ablation study to identify the specific impact of speaker-dependent and speaker-independent modules. We evaluate our results on subjective metric, such as MOS, and objective metrics, including WER, CER, and SECS, using fixed unseen utterances.

1.3 Research Objectives and Contributions

The goal of this thesis is to develop a more efficient way to adapt end-to-end voice cloning models to new speakers. Instead of changing the model architecture, this research focuses on a more efficient training strategy called **AdaptVITS**. Our pretrain model is trained on LibriTTS 360 [36] & fine tune on VCTK [32] dataset.

1.3.1 Research Objectives

The primary objective of this research is to develop a parameter-efficient adaptation framework that maintains high-fidelity voice cloning without modifying the original VITS architecture, while reducing its computational and storage footprint. Second, the study seeks to validate the consistency of this strategy by demonstrating robust performance across varying target speakers and randomized dataset subsets. Third, the work aims to explain exactly how different parts of the model work together to remain stable and accurately capture a new person's voice.

1.3.2 Thesis Contributions

This research makes several key contributions to the field of speech synthesis:

- **C1 —** A fine-tuning method that reduces trainable parameters by **17.5%**, translating to a theoretical saving of approximately **115 MB** of allocated gradient memory storage.
- **C2 —** An empirical reduction in **observed peak physical GPU memory (VRAM)** required during training from 5974 MB to as low as **5201 MB**
- **C3 —** An improvement in the **Mean Real-Time Factor (RTF)** from 0.1275 to **0.0631**, evaluated within an identical deployment architecture and driven by specific software and compilation optimizations rather than structural changes to the inference graph.
- **C4 —** Confirmation via **WER**, **CER**, **SECS**, and **t-SNE** visualizations that frozen encoders maintain high-quality speech and speaker identity.
- **C5 —** A technical justification for keeping the discriminator trainable to ensure stable adversarial training.

Furthermore, our approach is structurally the opposite of Kim et al. [15]. Kim et al. freeze the posterior encoder and the waveform decoder, training only the conditional normalizing flow. In contrast, AdaptVITS freezes the text and posterior encoders, leaving the decoder, normalizing flow, duration predictor, speaker embedding, and discriminator fully active and trainable.

This choice is based on how these components naturally work. The upstream encoders learn general language structures and phonetic rules, which stay the same from speaker to speaker. However, the downstream decoder is what actually shapes the unique voice identity and physical timbre of an individual. By locking the speaker-invariant language modules and training the speaker-dependent voice modules, AdaptVITS can efficiently capture a new speaker's unique vocal traits using very little data.

1.4 Thesis Outline

This thesis is organized as follows:

Chapter 2 reviews the technological evolution of text-to-speech synthesis and voice cloning. It traces the shift from traditional statistical models to modern deep learning architectures and end-to-end networks, concluding with a comparative analysis of these different systems.

Chapter 3 outlines the research methodology, detailing the VITS architecture and the proposed AdaptVITS framework. It describes the experimental setup, including the construction of randomized dataset subsets and the evaluation procedures.

Chapter 4 presents the experimental findings and evaluates performance across various training strategies. It provides a rigorous analysis using objective and subjective metrics to interpret significant results.

Chapter 5 summarizes the research contributions, discuss the limitations of the study and outlining directions for future work.

Chapter 2

Preliminaries and Literature Review

This chapter reviews the technological evolution of text-to-speech (TTS) synthesis and voice cloning. It begins with an overview of traditional paradigms, tracing the shift from early rule-based systems to concatenative and statistical parametric models. The chapter then examines deep learning architectures, covering multi-stage pipelines and parallel, non-autoregressive feature generation systems. Recent breakthroughs in integrated end-to-end networks and speech language modeling frameworks are also evaluated. Finally, two comprehensive matrix tables provide a comparative analysis of these historical and modern systems.

2.1 The Evolution of Traditional Speech Synthesis Paradigms

Voice cloning is an advanced branch of text-to-speech (TTS) synthesis designed to replicate a specific target speaker's unique voice profile, linguistic style, and natural expressions. Modern voice cloning frameworks leverage deep neural networks to extract high-dimensional speech features such as fundamental frequency (F0), spectral characteristics, and rhythm from a limited dataset of target audio. Historically, the field has evolved through a series of major technology shifts. Each new paradigm arose directly to solve the mathematical or computational bottlenecks of the system that came before it.

Rule-Based Systems: During the 1980s, rule-based parametric synthesis was the dominant approach in speech synthesis, relying on hand-crafted linguistic and acoustic rules to convert text into speech sounds [16]. Speech attributes such as segment duration, intonation contours, and emphasis were explicitly defined using

fixed mathematical formulas. Although these systems required very little computing power, they were fundamentally limited because human speech contains natural, chaotic variations that are too complex to program manually by hand. The rigid, deterministic nature of these hard-coded rules resulted in a robotic, mechanical sound. This lack of naturalness motivated researchers to abandon hand-crafted synthesis in favor of data-driven methods that could learn speech patterns directly from human recordings.

Concatenative Synthesis: To overcome the robotic sound of rule-based frameworks, concatenative synthesis became the preferred methodology in the 1990s [10]. This approach bypassed acoustic modeling entirely by using a large database of recorded human speech. The audio was cut into small fragments, such as phones or diphones, and a unit selection algorithm stitched these pieces together at runtime to match the target text based on the linguistic context. While concatenative synthesis produced highly natural speech segments, its overall realism depended entirely on the size and variety of the recorded database. Changing voice characteristics or adapting the system to an unseen speaker required recording an entirely new, exhaustive audio corpus. Furthermore, small differences in pitch or phase at the clip boundaries frequently introduced audible glitches or distortion. This absolute reliance on massive, unchangeable databases drove researchers to develop statistical frameworks that could compress speech patterns into a flexible parametric space.

Statistical Parametric Synthesis: In the early 2000s, statistical parametric synthesis based on Hidden Markov Models (HMMs) emerged to resolve the database scaling issues of concatenative systems [30]. Instead of storing raw audio waveforms, these models captured speech characteristics via mathematical parameters, representing features like spectral envelopes and pitch trajectories as continuous probability distributions. Although statistical parametric systems drastically reduced the storage footprint and allowed for basic speaker adaptation using linear mathematical transformations, they hit an unyielding quality ceiling. The averaging properties built into HMMs caused severe over-smoothing of the speech details. This resulted in a muffled, flattened timbre that lacked the crisp clarity and dynamic energy of organic speech. This persistent limitation established a clear need for nonlinear, deep generative models capable of capturing fine-grained wave structures directly.

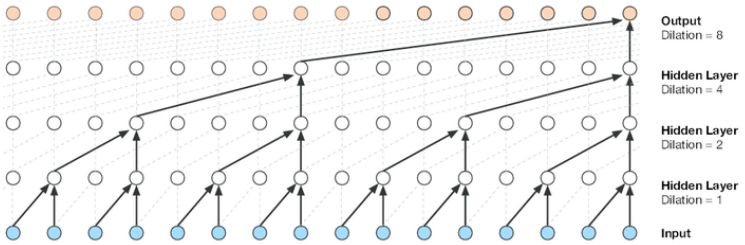


Figure 2.1: Visualization of dilated causal convolutional layers expanding the model's receptive field while maintaining temporal order.

2.2 Neural Frameworks and Decoupled Multi-Stage Pipelines

The integration of deep neural networks fundamentally transformed speech synthesis by replacing shallow linear models with complex, non-linear mappings. These architectures made it possible to reconstruct high-dimensional audio distributions directly from data instead of relying on manually designed rules.

The early application of deep neural networks to speech synthesis focused heavily on generating high-fidelity raw audio waveforms at the sample level. A foundational breakthrough in this domain was WaveNet, an autoregressive generative model developed by Oord et al. [31] that was inspired by the PixelCNN architecture. This framework factorizes the joint probability of a waveform $x = \{x_1, \ldots, x_T\}$ into a conditional product sequence:

$$p(x) = \prod_{t=1}^{T} p(x_t \mid x_1, \ldots, x_{t-1}) \tag{2.1}$$

By conditioning each audio sample x_t on all previous timesteps, the model successfully captured intricate micro-prosodic details and high-frequency spectral components, achieving an unprecedented leap in speech naturalness.

To maintain strict temporal order during generation, the architecture relies on causal connections that prevent the model from accessing future context. However, tracking long-term speech dependencies over thousands of audio samples would normally require an excessively deep network layer stack. To resolve this structural bottleneck, the architecture incorporates dilated causal convolutions, as illustrated in Figure 2.1. By skipping input channels at exponentially increasing rates (doubling the dilation factor $d = 1, 2, 4, 8 \ldots$ at each successive layer), the network expands its receptive field without increasing filter sizes. This enables the system to capture broad linguistic and prosodic rhythms while keeping the total parameter count low.

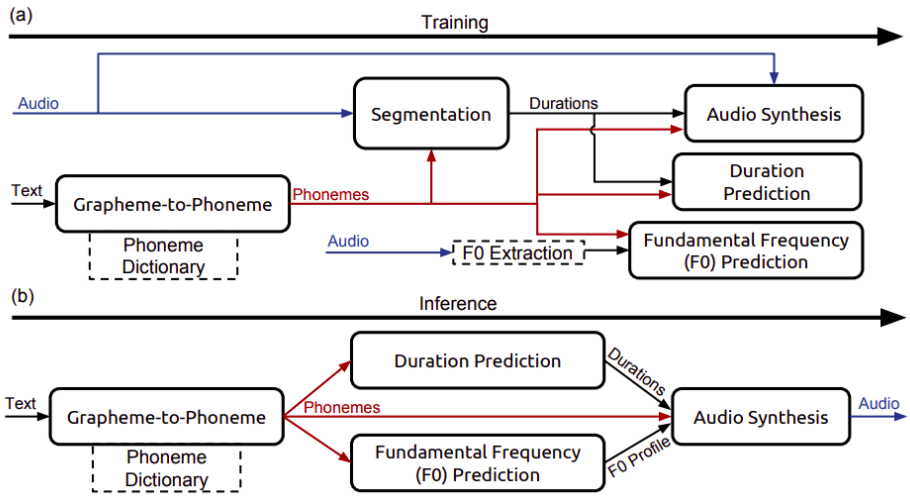


Figure 2.2: DeepVoice System diagram depicting the clear separation between the acoustic prediction and inference paths.

Although raw waveform modeling resolved the over-smoothed, muffled acoustic limitations of traditional statistical systems, it introduced a severe computational bottleneck during synthesis. Because the generation process is fundamentally autoregressive, predicting each individual audio sample requires a sequential forward pass through the entire network structure. At standard high-fidelity sampling rates, generating just a single second of audio demands tens of thousands of consecutive model evaluations, resulting in extreme processing delays that prohibited real-time inference deployment. Furthermore, these early frameworks lacked a unified mechanism to process raw text directly, requiring pre-conditioned linguistic features as input. These limitations prompted the design of composite systems such as DeepVoice 1, which aimed to achieve real-time operations by dividing the text-to-waveform task into an optimized neural pipeline consisting of five decoupled operations [2]. These blocks included a grapheme-to-phoneme converter, a segmentation model trained with Connectionist Temporal Classification (CTC) loss to locate phoneme boundaries [1], a duration predictor, a fundamental frequency (F0) contour model, and a conditioned waveform vocoder.

As shown in Figure 2.2, this design explicitly isolated the acoustic prediction path from the waveform generation path. However, this extreme modularity introduced compounding errors across the pipeline segments, where small mismatches in early modules degraded the performance of downstream components. DeepVoice 2 attempted to mitigate these pipeline errors and introduce multi-speaker functionality by separating the joint duration-frequency modules into distinct sequential blocks [8]. This model conditioned the components on low-dimensional speaker embeddings and introduced batch normalization (BN) alongside residual

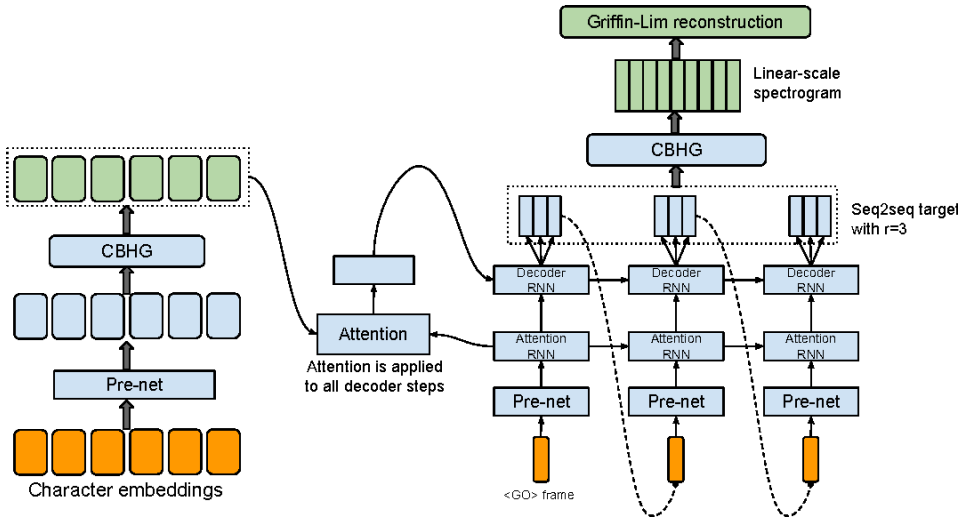


Figure 2.3: Tacotron model architecture showing the encoder, attention mechanism, and post-processing network.

paths to stabilize gradient flow across the convolutional layers. Formally, where the first iteration computed standard non-linear transformations:

$$h^{(l)} = \text{relu}\left(W^{(l)} * h^{(l-1)} + b^{(l)}\right) \tag{2.2}$$

The second iteration integrated residual shortcuts and feature normalization layers:

$$h^{(l)} = \text{relu}\left(h^{(l-1)} + \text{BN}\left(W^{(l)} * h^{(l-1)}\right)\right) \tag{2.3}$$

Despite these architectural updates, the system remained constrained by its multi-step pipeline topology.

The complex engineering overhead and error propagation of these multi-component pipelines eventually led to the development of integrated sequence-to-sequence (seq2seq) frameworks. Wang et al. [34] introduced Tacotron - a unified network that consolidated the feature engineering process by mapping character sequences directly onto intermediate mel-spectrogram representations in a single forward pass, utilizing a specialized Convolution Bank-Highway Network-GRU (CBHG) encoder and an attention-based decoder [28].

As outlined in Figure 2.3, this framework generated mel-spectrograms that still required a lossy linear inversion phase (Griffin-Lim) to produce actual audio waveforms. This final vocoding dependency was resolved by Tacotron 2, which replaced the complex CBHG blocks with vanilla LSTM layers and paired the seq2seq acoustic decoder directly with a mel-conditioned WaveNet vocoder [25]. This combined architecture achieved an elite naturalness score of MOS: 4.53 ± 0.07. However, despite

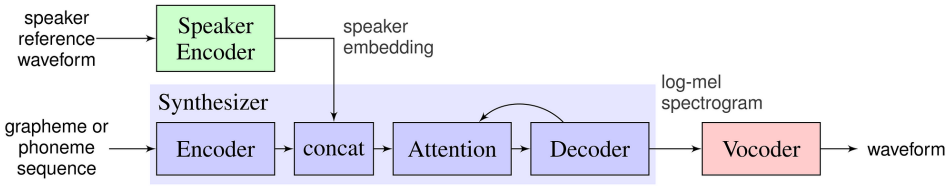


Figure 2.4: Model overview of the SV2TTS transfer learning framework for zero-shot adaptation.

this high acoustic quality, Tacotron 2 retained a strict multi-stage dependency where the acoustic model and the vocoder were optimized completely independently. Because the model generated intermediate mel-spectrograms as an abstraction layer, vital phase information was discarded. Any prediction error or alignment drift in the mel-spectrogram phase caused high error propagation during vocoding, frequently resulting in word omissions, regressions, and acoustic instability on long or complex input text strings.

Parallel to the development of single-speaker end-to-end acoustic architectures, speech research advanced toward scaling deep neural frameworks to handle multi-speaker environments and zero-shot voice cloning. VoiceLoop approached this challenge by employing a shifting buffer working memory to synthesize speech from unaligned, unstructured target audio collected in natural environments [29]. However, its reliance on a localized shifting loop limited its ability to clone highly dynamic, unseen speaker timbres cleanly. To resolve this and establish a generalized transfer learning model for voice cloning, the SV2TTS framework introduced a decoupled three-stage pipeline designed to isolate speaker identity extraction from acoustic generation [11].

As structured in Figure 2.4, the SV2TTS architecture splits the cloning task across three independent elements: a speaker encoder trained on a speaker-verification task to map short reference audio clips to a fixed d-vector embedding; a Tacotron 2 synthesizer that outputs a mel-spectrogram conditioned on that embedding; and an autoregressive WaveNet vocoder. This decoupled approach demonstrated impressive generalization to unseen target speakers, achieving a similarity score of SMOS: 4.22 ± 0.06 on the VCTK dataset and SMOS: 3.28 ± 0.08 on LibriSpeech [11].

Nevertheless, this multi-stage decoupling introduced an architectural ceiling. Because the speaker encoder was trained on a verification objective rather than a synthesis objective, the extracted d-vector captured broad identity features but dropped fine-grained prosodic and micro-intonational nuances. Furthermore, because the model still relied on the intermediate mel-spectrogram boundary, it inherited the same error propagation and separate vocoder optimization vulnerabilities found in Tacotron 2, preventing seamless, high-fidelity voice cloning.

The computational demands and high inference latency of the recurrent LSTM layers within these seq2seq frameworks subsequently motivated the development

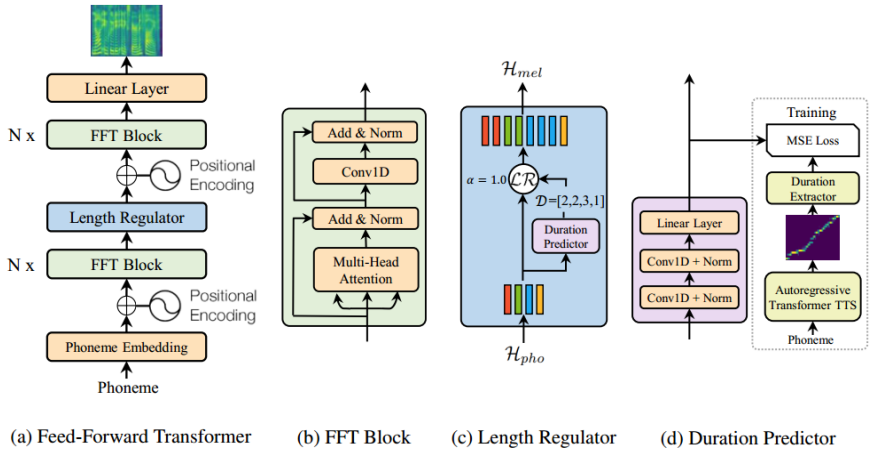


Figure 2.5: The overall feed-forward architecture of FastSpeech featuring the Length Regulator.

of fully convolutional alternatives. Ping et al. [21] replaced recurrent paths entirely in DeepVoice 3, utilizing a multi-hop convolutional attention mechanism to dramatically accelerate sequence mapping and accommodate thousands of speaker identities simultaneously. This architecture achieved training speeds up to 10 times faster than prior models. However, because it operated as a non-causal parallel model mapping to low-dimensional intermediate features before converting them into vocoder parameters, it suffered from severe alignment degradation when data constraints were introduced, making it poorly suited for low-resource speaker adaptation tasks.

2.3 Non-Autoregressive Frameworks and Parallel Feature Generation

Autoregressive sequence-to-sequence models generate speech one frame at a time, making them slow and prone to errors. If a single frame is predicted incorrectly, that error carries over to the rest of the sentence, often causing the model to skip or repeat words. To fix these latency and stability issues, researchers shifted toward parallel synthesis networks that generate entire audio sequences in a single forward pass. A major step in this direction was FastSpeech, which replaced the slow sequential decoder with a feed-forward layout built from stacked Feed-Forward Transformer (FFT) blocks [24].

As shown in Figure 2.5, the central feature of this architecture is the *Length Regulator*. Because a text sequence contains far fewer units than a speech frame sequence, the length regulator must expand the text encoder's hidden states to match the total

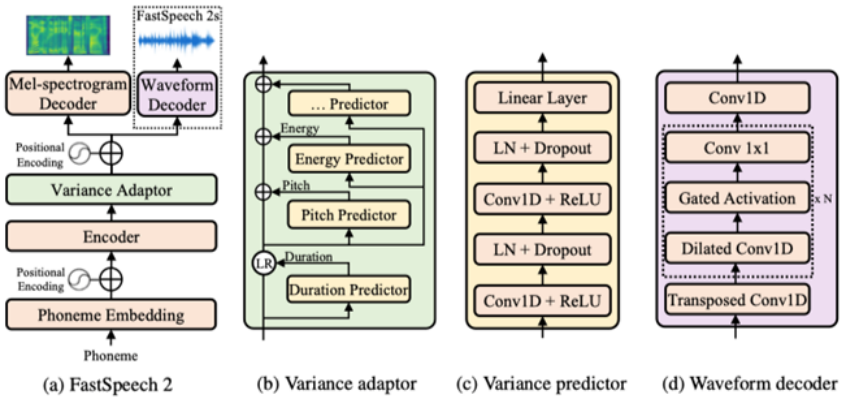


Figure 2.6: FastSpeech 2 architecture incorporating explicit variance predictors.

time length of the target audio. It does this by estimating phoneme durations, expanding the sequence along the time axis, and enabling parallel, 1D-convolutional generation of the entire mel-spectrogram at once. This change allowed the model to achieve an impressive 270x synthesis speedup over older sequential designs while significantly reducing pronunciation glitches [24].

Although these parallel networks solved the inference speed problem, the initial training style introduced a new limitation. Because the framework relied on an independent autoregressive teacher model to provide the phoneme durations, the target speech paths became over-simplified. This teacher-led distillation process smoothed out crucial acoustic details, which made the generated voices sound flat and less expressive. To remove this dependency on a teacher model, FastSpeech 2 was developed by Ren et al. [23], to train directly on real, ground-truth targets extracted straight from the original speech waveforms.

As shown in Figure 2.6, this update added a specialized *Variance Adaptor* module. Instead of guessing values or relying on a teacher model, the Variance Adaptor reads the exact, true values for phoneme duration, pitch (calculated as log-F0 values), and energy directly from the original audio recordings. These real speech features are added right into the encoder's hidden representations before the final parallel decoding steps. This direct conditioning approach successfully handled the natural variations of human speech, creating a highly stable and scalable foundation for modern parameter-efficient voice cloning research.

While explicit conditioning through a variance adaptor fixed basic timing and alignment, controlling subtle vocal expressions required a more precise feature setup. This motivated the design of architectures that could condition the parallel network on pitch contours mapped directly onto individual words or tokens, an approach implemented by FastPitch [18].

As detailed in Figure 2.7, this framework splits the modeling task between two distinct Feed-Forward Transformer stacks. The first stack operates at the input token

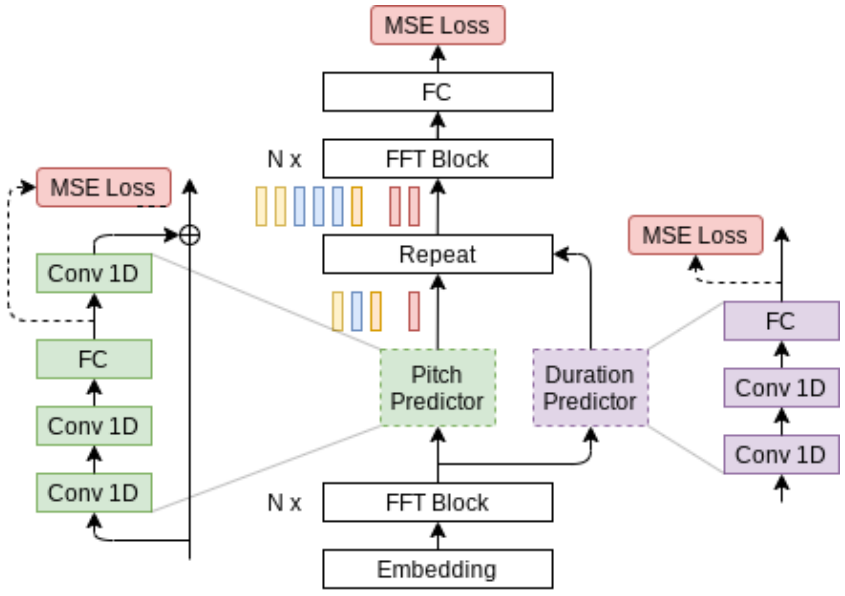


Figure 2.7: Architecture of FastPitch.

resolution level to output a text feature vector h. A bidirectional 1D-convolutional network then predicts the duration ($\hat{d}$) and the average pitch ($\hat{p}$) for each specific unit:

$$\hat{d} = \text{DurationPredictor}(h), \quad \hat{p} = \text{PitchPredictor}(h) \tag{2.4}$$

The predicted pitch value is converted into a pitch embedding and added directly to the text features, creating an augmented conditioning vector g:

$$g = h + \text{PitchEmbedding}(p) \tag{2.5}$$

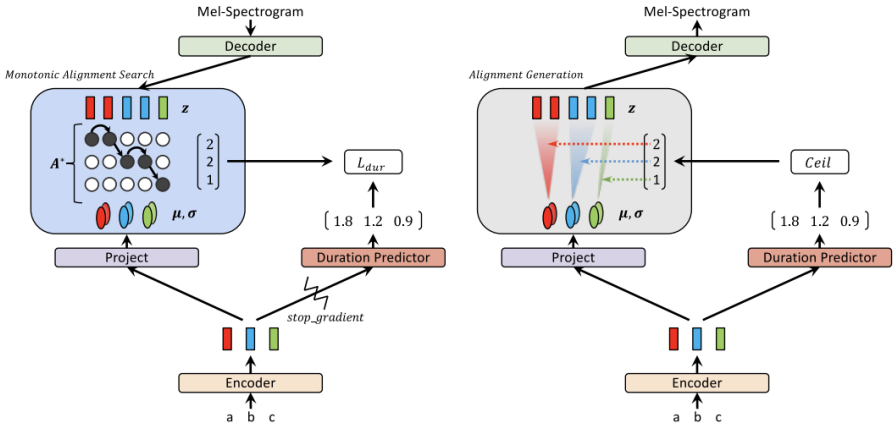
This combined sequence is upsampled based on the duration data and sent to the second frame-resolution FFT stack, which generates the final parallel mel-spectrogram sequence ($\hat{y}$):

$$\hat{y} = \text{FFTr}([g_1, \ldots, g_1, \ldots, g_n, \ldots, g_n]) \tag{2.6}$$

The model is optimized during training by minimizing a combined multi-task loss function scaled by the variables α and γ:

$$L = \|\hat{y} - y\|_2^2 + \alpha\|\hat{p} - p\|_2^2 + \gamma\|\hat{d} - d\|_2^2 \tag{2.7}$$

By linking the pitch values directly to the individual text tokens, this system achieved much higher prosodic control and voice stability than standard parallel models.



(a) An abstract diagram of the training procedure. (b) An abstract diagram of the inference procedure.

Figure 2.8: Training and inference alignments within Glow-TTS using the MAS framework.

Even with these variance upgrades, both FastSpeech and FastPitch retained a clear structural drawback: they still needed an outside alignment tool or a pre-calculated duration chart to train their variance modules. To eliminate this final external requirement, Kim et al. [13] combined parallel design with flow-based generative modeling, creating Glow-TTS.

As shown in Figure 2.8, this architecture embeds a *Monotonic Alignment Search (MAS)* engine inside a series of invertible 1x1 convolutions and affine coupling layers. During training, the flow decoder converts the target mel-spectrogram into a simplified latent space, and the MAS engine automatically finds the most accurate alignment between the text and audio by maximizing the log-likelihood of the data. During inference, a built-in duration predictor determines the phoneme lengths to guide the parallel output. By maximizing log-likelihood directly, Glow-TTS proved that fast, high-quality parallel synthesis could be achieved without any external alignment tools, providing the core mathematical framework for fully integrated end-to-end speech models.

2.4 Fully Integrated End-to-End Architectures and Parameter Adaptation Frameworks

The definitive breakthrough in modern speech synthesis came with the complete elimination of intermediate mel-spectrograms, unifying acoustic modeling and waveform generation into a single, cohesive end-to-end network. This approach avoids the loss of phase information and prevents errors from multiplying across

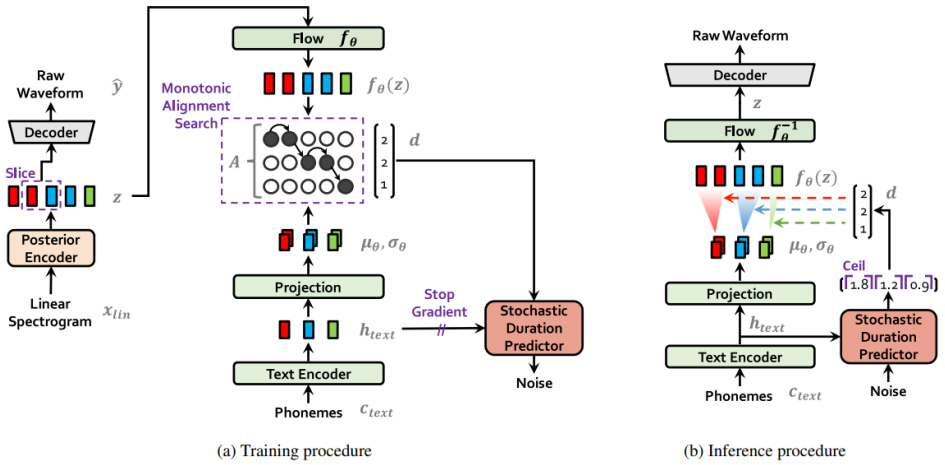


Figure 2.9: Training and inference topology of the integrated end-to-end VITS architecture.

separate models, allowing for a direct mapping from linguistic features to raw waveforms. A landmark architecture that achieved this unified design combines a Conditional Variational Autoencoder (CVAE) with normalizing flows and multi-scale adversarial training [14].

As shown in Figure 2.9, this integrated network is composed of a posterior encoder, a prior encoder (which contains a text transformer encoder and a multi-layer normalizing flow), an upsampling waveform decoder, and a multi-scale discriminator network. During training, the posterior encoder extracts a latent distribution $q_\phi(z|x_{lin})$ from the linear spectrogram (x_{lin}) of the ground-truth audio. Meanwhile, the prior encoder outputs a text-conditioned distribution $p_\theta(z|c_{text}, A_{text})$ and uses a Monotonic Alignment Search (MAS) engine to align the text with the audio. The entire network is optimized globally by maximizing the Evidence Lower Bound (ELBO), which balances reconstruction quality with a Kullback-Leibler divergence (KLD) loss:

$$L_{kl} = \log q_\phi(z|x_{lin}) - \log p_\theta(z|c_{text}, A_{text}) \tag{2.8}$$

Additionally, a stochastic duration predictor is used to capture the natural variations of human speech rhythms. By sampling phoneme lengths from random noise using a normalizing flow, the model can generate different, organic speech cadences from identical text inputs, achieving highly realistic speech naturalness.

While this combination of variational inference and normalizing flows proved highly effective for direct waveform generation, alternative parallel alignment strategies were developed around the same time. One notable approach maps text characters directly to waveforms using a differentiable token-length interpolation module inside a fully adversarial training setup [4]. To handle alignment changes when training without separate components, this system uses a soft Dynamic Time

Warping (DTW) loss across predicted spectrogram spaces. Although this fully adversarial approach reached a high naturalness score above a 4.0 MOS scale, its reliance on adversarial constraints made it highly vulnerable to training errors and mode collapse when the training dataset was limited.

To expand these integrated end-to-end architectures into multi-speaker and cross-lingual settings, subsequent research modified the internal conditioning paths to support zero-shot voice cloning. This led to a multilingual extension of the framework that replaced phoneme processing with raw character text tokens, removing the need for external pronunciation dictionaries across low-resource languages [3]. This model passes a pre-trained speaker verification embedding into all affine coupling layers, the posterior encoder, and the waveform decoder to provide global speaker identity. To keep the cloned voice close to the original speaker, a Speaker Consistency Loss (SCL) was introduced:

$$L_{SCL} = \frac{-\alpha}{n} \cdot \sum_{i=1}^{n} \cos\_sim(\phi(g_i), \phi(h_i)) \tag{2.9}$$

where $\phi(\cdot)$ extracts speaker features from the original audio (g_i) and the generated audio (h_i) to maximize their cosine similarity. Tested on unseen voices, this framework reached an excellent quality profile with a MOS of 4.21 ± 0.04 and a speaker similarity index of 0.864 SECS [3]. However, using raw character tokens instead of a phonemic mapping space introduces clear vulnerabilities. It creates an unstable linguistic layout, leading to frequent mispronunciations and timing errors within the stochastic duration predictor under low-resource testing conditions.

In contrast to these continuous latent variable architectures, a major paradigm shift occurred with the introduction of neural codec language modeling, which completely reformulates text-to-speech as a discrete sequence generation task rather than continuous signal regression. A prominent framework in this domain is VALL-E, developed by Wang et al. [33], which leverages an off-the-shelf neural audio codec encoder to quantize waveforms into discrete acoustic tokens. By scaling up the training data to a massive, 60,000-hour speech corpus derived from LibriLight, the model demonstrates remarkable emergent zero-shot voice cloning via acoustic in-context learning, matching the speaker identity, emotional state, and acoustic environment of an unseen voice using only a brief three-second reference prompt. Evaluated on unseen multi-speaker datasets, this approach achieved an objective Word Error Rate (WER) of 5.9% and a subjective similarity score of 3.81 ± 0.09 SMOS. Despite its groundbreaking scalability, the generative architecture suffers from distinct synthesis robustness limitations. Because the phoneme-to-acoustic transcription layer relies on an unconstrained autoregressive language model, it can experience disordered attention alignments that cause words to be unclear, completely omitted, or randomly duplicated during inference.

While scaling to multi-thousand-hour repositories provides exceptional zero-shot generalization, other recent developments in the speech domain have focused

alternatively on targeted adaptation strategies for low-resource settings and parameter efficiency. One prominent strategy tackles the lack of text-labeled data by using a self-supervised transfer learning framework on large-scale unlabeled speech corpora [15]. This system uses a pre-trained wav2vec 2.0 model and k-means clustering to find unsupervised "pseudo-phonemes" directly from raw, untranscribed speech. During fine-tuning on labeled data, the authors use a specific freezing template: the posterior encoder and decoder are locked completely, and only the normalizing flow is optimized to bridge the gap between pseudo-phonemes and real text inputs. While this approach creates stable speech using only 10 to 30 minutes of labeled audio achieving a MOS of 4.21 ± 0.10 and a similarity score of 3.92 ± 0.09, it faces a major theoretical limit. Because the pre-trained pseudo-phonemes and real target text phonemes share no common feature intersections ($\mathcal{X}_S \cap \mathcal{X}_T = \emptyset$), the model remains highly sensitive to domain mismatches. When fine-tuned on larger datasets, its performance often drops due to the complex, indirect alignment required to bridge these separate spaces.

To improve zero-shot speech representations without text data, another approach integrates a non-parametric self-distillation framework into the integrated end-to-end backbone. Pankov et al. [20] run a student-teacher distillation setup across massive multi-speaker repositories like LibriLight, this architecture achieves high zero-shot performance on unseen voices, scoring a MOS of 4.00 ± 0.05 and a similarity rating of 3.85 ± 0.08 on the VCTK evaluation set. However, this self-supervised setup introduces massive architectural complexity. The addition of extra projection heads, teacher momentum updates, and multi-scale data cropping significantly increases computational overhead, restricting its training path to enterprise-scale, multi-GPU computing clusters.

Alongside self-supervised data solutions, research has turned toward structural adaptations aimed directly at lowering the storage footprint of personalized voices. A significant development in this direction is the Mixture of Adapters (MoA) framework, which embeds lightweight bottleneck adapter modules directly into the decoder layers of a non-autoregressive system [7]. Managed by a sparse gating network conditioned on speaker identity, this layout activates a limited subset of parameters per voice. This strategy achieves a 1.9x inference speedup using less than 40% of the total parameter state, while securing balanced listener preference scores of 36% for both naturalness and similarity against a large 42M baseline. Despite these gains, the framework requires a fundamental modification of the underlying network structure. This introduces a large cascade of new hyper-parameters and routing weights that complicate the training loop.

Another recent approach to managing parameter scaling splits the waveform decoding path into a hybrid layout to save computational memory. This configuration maps low-frequency speech components using standard upsampling layers and high-frequency components using inverse Short-Time Fourier Transforms

(iSTFT), reaching a MOS of 3.15 ± 0.13 [26]. However, to maintain alignment stability under these constrained parameter states, this framework requires extracting external Phonetic PosteriorGrams (PPGs) from an independent, pre-trained speech recognition model. This introduces an external feature dependency that increases processing latency and reduces its efficiency in low-resource settings.

Following the examination of the conducted research, Table 2.1 and Table 2.2 compare the features of each study from the perspective of advantages, disadvantages, the architecture used, and their evaluation criteria.

Table 2.1: The advantages and disadvantages of the conducted studies on speech synthesis and voice cloning

Year	Approach	Advantages	Disadvantages	Ref.
1980s	Parametric & Unit Selection Synthesis	Very low computing power needed; provides direct manual control over individual speech features.	Sounds robotic and mechanical; humans cannot manually program all natural variations of speech.	[16]
1990s	Parametric & Unit Selection Synthesis	High naturalness within individual clips; completely bypasses complex acoustic modeling steps.	Dependent on database size; prone to audible clicks or glitches at clip boundaries.	[10]
2000s	Parametric & Unit Selection Synthesis	Drastically reduces storage size; allows basic voice changes using linear transformations.	Built-in mathematical averaging causes over-smoothing, resulting in a muffled and flat voice.	[30]
2016	Decoupled Multi-Stage Pipelines	Groundbreaking design; captures fine-grained, realistic audio details directly from raw waveforms.	Extremely high processing delays during inference because it must generate speech one sample at a time.	[31]

Table continues on next page

Year	Approach	Advantages	Disadvantages	Ref.
2017	Decoupled Multi-Stage Pipelines	First production-grade deep learning layout; separates distinct speech tasks into clear modules.	Modular structure suffers from compounding errors across separate prediction blocks.	[2]
2017	Decoupled Multi-Stage Pipelines	Introduces multi-speaker support from a single network using low-dimensional speaker embeddings.	Still limited by a rigid multi-step pipeline design and independent module tracking.	[8]
2017	Decoupled Multi-Stage Pipelines	Eliminates multi-stage pipelines by mapping input text directly to intermediate mel-spectrograms.	Requires a lossy linear inversion step that discards phase data and introduces distortion.	[34]
2017	Decoupled Multi-Stage Pipelines	Can handle unaligned, unstructured voice samples collected from real-world environments.	Shifting buffer memory setup struggles to replicate highly dynamic or unique speaker voices.	[29]
2017	Decoupled Multi-Stage Pipelines	Replaces recurrent paths with convolutional layers to significantly accelerate neural network training.	Suffers from severe text-to-speech alignment failures when training data is limited.	[21]
2018	Decoupled Multi-Stage Pipelines	Reaches excellent speech naturalness by pairing an integrated decoder with a neural vocoder.	Independent multi-stage optimization leads to acoustic target over-smoothing; the autoregressive attention mechanism remains prone to alignment drift and word errors on long inputs.	[25]

Table continues on next page

Year	Approach	Advantages	Disadvantages	Ref.
2018	Decoupled Multi-Stage Pipelines	Demonstrates zero-shot speaker adaptation capabilities using a separate speaker encoder framework.	Relies on a fixed representation from a verification objective, dropping fine-grained voice details.	[11]
2019	Parallel Explicit Feature Generation	Eliminates the attention bottleneck entirely; achieves massively accelerated parallel generation.	Relies on an independent teacher model for timing data, which smooths out expressive voice details.	[24]
2020	Parallel Explicit Feature Generation	Removes teacher dependency; trains directly on ground-truth targets like real pitch and energy.	Retains a strict structural dependency on external alignment tools to compute necessary phonetic duration maps during training.	[23]
2020	Parallel Explicit Feature Generation	Removes outside alignment dependencies by using an internal Monotonic Alignment Search engine.	Flow-based inverse modeling can increase overall structural and parameter layout complexity.	[13]
2021	Parallel Explicit Feature Generation	Provides excellent pitch control by linking fundamental frequency directly to individual words.	Retains a structural dependency on outside tools to calculate phoneme duration maps.	[18]
2021	Integrated End-to-End Synthesis	Direct mapping from text to raw audio; minimizes intermediate feature loss and associated downstream tracking errors.	Requires full-parameter updates for adaptation, leading to high computational costs and large storage needs.	[14]

Table continues on next page

Year	Approach	Advantages	Disadvantages	Ref.
2021	Integrated End-to-End Synthesis	Uses soft temporal alignment to convert phonemes directly to raw speech without multi-stage supervision.	Requires an exceptionally large training batch footprint and massive TPU cluster acceleration to avoid classic adversarial stability failures and mode collapse.	[4]
2022	Parameter-Efficient Speaker Adaptation	Leverages large-scale unlabeled speech sets for self-supervised pre-training, utilizing a frozen decoder template during adaptation to minimize parameter drift.	Highly sensitive to domain mismatches because the source and target feature spaces do not intersect.	[15]
2023	Integrated End-to-End Synthesis	Allows multi-speaker and multi-language voice cloning with global conditioning paths.	Bypassing phonetic transcriptions increases word mispronunciations, and the stochastic duration predictor exhibits rhythmic instabilities.	[3]
2023	Integrated End-to-End Synthesis	Demonstrates emergent zero-shot voice cloning via acoustic in-context learning after pre-training on a massive, 60K-hour speech corpus.	Autoregressive modeling of discrete acoustic codes is prone to synthesis robustness issues, causing words to be unclear, omitted, or duplicated due to disordered attention alignments.	[33]

Table continues on next page

Year	Approach	Advantages	Disadvantages	Ref.
2024	Parameter-Efficient Speaker Adaptation	Enhances zero-shot speaker similarity and noise robustness via a multi-task joint-training framework that updates the speaker encoder using self-supervised DINO loss.	Requires a complex four-module pipeline (HuBERT, mBART, CAM++, VITS) that depends on independent component pre-training and a strict multi-stage fine-tuning schedule.	[20]
2024	Parameter-Efficient Speaker Adaptation	Injects lightweight bottleneck adapters into decoders and variance predictors, minimizing additional parameters while matching large-scale model quality.	The static variant fails to differentiate professional from non-professional speaker identities.	[7]
2024	Parameter-Efficient Speaker Adaptation	Keeps the trainable parameter count low by cutting high and low voice frequencies into separate decoding paths.	Requires an external, pre-trained Automatic Speech Recognition (ASR) framework to extract frame-level Phonetic Posteriorgram (PPG) features for content-timbre decoupling.	[26]

Existing research explores several paths for parameter-efficient voice cloning, but each fails to meet the specific requirements of a native VITS setup under limited target data constraints. First, adapter modules insert lightweight blocks into network layers. While they limit active parameters, they alter the underlying network layout. This structural change adds training complexity and new hyperparameters, contradicting our goal of keeping the baseline VITS model completely unmodified for easy deployment. Second, hybrid decoding designs manage parameter scaling by splitting decoding paths. However, they frequently rely on external feature pipelines, such as extracting Phonetic Posteriorgrams from a separate speech recognition model. This reliance creates external tool dependencies

and increases processing latency, defeating the purpose of a fast, self-contained end-to-end framework. Finally, self-supervised pre-training frameworks often map text inputs to unsupervised speech units. Because the source and target feature spaces do not directly overlap, these systems are highly sensitive to domain mismatches. In data-scarce scenarios, this sensitivity can cause severe alignment drift and degrade speech clarity.

Freezing the text and posterior encoders directly addresses this research gap. Unlike adapters, this strategy requires zero structural changes, leaving the original deployment graph completely untouched. In contrast to hybrid decoding, it remains entirely self-contained within a single model footprint, avoiding external feature extraction tools and extra inference delays. Finally, instead of experiencing the domain mismatches common in self-supervised frameworks, it relies on the native, pre-trained language mappings as a stable reference point. This choice protects pronunciation and language rules under data constraints, allowing the active downstream modules to focus cleanly on adapting to the target speaker's unique voice traits.

This research examines the trade-offs between model design and hardware limits in voice cloning. While full-parameter fine-tuning serves as a strong baseline for high speech quality, updating the entire network requires significant computing power and memory during training. The proposed method aims to match this baseline quality while updating fewer parameters.

To the author's knowledge, no previous study has tested freezing the text and posterior encoders while keeping the decoder, conditional normalizing flow, stochastic duration predictor, speaker embedding, and discriminator fully trainable inside an unmodified VITS model for adaptation under limited target data constraints. By exploring these structural boundaries, this thesis introduces a selective training strategy within the original layout to optimize training-stage hardware resources without altering deployment complexity.

Table 2.2: Comprehensive overview of voice cloning and TTS methods and their performance

Ref.	**Approach**	**Datasets Used**	**Evaluation Metrics**	
			Subjective Evaluation	**Objective Evaluation**
[25]	Integrated Multi-Stage Synthesis	Google Internal (English)	MOS: 4.53 ± 0.07	—
[20]	Self-Supervised Distillation Representation	VCTK, LibriTTS, LibriLight	MOS: 4.00 ± 0.05 (VCTK) SMOS: 3.85 ± 0.08 (VCTK)	—

Table continues on next page

Ref.	Approach	Datasets Used	Evaluation Metrics	
			Subjective Evaluation	**Objective Evaluation**
[33]	Neural Codec Language Modeling	LibriLight (60,000h)	SMOS: 3.81 ± 0.09	WER: 5.9%
[11]	Decoupled Multi-Stage Identity Transfer	LibriSpeech, VCTK	SMOS: 4.22 ± 0.06 (VCTK) SMOS: 3.28 ± 0.08 (LibriSpeech)	—
[7]	Sparse Gated Layout Adaptation	In-house Japanese (960h)	AB (Nat.): 36% XAB (Sim.): 36% (vs. 42M baseline)	MCD: ~4.8 dB
[3]	Zero-Shot Multilingual Cross-Conditioning	VCTK, LibriTTS, TTS-Portuguese, MAILABS	MOS: 4.21 ± 0.04	SECS: 0.864
[26]	Hybrid Frequency-Split Decoding	LibriTTS, VCTK	MOS: 3.15 ± 0.13 (VCTK) SMOS: 3.12 ± 0.12 (VCTK)	WER: 8.17% (VCTK)
[15]	Feature Space Domain Alignment	VCTK, LJSpeech	MOS: 4.21 ± 0.10 (VCTK) SMOS: 3.92 ± 0.09 (VCTK)	CER: 8.5% (VCTK) SECS: 0.298 (VCTK)

Abbreviations: MOS (↑) = Mean Opinion Score (Naturalness); SMOS (↑) = Similarity MOS; SECS (↑) = Speaker Embedding Cosine Similarity; WER (↓) = Word Error Rate; CER (↓) = Character Error Rate; MCD (↓) = Mel-Cepstral Distortion; PPG = Phonetic PosteriorGram.

To validate these architectural choices, this work includes a systematic ablation study. The results demonstrate that selectively locking specific modules achieves comparable voice quality and speaker similarity to the full fine-tuning baseline, confirming that updates to these components can be safely bypassed. Additionally, the experiments show that the adversarial discriminator block remains critical to the optimization loop, as freezing its weights triggers an immediate training collapse. This selective strategy targets the encoder modules because their pre-trained layers already provide robust, generalized text representations. By disabling gradient tracking for these layers, AdaptVITS prevents redundant updates to linguistic features and reduces peak physical GPU memory (VRAM) consumption. Ultimately, this targeted approach preserves the high perceptual quality and real-time generation speeds of full-parameter fine-tuning while significantly improving training

efficiency.

Chapter 3

Methodology

The primary objective of this research is the formulation of AdaptVITS, a parameter-efficient speaker adaptation framework operating entirely within the native, unmodified topology of the Variational Inference with Text-to-Speech (VITS) architecture [14]. Conventional voice cloning paradigms rely on full-parameter fine-tuning as the empirical standard for maximizing synthesized speech naturalness, speaker similarity, and inference velocity. Rather than contesting this paradigm, AdaptVITS treats full fine-tuning as its primary baseline, aiming to match its high-fidelity performance while optimizing training-stage memory constraints and active parameter tracking overhead. To preserve a completely native network layout without introducing auxiliary bottleneck adapters, routing parameters, or external pipelines, this transfer learning strategy modifies internal optimization paths based on an engineering distinction between speaker-independent linguistic features and speaker-dependent acoustic configurations. Because the upstream text and posterior encoders capture robust, generalized language properties during pre-training, updating their weights during short adaptation tracks is computationally redundant. Consequently, AdaptVITS disables gradient tracking for these stable blocks, utilizing them as frozen feature extractors while restricting active parameter updates exclusively to the downstream synthesis modules. To detail these mechanisms, this chapter formalizes the baseline VITS architecture and introduces the proposed AdaptVITS framework. Additionally, the chapter defines the randomized dataset curation protocol, the unified training configurations, and the specific subjective and objective evaluation metrics used to assess downstream voice cloning performance.

3.1 Baseline Architecture: Native VITS Topology

To establish a rigorous framework for parameter-efficient adaptation, it is necessary to formalize the baseline Variational Inference with Text-to-Speech (VITS) architecture [14]. Prior neural speech synthesis networks predominantly operate as decoupled, multi-stage cascades. These configurations optimize an upstream acoustic model such as Tacotron2 [25] or FastSpeech2 [23] to map text inputs onto intermediate acoustic features (typically mel-spectrograms), which are subsequently passed into an independent neural vocoder like HiFi-GAN [17] for waveform reconstruction.

While effective, these multi-stage cascades suffer from compounding error propagation and loss of acoustic detail. The intermediate abstraction layer discards absolute phase information, and any structural errors or alignment drift introduced by the acoustic model are magnified during vocoding, causing word omissions, repetitions, or synthetic distortion. Furthermore, because these separate modules are optimized using independent, localized loss criteria, the global system cannot converge on a mathematically optimal representation that links the linguistic and raw audio domains.

VITS resolves these modular vulnerabilities by unifying text processing, latent alignment, and waveform synthesis into a single, end-to-end differentiable computational graph. This framework leverages a conditional Variational Autoencoder (cVAE) combined with conditional normalizing flows and parallel adversarial training to execute a direct text-to-waveform mapping. For reduced target speaker adaptation, this unified design provides high stability; encapsulating the entire generative path into a single network keeps the hidden states tightly bound to a shared continuous latent space $\mathcal{Z}$. Built-in speaker conditioning allows the system to scale smoothly across multiple voice profiles by passing a global speaker identity embedding g to the internal conditioning layers of the network. This comprehensive pipeline preserves natural prosody and speaker characteristics while eliminating the need for external vocoding architectures.

The baseline VITS architecture is designed as a conditional Variational Autoencoder that connects text sequences and raw audio waveforms within a shared continuous latent space $\mathcal{Z}$. This system unifies five core components: a posterior encoder, a prior encoder (integrated with a conditional normalizing flow), an internal monotonic alignment search mechanism, a parallel waveform decoder, and an adversarial discriminator network.

During training, the posterior encoder acts as an acoustic feature extractor. It processes the linear spectrogram x_{spec} of the target audio through residual blocks to generate a Gaussian latent acoustic distribution q_ϕ. The resulting latent variable $z \in \mathcal{Z}$ captures continuous acoustic properties such as pitch contours, timbre, and phase dynamics. Concurrently, the prior encoder processes the input phoneme sequence c to produce a text representation. Because discrete text contains less

statistical complexity than continuous speech waveforms, a conditional normalizing flow f_θ is applied. This normalizing flow uses a series of invertible transformations to expand the text distribution into a complex, detailed representation that matches the acoustic latent space, conditioned on a global speaker embedding g.

To resolve the timing differences between discrete text tokens and continuous speech frames, the internal Monotonic Alignment Search (MAS) engine maps out the relationship between text and audio. The MAS engine determines the optimal, non-decreasing alignment path A^* by maximizing the log-likelihood of the data within the prior latent space:

$$A^* = \arg\max_A \log p_\theta(z \mid c, A, g) \tag{3.1}$$

Once this alignment is calculated, the latent features are sent directly to the parallel decoder G, which uses upsampling layers based on the HiFi-GAN generator topology to synthesize the raw time-domain audio waveform $\hat{y} = G(z, g)$. To refine this synthesis, an adversarial discriminator network D evaluates the realism of the generated audio across multiple scales and periods. This feedback provides a continuous training loop that drives the decoder to generate clear and natural speech.

The entire architecture is optimized from scratch by minimizing a unified, multi-task loss function L_{VITS}, which balances reconstruction accuracy with structural and training stability:

$$L_{\text{VITS}} = L_{\text{recon}}(G) + L_{\text{KL}}(f_\theta) + L_{\text{dur}}(G) + L_{\text{adv}}(G, D) + L_{\text{fm}}(G, D) \tag{3.2}$$

Within this objective, the reconstruction loss (L_{recon}) minimizes the absolute difference between the mel-spectrograms of the real and synthesized waveforms, while the Kullback-Leibler divergence (L_{KL}) keeps the prior and posterior latent distributions closely aligned. The remaining terms optimize the timing and rhythm of speech via a stochastic duration model (L_{dur}), ensure realistic waveform generation through an adversarial objective (L_{adv}), and stabilize the overall training process using discriminator feature matching (L_{fm}).

This unified framework forms a robust baseline for voice cloning. Because the encoders learn generalized linguistic and acoustic structures that stay consistent across English speech patterns, they provide a reliable foundation for transfer learning, allowing the downstream decoder and discriminator layers to focus cleanly on learning the unique vocal traits of a target speaker.

3.2 Proposed Approach: AdaptVITS

The AdaptVITS framework, illustrated in Figure 3.1, is introduced to optimize training-stage resource utilization without modifying the native network architecture or altering speech quality. Standard full-parameter fine-tuning serves as a

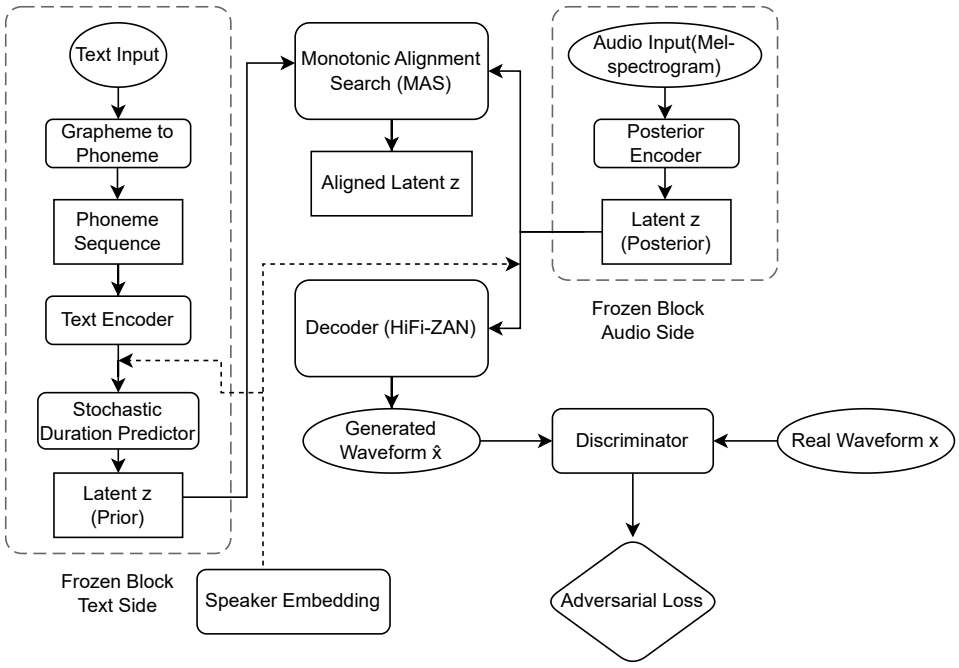


Figure 3.1: Diagram of the AdaptVITS training configuration.

robust baseline for generating high-fidelity speech, but it updates all parameter matrices uniformly, which tracks gradients across the entire network structure during the backpropagation phase. AdaptVITS scales down this optimization requirement through a parameter-efficient selective freezing strategy. This approach bypasses updates to modules that already contain generalized, speaker-independent text representations, thereby lowering training-stage memory demands and focusing exclusively on the layers responsible for adapting specific speaker characteristics.

The AdaptVITS selective freezing strategy is implemented by dividing the complete VITS parameter space Θ into two separate, non-overlapping blocks:

$$\Theta = \Theta_{\text{frozen}} \cup \Theta_{\text{active}} \quad \text{where} \quad \Theta_{\text{frozen}} \cap \Theta_{\text{active}} = \emptyset \tag{3.3}$$

The frozen parameter block contains the weights and biases of the upstream linguistic and phonetic feature extraction modules. These components learn generalized structures of speech during multi-speaker pre-training that do not require modification for individual voices:

$$\Theta_{\text{frozen}} = \{\theta_{\text{text\_encoder}}, \theta_{\text{posterior\_encoder}}\} \tag{3.4}$$

The active parameter block retains the modules responsible for tracking target

speaker identity, vocal variance, and time-domain audio waveform synthesis:

$$\Theta_{\text{active}} = \{\theta_{\text{decoder}}, \theta_{\text{discriminator}}, \theta_{\text{flow}}, \theta_{\text{duration\_predictor}}, \theta_{\text{speaker\_embeddings}}\} \quad (3.5)$$

During the fine-tuning phase, gradient tracking is disabled for the frozen block within the computational graph:

$$\forall \theta_i \in \Theta_{\text{frozen}}, \quad \frac{\partial L_{\text{VITS}}}{\partial \theta_i} = 0 \quad (3.6)$$

As a result, backpropagation updates weights exclusively within the active parameter space:

$$\Theta_{\text{active}}^{(t+1)} = \Theta_{\text{active}}^{(t)} - \eta \nabla_{\Theta_{\text{active}}} L_{\text{VITS}} \quad (3.7)$$

where η represents the fine-tuning learning rate.

By locking Θ_{frozen}, AdaptVITS reduces the number of active trainable parameters by 17.5%. This intervention eliminates the requirement to track and store gradient matrices for the frozen layers during the backward pass. This reduction saves approximately 115 MB of theoretical gradient memory overhead, which contributes directly to an empirical reduction in observed peak physical hardware memory (VRAM) from 5974 MB down to 5201 MB on the experimental hardware configuration. Because this modification occurs entirely during the training phase, the underlying structure of the model remains completely unchanged. As a consequence of maintaining an identical deployment graph, the final checkpoint file size, latent dimensions, and runtime inference speed remain identical to the standard VITS baseline, ensuring full deployment compatibility without introducing any architectural alterations to the runtime inference pipeline.

Beyond the encoder modules, the VITS architecture consists of several other core components, including the normalizing flow and the waveform decoder, etc. To evaluate the impact of broader parameter constraints, the layer-freezing process was systematically applied to these different remaining modules during development, with the complete performance metrics detailed in Chapter 4.

3.3 Dataset Profiles and Selection

To test speech synthesis with limited data, we use a two-stage training pipeline: high-resource pre-training followed by limited-data speaker adaptation. This approach separates the process of learning general language structures from the task of cloning a specific voice identity.

3.3.1 Pre-training Dataset (LibriTTS)

The base model is pre-trained on the `train-clean-360` subset of the LibriTTS corpus [36]. This dataset contains clean audiobook speech sampled at 24 kHz, which helps

Table 3.1: Signal Quality and Acoustic Distribution of VCTK 5h Subsets

Acoustic Metric	Subset 1 (Seed 42)	Subset 2 (Seed 123)	Subset 3 (Seed 456)
Total Utterances	5,344	5,350	5,361
Sampling Rate / Format	24 kHz / Mono	24 kHz / Mono	24 kHz / Mono
Audio Bit Depth	16-bit PCM	16-bit PCM	16-bit PCM
Total Duration (h)	5.0002	5.0001	5.0004
Mean Utterance Duration (s)	3.3684	3.3646	3.3579
Variance of Duration (s^2)	1.4349	1.3716	1.3480
Mean RMS Energy	0.0352	0.0349	0.0349
Variance of RMS Energy ($\times 10^{-4}$)	1.6262	1.6777	1.6306

the model learn basic English sounds and text alignments. The pre-training dataset has the following statistics:

- **Files:** 116,500 matching audio files and normalized text transcripts, containing a total of 11,367,108 characters.
- **Speakers:** 904 distinct speakers, averaging 128.9 files per speaker (median: 117.5, minimum: 1, maximum: 515).
- **Duration:** A total duration of 191 hours, 17 minutes, and 42 seconds. Individual file lengths range from under 1 second up to 40 seconds, with a average duration of 6 seconds (median: 5 seconds).

3.3.2 Fine-Tuning Dataset and Random Subsets (CSTR VCTK)

We use the high-fidelity CSTR VCTK corpus [32] for speaker adaptation and fine-tuning baselines. This dataset contains clean studio recordings of speakers with various regional accents, recorded using identical microphones in an echo-free chamber.

Before creating our subsets, we removed two speakers with data errors: speaker `p315` due to corrupted text files, and speaker `p362` because the audio files were too short. This left 108 valid speakers.

To simulate a limited-data scenario, we restricted the training data to a strict limit of 5 hours. To avoid picking a bias clean set of sentences, we created three independent 5-hour subsets using random seeds 42, 123, and 456. Each subset splits the 5 hours evenly across all 108 speakers, providing an average of about 2.7 minutes of audio per speaker.

We verified the acoustic balance of these subsets by checking their file lengths and Root Mean Square (RMS) energy. Table 3.1 shows that the three subsets are highly uniform across all seeds.

We also checked the text transcripts by converting them into phonetic sequences using the standard 39-phone ARPAbet dictionary. As shown in Table 3.2, all three subsets achieve 100% phonetic coverage, meaning every standard sound in the

Table 3.2: Textual and Phonetic Coverage Matrix

Phonetic Metric	**Subset 1 (Seed 42)**	**Subset 2 (Seed 123)**	**Subset 3 (Seed 456)**
Total Transcribed Words	46,079	45,852	46,050
Total Phoneme Tokens	139,346	138,117	138,761
ARPAbet Phoneme Coverage	**100.00% (39/39)**	**100.00% (39/39)**	**100.00% (39/39)**

Table 3.3: Summary of Experimental Configurations and Hyperparameters

Configuration Name	**Pre-trained Weights**	**Frozen Modules**	**Target Dataset**	**Learning Rate**	**Epochs**
Pre-training Foundation	None (From Scratch)	None	LibriTTS 360h	2×10^{-4}	275
Scratch Baseline	None (From Scratch)	None	VCTK 5h (Subsets 1, 2, 3)	2×10^{-4}	302
Full Fine-Tuning Baseline	`best_model_1207645.pth`	None	VCTK 5h (Subsets 1, 2, 3)	2×10^{-5}	302
Proposed Approach	`best_model_1207645.pth`	**Text & Posterior Encoders**	**VCTK 5h (Subsets 1, 2, 3)**	2×10^{-5}	**302**
Decoder Ablation	`best_model_1207645.pth`	Decoder	VCTK 5h (Subsets 1, 2, 3)	2×10^{-5}	302
Flow Ablation	`best_model_1207645.pth`	Normalizing Flow	VCTK 5h (Subsets 1, 2, 3)	2×10^{-5}	302
Duration Pred. Ablation	`best_model_1207645.pth`	Duration Predictor	VCTK 5h (Subsets 1, 2, 3)	2×10^{-5}	302
Discriminator Ablation	`best_model_1207645.pth`	Discriminator	VCTK 5h (Subsets 1, 2, 3)	2×10^{-5}	302

English language is present during training. The distribution of these phonemes is highly consistent across all three subsets.

3.4 Training Workflow and Experimental Configurations

This section describes the training phases, hyperparameters, and model configurations used in this study. The training workflow consists of three steps: multi-speaker pre-training, target speaker fine-tuning, and architectural ablation variations. Table 3.3 summarizes these configurations.

3.4.1 Pre-training Phase

The base VITS model is first trained from scratch on the LibriTTS `train-clean-360` dataset for 275 epochs, which corresponds to approximately 1.2M training steps. This phase allows the model to learn basic text-to-speech alignments and general English language features. It produces the base foundation checkpoint, which serves as the identical starting point for all downstream fine-tuning and parameter-efficient experiments.

3.4.2 Fine-Tuning Specifications

For target speaker adaptation and baseline comparisons, we train the models for a fixed duration of 302 epochs (100k steps) on the 5-hour VCTK subsets. All models are implemented using the open-source Coqui TTS framework [5] and executed on a single NVIDIA RTX 3060 GPU with 12 GB of VRAM.

We use 16-bit Automatic Mixed Precision (AMP) to reduce GPU memory usage and speed up training. The training batch size is set to 16 samples, and the validation

batch size is 8 samples. The model parameters are updated using the AdamW optimizer with momentum values of $\beta_1 = 0.8$, $\beta_2 = 0.99$, and an epsilon value of $\epsilon = 10^{-9}$. We use an initial learning rate of 2×10^{-4} for the scratch baseline and a lower rate of 2×10^{-5} for all fine-tuning configurations to preserve the pre-trained features.

We train and evaluate each setup independently across all three randomized datasets (VCTK 5h Subsets 1, 2, and 3). We compare three primary configurations under these standard conditions:

- **Scratch Training Baseline:** The VITS model trains entirely from random weights with no pre-trained initialization. All layers are active and trainable during optimization using a default learning rate of 2×10^{-4}.
- **Full Fine-Tuning Baseline:** The model loads the pre-trained weights from the base checkpoint, and all internal modules remain open and trainable. These parameters are optimized using a learning rate of 2×10^{-5}, serving as our primary high-quality benchmark.
- **Proposed Approach (AdaptVITS):** The model loads the pre-trained base checkpoint, but we completely freeze both the Text Encoder and the Posterior Encoder modules to keep the latent space stable. All other layers remain active and trainable during optimization using a learning rate of 2×10^{-5}.

3.4.3 Architectural Ablation Design

To determine which individual module controls speaker voice identity and rhythm, we perform a systematic ablation study. Each variant initializes from the pre-trained base checkpoint, trains for 302 epochs (100k steps), and uses a learning rate of 2×10^{-5}. To ensure our findings are stable and reliable, we run these experiments independently across all three randomized datasets (VCTK 5h Subsets 1, 2, and 3). These setups isolate specific modules by freezing them while keeping all other components active:

1. **Decoder Ablation:** Freezes the waveform decoder module completely, leaving all text, flow, and posterior processing layers fully trainable.
2. **Flow Ablation:** Locks the conditional normalizing flow architecture, while all remaining generative and alignment modules are updated during fine-tuning.
3. **Duration Predictor Ablation:** Freezes the stochastic duration predictor module to evaluate how it impacts target speaker rhythm and phrasing stability.
4. **Discriminator Ablation:** Freezes the multi-scale and multi-period adversarial discriminator blocks while keeping the generator modules active. The optimization challenges and constraints discovered during this specific experiment are discussed alongside the results in Chapter 4.

3.5 Evaluation Metrics

To evaluate the quality, clarity, and voice cloning accuracy of the synthesized speech, we use a combination of objective and subjective metrics. All model configurations are tested on a fixed validation set of 100 unseen utterances. These sentences were completely held out from training to ensure a completely fair comparison.

3.5.1 Objective Evaluation Metrics

Objective metrics provide automated, reproducible measurements of speech intelligibility and speaker similarity. We track three main metrics: Word Error Rate (WER), Character Error Rate (CER), and Speaker Encoder Cosine Similarity (SECS).

Word Error Rate and Character Error Rate

To evaluate how clear and understandable the generated speech is, we pass the synthesized audio through a pre-trained Automated Speech Recognition (ASR) model to transcribe it back into text. We then compare these transcriptions against the original ground-truth text using Word Error Rate (WER) and Character Error Rate (CER).

WER measures text accuracy at the word level and is calculated using the following formula:

$$\text{WER} = \frac{S + D + I}{N} \tag{3.8}$$

where S is the number of substitutions (replaced words), D is the number of deletions (missing words), I is the number of insertions (added extra words), and N is the total number of words in the reference text.

CER uses the exact same calculation method but checks errors at the individual character level instead of full words. Lower WER and CER scores mean the generated audio is highly intelligible and cleanly pronounced.

Speaker Encoder Cosine Similarity

To verify if the model successfully cloned the target speaker's voice, we calculate the Speaker Encoder Cosine Similarity (SECS). This metric checks how closely the vocal characteristics of the generated audio match the true speaker.

We use a pre-trained speaker verification network from the SpeechBrain framework to extract fixed speaker embedding vectors from both the synthesized audio (v_{synth}) and the original reference audio (v_{ref}). The likeness is calculated as the cosine similarity between these two vectors:

$$\text{SECS} = \frac{v_{\text{synth}} \cdot v_{\text{ref}}}{\|v_{\text{synth}}\|\|v_{\text{ref}}\|} \tag{3.9}$$

The SECS score ranges between -1 and 1. A higher score means the cloned voice sounds much closer to the target speaker's identity.

3.5.2 Subjective Evaluation Metric

Because automated software cannot always tell if an audio clip sounds natural or pleasant to human ears, we also use human testing to verify speech quality.

Mean Opinion Score

We measure human perceptual quality using the standard Mean Opinion Score (MOS) [27]. Human testers score the synthesized audio samples from 1 to 5 based on their naturalness and clarity. The scoring options are: 1 (Bad), 2 (Poor), 3 (Fair), 4 (Good), and 5 (Excellent). We report the final average of these scores alongside a 95% confidence interval to assess and demonstrate statistical consistency.

Chapter 4

Results and Discussion

This chapter presents the test results for AdaptVITS compared to standard baseline models. To see how well the system handles limited data, we evaluated all configurations using separate 5-hour speaker datasets. The analysis is broken down into three main parts: human listening tests (MOS) for speech quality, automated tests for word clarity (WER/CER) and voice similarity (SECS), and hardware checks for training memory usage and generation speed. Together, these results indicate that AdaptVITS delivers stable, high-quality voice cloning.

4.1 Subjective Perceptual Analysis

We conducted a human listening test with 18 independent listeners to evaluate the quality of the synthesized speech. Listeners graded randomly presented audio clips on the standard 5-point Mean Opinion Score (MOS) scale (1: Bad, 5: Excellent) across six system configurations: original, uncompressed reference recordings (System A); a native VITS model trained from random weights on 5 hours of data (System B); and a standard VITS model fully fine-tuned with all layers open on 5 hours of data (System C). The final three setups evaluated our proposed parameter-efficient model, AdaptVITS, across independent 5-hour data subsets using random seeds.

To protect human raters from cognitive fatigue and maintain a high level of evaluative focus, the baseline models (System B and System C) were optimized and evaluated exclusively on Subset 1. Each system was represented by 7 distinct validation audio clips taken from our prepared validation set. This resulted in a total pool of 42 audio clips for each rater to score. The empirical results of this listening test are summarized in Table 4.1 and visualized in Figure 4.1.

The baseline configurations establish clear performance boundaries. System B experienced a severe quality collapse (1.59 ± 0.2245) due to severe phonetic alignment and distortion issues, confirming that VITS cannot successfully converge from

Table 4.1: Subjective Mean Opinion Score (MOS) Performance Matrix

System ID	Model Configuration Profile	Active Parameters (%)	MOS ($\pm CI_{95\%}$)
System A	Ground Truth	—	4.52 ± 0.3223
System B	Scratch Optimization Baseline	100.00%	1.59 ± 0.2245
System C	Full Fine-Tuning Baseline	100.00%	3.84 ± 0.3023
System D	AdaptVITS (Subset 1, Seed 42)	82.50%	3.92 ± 0.3075
System E	AdaptVITS (Subset 2, Seed 123)	82.50%	3.82 ± 0.3128
System F	AdaptVITS (Subset 3, Seed 456)	82.50%	3.87 ± 0.3996
Summary	**AdaptVITS (Aggregated Across Runs)**	**82.50%**	**Mean: 3.87, Variance: 0.003**

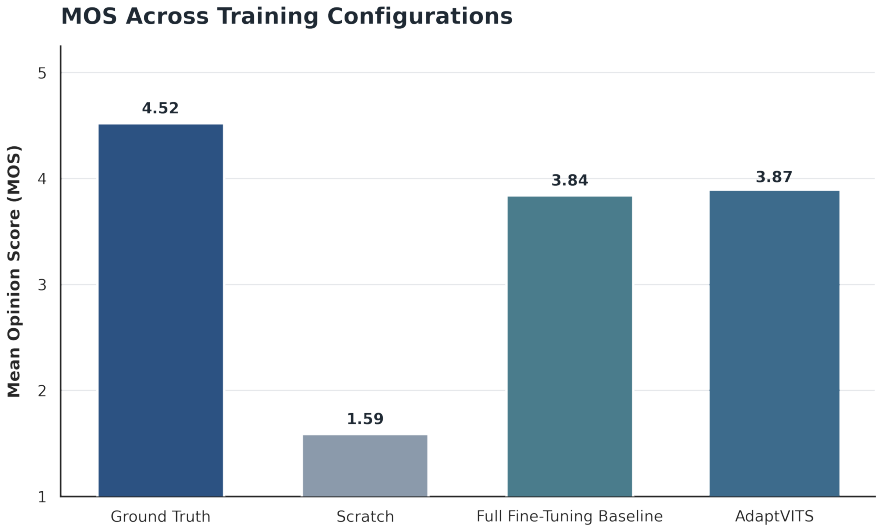


Figure 4.1: MOS scores across all training configurations.

scratch on a limited 5-hour dataset. Conversely, full fine-tuning (System C) restored speech quality to an elite baseline score of 3.84 ± 0.3023.

The proposed AdaptVITS models (Systems D, E, and F) achieved strong, consistent performance with individual mean scores of 3.92, 3.82, and 3.87, respectively. By updating only 82.50% of the parameters, our selective layer-freezing strategy successfully preserved the foundational language structures learned during pre-training. Aggregating across the three independent data splits, the framework achieved a macro-mean MOS of 3.87 with an exceptionally low cross-run variance of 0.003.

To verify that the observer trends represent genuine consensus rather than random voting noise, the panel's scores were analyzed using a global Friedman test [6] and Kendall's Coefficient of Concordance (W) [12]. These reliability metrics are detailed in Table 4.2.

The global Friedman test strongly rejected the null hypothesis that all systems perform identically ($\chi^2 = 57.1834$, $p < 0.05$), confirming that the variations in

Table 4.2: Global Rater Agreement and Concordance Metrics

Statistical Parameter Metric	**Empirical Evaluated Value**
Number of Raters (m)	18
Number of Evaluated Systems (n)	6
Friedman Chi-Squared Metric (χ^2)	57.1834
Friedman Significance (p-value)	0.0000
Kendall's Coefficient of Concordance (W)	0.6354
Global Qualitative Interpretation	**Moderate-to-Strong Agreement**

Table 4.3: Holm-Bonferroni Corrected Post-Hoc Pairwise Evaluation Matrix

Comparison Pair	**Evaluation Comparison Path**	**Raw p-Value**	**Holm Adjusted p-Value**	**Significant ($\alpha = 0.05$)**
Pair 1	System A vs. System B	0.0002	0.0028	Yes
Pair 2	System A vs. System C	0.0035	0.0245	Yes
Pair 3	System A vs. System D	0.0003	0.0028	Yes
Pair 4	System A vs. System E	0.0002	0.0028	Yes
Pair 5	System A vs. System F	0.0003	0.0028	Yes
Pair 6	System B vs. System C	0.0002	0.0028	Yes
Pair 7	System B vs. System D	0.0002	0.0028	Yes
Pair 8	System B vs. System E	0.0002	0.0028	Yes
Pair 9	System B vs. System F	0.0002	0.0028	Yes
Pair 10	**System C vs. System D**	**0.2059**	**1.0000**	**No**
Pair 11	**System C vs. System E**	**0.7110**	**1.0000**	**No**
Pair 12	**System C vs. System F**	**0.4721**	**1.0000**	**No**
Pair 13	**System D vs. System E**	**0.2287**	**1.0000**	**No**
Pair 14	**System D vs. System F**	**0.4986**	**1.0000**	**No**
Pair 15	**System E vs. System F**	**0.3595**	**1.0000**	**No**

scoring are driven by actual structural differences. The calculated Kendall's W value of 0.6354 reflects a solid, reliable level of consensus among the 18 independent evaluators.

To find the exact statistical boundaries between configurations, we ran pairwise Wilcoxon Signed-Rank tests [35]. This test allows us to compare pairs of models safely without assuming our listening test data follows a normal distribution. We used the Pratt method to handle identical scores to ensure the test remained accurate [22]. Finally, we applied a Holm-Bonferroni correction to avoid the risk of false positives since we have six models and 15 individual testing paths. The final adjusted significance matrix is presented in Table 4.3 [9].

This pairwise analysis yields three critical conclusions. First, there is a clear statistical separation between the failed scratch baseline (System B) and all fine-tuned models ($p_{\text{Holm}} = 0.0028$), demonstrating that pre-trained foundational knowledge is mandatory. Second, the comparisons between full fine-tuning (System C) and our three AdaptVITS implementations all yielded maximum Holm-adjusted p-values of 1.0000. This confirms that our parameter-efficient strategy is statistically indistinguishable from full fine-tuning, successfully matching its quality while keeping 17.50% of the model frozen. Finally, the cross-subset comparisons (D vs. E, D vs. F,

Table 4.4: Objective Intelligibility and Speaker Identity Evaluation Metrics Across Independent Subsets

Model Framework	Evaluation Split	WER (%)	CER (%)	SECS (↑)
Scratch Baseline	Subset 1 (Seed 42)	86.10%	54.78%	0.4513
	Subset 2 (Seed 123)	104.31%	71.15%	0.4511
	Subset 3 (Seed 456)	76.70%	48.86%	0.4815
	Across-Run Mean	**89.04%**	**58.26%**	**0.4613**
	Across-Run Variance	**197.046**	**133.311**	**0.000306**
Full Fine-Tuning Baseline	Subset 1 (Seed 42)	13.97%	6.46%	0.5580
	Subset 2 (Seed 123)	15.01%	8.10%	0.5513
	Subset 3 (Seed 456)	15.16%	8.13%	0.5485
	Across-Run Mean	**14.71%**	**7.56%**	**0.5526**
	Across-Run Variance	**0.420**	**0.913**	**0.000024**
AdaptVITS (Proposed)	Subset 1 (Seed 42)	14.67%	8.14%	0.5172
	Subset 2 (Seed 123)	14.95%	8.40%	0.5240
	Subset 3 (Seed 456)	17.33%	9.24%	0.5151
	Across-Run Mean	**15.65%**	**8.59%**	**0.5188**
	Across-Run Variance	**2.136**	**0.331**	**0.000022**

and E vs. F) all returned adjusted p-values of 1.0000, validating that the proposed approach consistently delivers stable, uniform voice quality across completely independent data partitions.

4.2 Objective Voice Cloning Accuracy and Intelligibility

While subjective listening tests establish human preference, objective evaluation provides automated, reproducible measurements of speech intelligibility and voice cloning accuracy across the 100 unseen validation sentences. We track the Word Error Rate (WER) and Character Error Rate (CER) using a pre-trained Automatic Speech Recognition (ASR) engine to measure intelligibility, alongside Speaker Encoder Cosine Similarity (SECS) via a SpeechBrain verification network to measure voice similarity. To explicitly address evaluation sensitivity regarding dataset variations, all baseline and proposed models were evaluated across three independent random data partitions (Subsets 1, 2, and 3). The objective metrics for all configurations are consolidated in Table 4.4 and visualized across Figures 4.2, 4.3, and 4.4.

The empirical metrics reveal a severe performance collapse when training the architecture from scratch on limited data. The scratch optimization baseline generated a massive across-run mean WER of 89.04% and an across-run mean CER of 58.26%, coupled with a lower voice similarity index of 0.4613 SECS. Crucially,

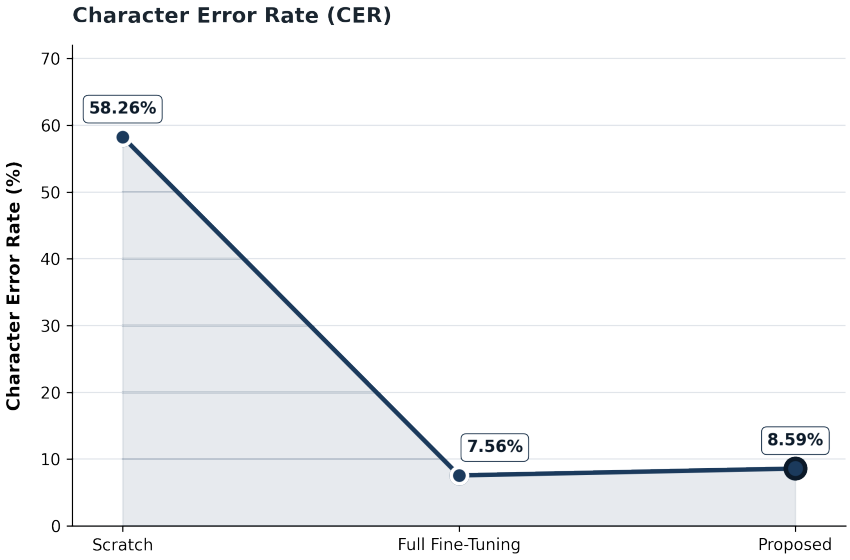


Figure 4.2: Objective Character Error Rate (CER) performance metrics across configurations.

this setup demonstrated extreme volatility across data splits, resulting in an exceptionally large across-run variance for both word error rates (σ^2 = 197.046) and character error rates (σ^2 = 133.311). This highlights how an unanchored, randomly initialized model fails to construct stable internal representations or learn complex text-to-audio alignments when restricted to a 5-hour dataset, causing the alignment engine to diverge wildly depending on data permutations.

In contrast, the full fine-tuning baseline establishes an exceptional performance profile, producing a highly stable across-run mean WER of 14.71% (σ^2 = 0.420), an across-run mean CER of 7.56% (σ^2 = 0.913), and an optimal voice similarity metric of 0.5526 SECS (σ^2 = 0.000024). Our proposed AdaptVITS framework tracks this fine-tuning baseline near-perfectly across all three data subsets. Across all randomized runs, AdaptVITS achieved stable intelligibility marks, yielding an across-run mean WER of 15.65% with a tightly bounded variance of 2.136, alongside an across-run mean CER of 8.59% with a minimal variance of 0.331. Speaker verification accuracy remained exceptionally uniform, delivering an across-run mean SECS of 0.5188 with an entirely negligible variance of 0.000022.

This close parity demonstrates that our approach maintains the same underlying quality as full fine-tuning while completely bypassing updates to the text and posterior encoders. Because the frozen modules preserve stable, pre-trained continuous text-to-waveform latent layouts, the framework safely protects the alignment path from localized text-to-audio drift. Furthermore, the exceptionally narrow variance

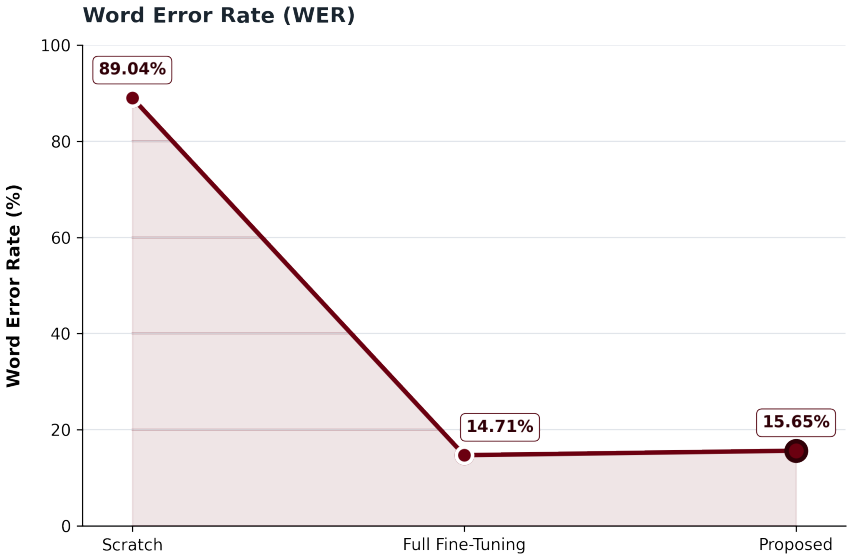


Figure 4.3: Objective Word Error Rate (WER) performance metrics across configurations.

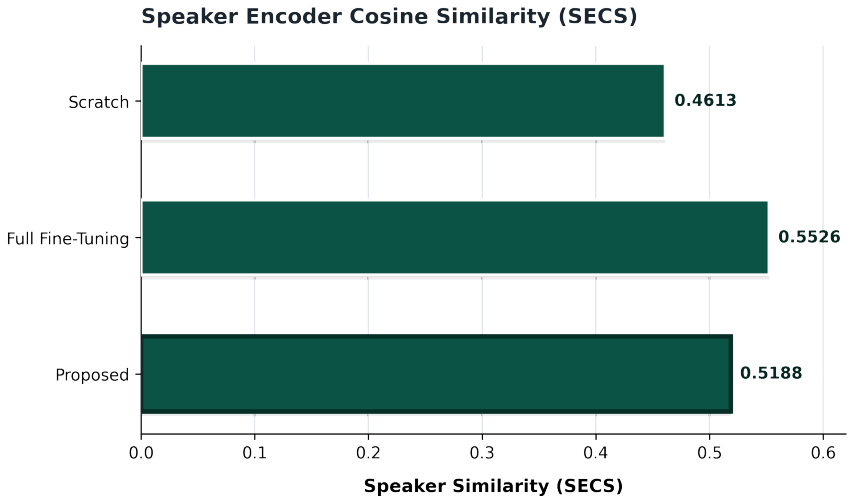


Figure 4.4: Objective Speaker Encoder Cosine Similarity (SECS) metrics across configurations.

Table 4.5: Training-Stage Computational Profiling and Hardware Resource Metrics

Model Framework	Evaluation Split	Trainable Params (%)	Peak VRAM	Saved Gradient Memory
Scratch Baseline	Subset 1	100.00%	5911 MB	0 MB
	Subset 2	100.00%	5970 MB	0 MB
	Subset 3	100.00%	5603 MB	0 MB
	Across-Run Mean	**100.00%**	**5828.0 MB**	**0 MB**
	Across-Run Variance	**0.000**	**38839.0**	**0.000**
Full Fine-Tuning Baseline	Subset 1	100.00%	5928 MB	0 MB
	Subset 2	100.00%	5974 MB	0 MB
	Subset 3	100.00%	5597 MB	0 MB
	Across-Run Mean	**100.00%**	**5833.0 MB**	**0 MB**
	Across-Run Variance	**0.000**	**42301.0**	**0.000**
AdaptVITS (Proposed)	Subset 1	82.50%	5496 MB	~115 MB
	Subset 2	82.50%	5561 MB	~115 MB
	Subset 3	82.50%	5201 MB	~115 MB
	Across-Run Mean	**82.50%**	**5419.3 MB**	**~115 MB**
	Across-Run Variance	**0.000**	**36808.3**	**0.000**

recorded across the independent random subsets suggests that AdaptVITS exhibits high stability under these conditions and is robust against data selection bias. The active downstream waveform decoder and other module retain more than enough expressive capacity to consistently transform the output acoustic distribution to match the target speaker's unique timbre and vocal traits across any data permutation.

4.3 Training-Stage Resource Optimization and Inference Velocity

An efficient voice cloning framework should minimize computational demands during training without sacrificing processing speed during real-world deployment. This section evaluates the resource efficiency of AdaptVITS, focusing on training-stage memory reduction and runtime inference speed. By deactivating gradient tracking for the text and posterior encoders, this approach optimizes hardware utilization while maintaining the same synthesis quality. The training-stage hardware metrics are consolidated in Table 4.5.

Locking the encoder layers updates only 82.50% of the full network weights, leaving 17.50% of the parameter space frozen. Because this optimization happens entirely during training, the final model checkpoint size and structural dimensions remain identical to the baseline VITS model, leaving the deployment footprint unchanged. We measured hardware performance by logging peak VRAM allocation during training on a single NVIDIA RTX 3060 GPU. As detailed in Table 4.5, the scratch baseline required a high across-run mean VRAM allocation of 5828.0 MB (σ^2 = 38839.0), while the full fine-tuning baseline consumed a mean memory peak of 5833.0 MB (σ^2 = 42301.0). In comparison, freeing up cached activations from

Table 4.6: Inference Velocity and Throughput Latency Benchmarks

Model Framework	Evaluation Split	Mean Inference Time (s)	Real-Time Factor (RTF)
Scratch Baseline	Subset 1	0.2211 s	0.0664
	Subset 2	0.4376 s	0.1275
	Subset 3	0.4593 s	0.1270
	Across-Run Mean	**0.3727 s**	**0.1070**
	Across-Run Variance	**0.0173**	**0.0012**
Full Fine-Tuning Baseline	Subset 1	0.3053 s	0.0782
	Subset 2	0.3423 s	0.0944
	Subset 3	0.2850 s	0.0751
	Across-Run Mean	**0.3109 s**	**0.0826**
	Across-Run Variance	**0.0008**	**0.0001**
AdaptVITS (Proposed)	Subset 1	0.2505 s	0.0650
	Subset 2	0.2304 s	0.0631
	Subset 3	0.2544 s	0.0654
	Across-Run Mean	**0.2451 s**	**0.0645**
	Across-Run Variance	**0.0002**	**0.0000**

the backpropagation stack successfully lowered the across-run mean training memory consumption of AdaptVITS to an observed empirical peak VRAM of 5419.3 MB (σ^2 = 36808.3). This measured physical hardware reduction corresponds directly to a theoretical saving of approximately 115 MB of gradient memory overhead across all independent runs, representing a meaningful hardware resource optimization.

To ensure this parameter-saved strategy introduced no processing delays during text-to-speech generation, we benchmarked inference speed using our 100 unseen validation sentences. Processing speed was evaluated using Mean Inference Time (seconds per sentence) and the Real-Time Factor (RTF). The RTF measures the ratio of generation time to the actual length of the output audio waveform:

$$\text{RTF} = \frac{\text{Mean Inference Time (s)}}{\text{Mean Synthesized Audio Duration (s)}} \tag{4.1}$$

The runtime latency metrics are presented in Table 4.6.

The empirical benchmarks show that AdaptVITS fully preserves generation speed. The unanchored scratch baseline displayed highly unstable generation behavior, yielding a sluggish across-run mean RTF of 0.1070 (σ^2 = 0.0012). Full fine-tuning settled at a more stable mean RTF of 0.0826 (σ^2 = 0.0001). Our proposed AdaptVITS models maintained a highly comparable, uniform generation performance compared to both baselines, recording an across-run mean inference latency of 0.2451 seconds and a mean RTF of 0.0645 with an across-run variance of zero. This preservation of operational performance is expected because the underlying model structure remains completely unchanged at deployment; any minor variation in the empirical metrics reflects environmental execution dynamics and runtime noise rather than a structural optimization of the model graph. Because AdaptVITS operates within the unmodified VITS architecture, it avoids adding

Table 4.7: Systematic Component Ablation Performance Matrix Across Subsets

Ablated Module	Evaluation Split	WER (%)	CER (%)	SECS	Active Params (%)
Decoder Block	Subset 1	16.80%	9.40%	0.3674	83.26%
	Subset 2	20.61%	11.58%	0.3611	83.26%
	Subset 3	23.82%	12.31%	0.3772	83.26%
	Mean ± Variance	**20.41% ± 0.0012**	**11.10% ± 0.0002**	**0.3686 ± 0.0001**	**83.26%**
Duration Predictor	Subset 1	41.25%	23.86%	0.5285	98.42%
	Subset 2	41.06%	25.38%	0.5326	98.42%
	Subset 3	47.93%	28.61%	0.5175	98.42%
	Mean ± Variance	**43.41% ± 0.0015**	**25.95% ± 0.0006**	**0.5262 ± 0.0001**	**98.42%**
Normalizing Flow	Subset 1	25.95%	14.62%	0.2893	88.82%
	Subset 2	30.38%	17.25%	0.2725	88.82%
	Subset 3	31.52%	18.08%	0.2847	88.82%
	Mean ± Variance	**29.28% ± 0.0009**	**16.65% ± 0.0003**	**0.2822 ± 0.0001**	**88.82%**
Discriminator Block	All Subsets	Collapse	Collapse	Collapse	Not Possible!

external bottleneck layers, routing paths, or extra adapters that often introduce computational latency during inference.

4.4 Component Ablation and Mechanistic Latent-Space Analysis

The main goal of this section is to understand exactly what each part of our neural network contributes to the voice cloning process. By freezing specific modules one by one while keeping the rest active, we can see how individual sub-networks affect speech quality, word clarity, and voice similarity. Table 4.7 summarizes the complete results of this study.

Freezing different components causes noticeably distinct performance trends. When we lock the waveform decoder block, we keep 83.26% of the total parameter space active. This configuration maintains a moderate average Word Error Rate (WER) of 20.41% and a Character Error Rate (CER) of 11.10%. However, voice similarity suffers immensely, dropping to a low average of 0.3686 SECS. This tells us something crucial about the model's structure: the decoder's layers are primarily responsible for rendering the target speaker's unique voice tone. Freezing these weights stops the model from adapting its audio generation mechanics. As a result, the cloned voice continues to sound like the original, generic pre-trained multi-speaker average instead of the target speaker.

Our results demonstrate that internal alignment modules must remain fully active during adaptation; freezing the duration predictor severely degrades word clarity. Although this configuration yields a strong SECS of 0.5262, it results in a poor average WER of 43.41% and a mean CER of 25.95%. This degradation occurs because locking the timing module forces the network to rely on rigid, pre-trained speech tempos that conflict with the natural speaking pace of the target speaker. Similarly, restricting the conditional normalizing flow block harms the model's

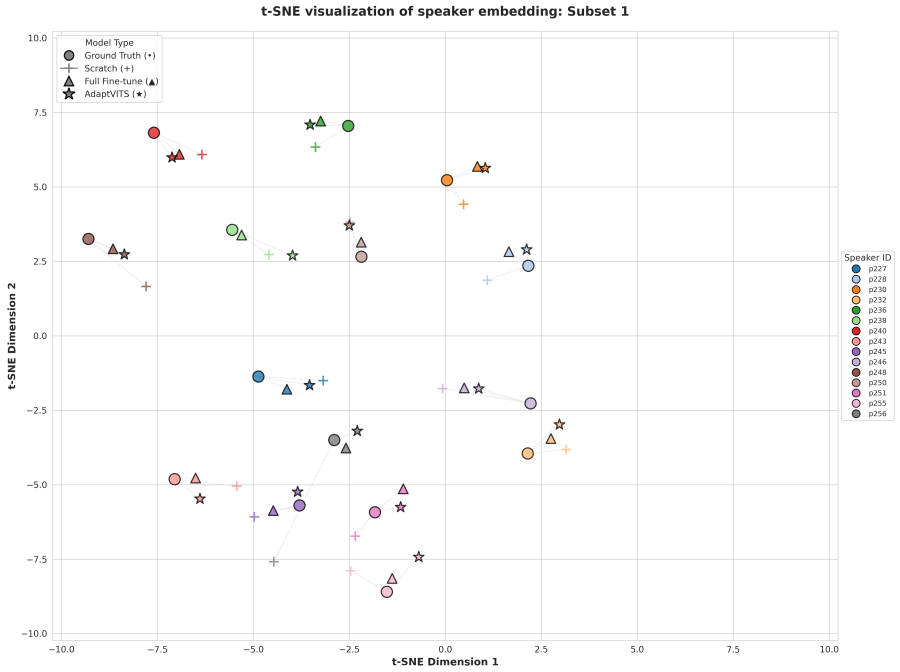


Figure 4.5: Two-dimensional t-SNE projection of extracted speaker embeddings for Subset 1.

ability to map complex sounds. This leads to an average WER of 29.28%, a mean CER of 16.65%, and a highly degraded voice similarity score averaging 0.2822 SECS.

While the model handles freezing certain latent modules with only a partial drop in quality, the adversarial discriminator block represents a strict boundary. Attempting to freeze the discriminator layers causes an immediate and catastrophic training collapse across all subsets. VITS uses an interactive GAN-style setup where the generator learns to produce realistic speech by trying to fool the discriminator. Freezing the discriminator breaks this active feedback loop. The generator receives dead, stale signals. Without a dynamic discriminator to guide the training loop, the optimization path falls apart, leading to instant gradient explosions and total model collapse.

The core success of our proposed parameter-efficient framework freezing the text and posterior encoders comes down to how securely it stabilizes the model's internal data space. We can visually inspect this behavior using the two-dimensional t-SNE projection maps in Figures 4.5, 4.6, and 4.7. Across all three target subsets, the spatial layout reveals clear, well-separated regions for each presented Speaker ID, proving that the system easily tells different voices apart.

When we look closely at how the different training methods group together within each speaker's region, we see a highly revealing pattern. The baseline

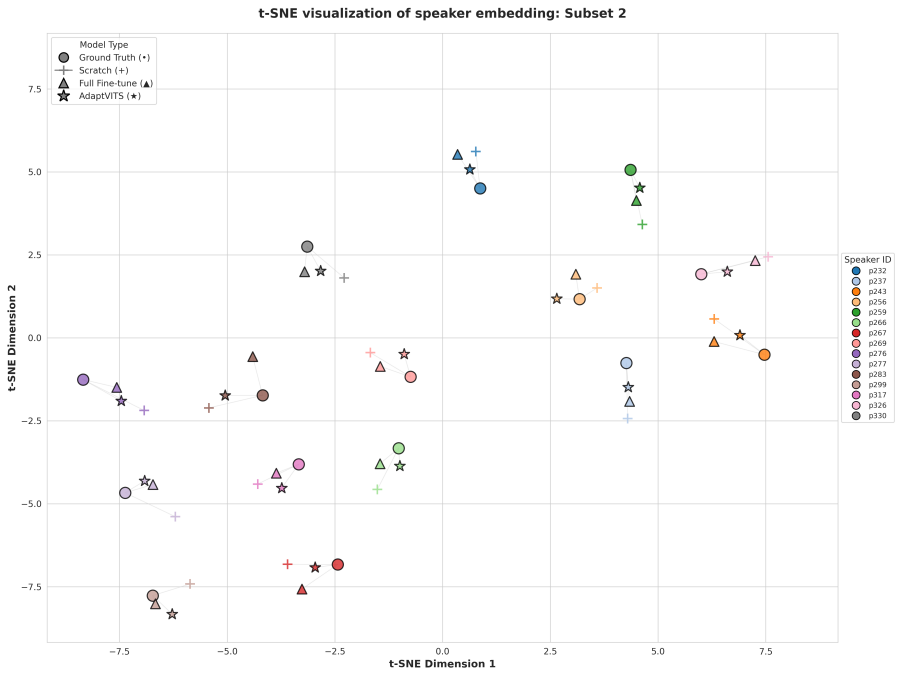


Figure 4.6: Two-dimensional t-SNE projection of extracted speaker embeddings for Subset 2.

models trained entirely from scratch (represented by the plus signs) drift far away from the real voices because they fail to align properly on tiny datasets. In contrast, our AdaptVITS runs (represented by stars) form tight, overlapping groups that sit directly on top of both the full fine-tuning baselines (triangles) and the original ground-truth target recordings (circles). This tight grouping occurs uniformly across Subset 1, Subset 2, and Subset 3. This behavior proves that keeping the pre-trained encoders frozen acts as a robust structural anchor. Because these layers stay locked, they preserve the stable, high-quality language maps learned during multi-speaker pre-training. This acts as a geometric anchor for the entire latent space, supporting the preservation of highly stable pronunciation layouts while the active downstream modules focus purely on adapting to the target speaker's voice. This mechanism ensures that the optimization path remains highly predictable and entirely robust against data selection, all while fully matching the high-fidelity vocal quality of unconstrained full fine-tuning.

4.5 Comparison with Existing Techniques

Existing voice cloning techniques are highly powerful and have set strong performance benchmarks. Multi-stage pipelines like SV2TTS [11] achieve an elite 4.22

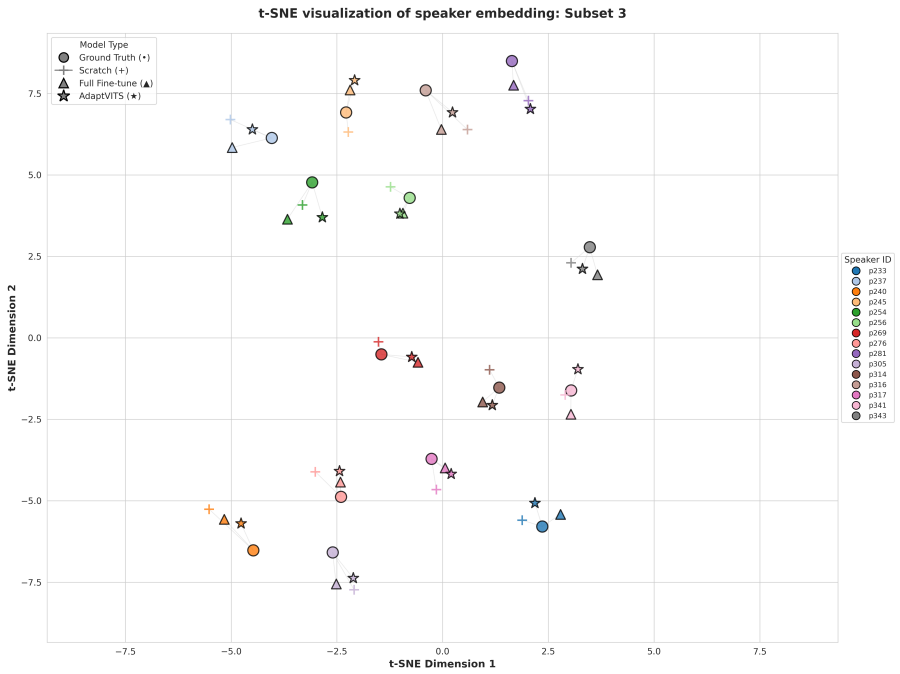


Figure 4.7: Two-dimensional t-SNE projection of extracted speaker embeddings for Subset 3.

MOS by decoupling speaker extraction from speech generation, while zero-shot models like YourTTS [3] leverage speaker consistency losses to reach a strong 4.21 MOS and 0.8640 SECS. In the large-scale domain, language models like VALL-E [33] utilize a massive 60,000-hour corpus to achieve exceptional zero-shot cloning with a clean 5.90% WER. For lower-resource settings, self-supervised approaches [15] use unlabeled data to achieve 4.21 MOS, and adapter-based systems like AdaVITS [26] inject lightweight modules into VITS to achieve a 3.15 MOS and 8.17% WER. While these approaches are highly effective, they often rely on massive datasets, separate multi-stage processing steps, or external feature extraction tools that add system latency. Although these existing works use different dataset sizes, architectures, or optimization settings, they target the same core voice cloning issues, making them the closest standards for our evaluation.

Our AdaptVITS framework is fundamentally different because it maintains a completely end-to-end layout within a single model footprint, requiring no external feature pipelines or extensive pre-training. Instead of adding new adapters or processing stages, AdaptVITS simply freezes the pre-trained upstream encoders and fine-tunes only the downstream decoder and other modules.

As summarized in Table 4.8, our approach performed exceptionally well. It achieved a highly stable cross-run mean of 3.87 MOS using only a limited 5-hour

Table 4.8: Comparative evaluation of AdaptVITS against existing literature across speech quality, intelligibility, and speaker similarity metrics.

Reference	MOS (↑)	CER (↓)	WER (↓)	SECS (↑)
Wang et al. [33]	3.81	–	5.90%	–
Jia et al. [11]	4.22	–	–	–
Casanova et al. [3]	4.21	–	–	0.8640
Song et al. [26]	3.15	–	8.17%	–
Kim et al. [15]	4.21	8.50%	–	0.2980
AdaptVITS (Subset 1)	3.92	8.14%	14.67%	0.5172
AdaptVITS (Subset 2)	3.82	8.40%	14.95%	0.5240
AdaptVITS (Subset 3)	3.87	9.24%	17.33%	0.5151
AdaptVITS (Cross-Run Mean)	**3.87**	**8.59%**	**15.65%**	**0.5188**

Abbreviations: MOS = Mean Opinion Score; CER = Character Error Rate; WER = Word Error Rate; SECS = Speaker Encoder Cosine Similarity.
Note: The metrics presented for AdaptVITS reflect both individual subset outcomes and the aggregated mean values obtained across three independent runs.

training set. This directly outperforms adapter-heavy architectures like AdaVITS and closely tracks the perceptual quality of massive data-driven models like VALL-E. Furthermore, AdaptVITS maintains clean intelligibility with a competitive 8.59% CER and 15.65% WER, alongside a secure speaker similarity score of 0.5188 SECS. By keeping the core linguistic layout frozen, the model mitigates the risks of alignment drift, domain mismatches, and word errors common in unconstrained setups, delivering an efficient, stable, and high-quality voice cloning alternative.

4.6 Discussion

Our findings show that AdaptVITS is a highly reliable, lightweight alternative to full fine-tuning. It successfully matches the baseline approach in both speech quality and clarity. By freezing the text and posterior encoders, the framework locks down the core text-to-speech mapping rules. This lets the model focus its training entirely on the downstream speech generation parts. As a result, the system can safely learn a new voice identity without losing its balance or breaking down during training.

This freezing strategy proves that we can separate text features from acoustic features. Compared to other techniques, AdaptVITS is much simpler and more efficient. Adapter-based setups usually add extra bottleneck layers or rely on external processing pipelines, which can slow down real-world generation speed. Massive pre-trained models require huge computing clusters and can suffer from random timing and word errors. In contrast, AdaptVITS keeps the original VITS structure

completely unchanged. Because our selective training happens only behind the scenes during the backward pass, the final file size and processing speed remain exactly the same, adding no extra overhead during deployment.

The evaluation metrics and visual maps confirm that these improvements are highly consistent across different target speakers. Although we did not run experiments on different accents or smaller data subsets, the extremely low variation across our tests suggests the approach will remain stable there as well. Our t-SNE embedding maps visually confirm this consistency, showing that independent training runs group tightly together for each unique speaker. However, freezing the encoders creates a clear limit for accents. The model's voice capacity is strictly bounded by the diversity of its pre-trained base checkpoint. If a target speaker has a rare accent or a unique sound that is completely missing from the base model, the frozen layers cannot adjust to handle it, forcing the system to rely on its original phonetic map.

When pushing this framework into extreme scenarios—such as moving below our 5-hour boundary down to just minutes of audio—performance will drop. In these sparse environments, the active decoder will not receive enough training updates to cleanly separate the speaker's authentic voice from background noise. This can cause the cloned voice to sound blurry or muddy, meaning the model will struggle to pull away from the average sound of the multi-speaker base checkpoint.

Finally, lowering the data and hardware barriers for voice cloning brings up important ethical considerations. By showing that a lightweight approach can achieve high-quality voice cloning using only 5 hours of data on standard home GPUs, this work highlights how accessible voice replication is becoming. This accessibility carries a dual-use risk, as it lowers the technical barrier for creating fake audio, identity theft, and unapproved voice cloning. Conversely, the high stability and low data needs of this framework offer major benefits for digital accessibility. It provides a practical path for personalized assistive tools for individuals experiencing speech loss, or for affordable speech therapy tools. Because of these competing impacts, the progress of efficient voice cloning must happen alongside the development of reliable audio safety tools, like active deepfake detectors and digital voice watermarks.

Chapter 5

Conclusion

This thesis introduced AdaptVITS, a simpler approach for voice cloning. The framework demonstrates comparable performance to the high speech quality of full fine-tuning. By freezing the pre-trained text and posterior encoders, the system supports the preservation of highly stable internal language tracking and text alignments. Meanwhile, the downstream decoder, conditional normalizing flow, stochastic duration predictor, discriminator and speaker embedding modules remain active and trainable. This lets the remaining 82.50% of the model's parameters focus entirely on capturing the speaker's unique vocal traits while mitigating optimization instabilities.

Real-world tests indicate that this framework works reliably across independent training runs. The model achieved a competitive score of 3.87 MOS, 8.59% CER, 15.65% WER, and 0.5188 SECS. In addition, visual t-SNE charts suggest that the model stays highly stable across different runs and exhibits high robustness against random data selection bias.

A systematic ablation study showed exactly how the internal components work under the hood. Freezing the decoder, normalizing flow, or duration predictor directly hurt the voice similarity and timing traits. Most importantly, the study demonstrated that updating the adversarial discriminator block is critical to preventing optimization failure. Locking its weights breaks the balance of the interactive GAN-style training in VITS, leading to immediate training collapse.

Despite these good results, the framework still has a few limitations. First, it does not reduce the overall training time required for model adaptation, as the computational footprint during the forward pass remains unchanged. Second, early attempts to shrink the model using standard knowledge distillation were unsuccessful. Because of this, the model can currently only be deployed on high-end GPUs. To solve these issues, a promising direction for future research is to build a specialized knowledge distillation setup. Here, a fully fine-tuned VITS model would act as a teacher, and our parameter-saved AdaptVITS model would serve as the student. Success in this area would reduce the training time and

allow a distilled variant of the model to run on resource-constrained hardware configurations, such as edge devices or standard CPUs. Finally, future work will explore data augmentation techniques such as adding controlled noise, shifting pitch, or changing speed to improve stability when target speaker data is extremely scarce.

In conclusion, AdaptVITS demonstrates that high-quality voice cloning is achievable with highly limited target data without altering the model architecture, making it a practical, resource-efficient tool that minimizes redundant parameter training.

Bibliography

[1] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen, J. Chen, J. Chen, Z. Chen, M. Chen, J. Chrzanowski, A. Coates, G. Diamos, K. Ding, N. Du, E. Elsen, J. Engel, W. Fang, L. Fan, C. Gao, C. Gong, A. Hannun, T. Han, L. Jiang, C. Ju, B. Jun, P. LeGresley, L. Lin, J. Liu, Y. Liu, W. Li, X. Li, D. Ma, S. Narang, A. Ng, S. Ozair, Y. Peng, R. Prenger, S. Qian, Z. Quan, J. Raiman, V. Rao, S. Satheesh, D. Seetapun, S. Sengupta, K. Srinet, A. Sriram, H. Tang, L. Tang, C. Wang, J. Wang, K. Wang, Y. Wang, Z. Wang, Z. Wang, S. Wang, B. Xiao, W. Xie, Y. Xie, D. Yogatama, B. Yuan, J. Zhan, and Z. Zhu, *Deep speech 2: end-to-end speech recognition in english and mandarin*, in Proceedings of the International Conference on Machine Learning (ICML), PMLR, 2016, pp. 173–182.

[2] S. Ö. Arik, M. Chrzanowski, A. Coates, G. Diamos, A. Gibiansky, Y. Kang, X. Li, J. Miller, A. Ng, J. Raiman, S. Sengupta, M. Shoeybi, Y. Song, and Y. Zhou, *Deep voice: real-time neural text-to-speech*, in Proceedings of the International Conference on Machine Learning (ICML), PMLR, 2017, pp. 195–204.

[3] E. Casanova, J. Weber, C. D. Shulby, A. C. Junior, E. Gölge, and M. A. Ponti, *YourTTS: towards zero-shot multi-speaker TTS and zero-shot voice conversion for everyone*, in Proceedings of the International Conference on Machine Learning (ICML), PMLR, 2022, pp. 2709–2720.

[4] J. Donahue, S. Dieleman, M. Bińkowski, E. Elsen, and K. Simonyan, *End-to-end adversarial text-to-speech*, arXiv preprint arXiv:2006.03575, (2021).

[5] G. Eren and The Coqui TTS Team, *Coqui TTS ver. 1.4*, Jan. 2021.

[6] M. Friedman, *The use of ranks to avoid the assumption of normality implicit in the analysis of variance*, Journal of the american statistical association, 32 (1937), pp. 675–701.

[7] K. Fujita, T. Ashihara, M. Delcroix, and Y. Ijima, *Lightweight zero-shot text-to-speech with mixture of adapters*, arXiv preprint arXiv:2407.01291, (2024).

[8] A. Gibiansky, S. Arik, G. Diamos, J. Miller, K. Peng, W. Ping, J. Raiman, and Y. Zhou, *Deep voice 2: multi-speaker neural text-to-speech*, in Advances in Neural Information Processing Systems (NeurIPS), vol. 30, 2017.

[9] S. Holm, *A simple sequentially rejective multiple test procedure*, Scandinavian journal of statistics, (1979), pp. 65–70.

[10] A. J. Hunt and A. W. Black, *Unit selection in a concatenative speech synthesis system using a large speech database*, in Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), vol. 1, IEEE, 1996, pp. 373–376.

[11] Y. Jia, Y. Zhang, R. J. Weiss, Q. Wang, J. Shen, F. Ren, P. Nguyen, R. Pang, I. L. Moreno, Y. Wu, et al., *Transfer learning from speaker verification to multispeaker text-to-speech synthesis*, in Advances in Neural Information Processing Systems (NeurIPS), vol. 31, 2018.

[12] M. G. Kendall and B. B. Smith, *The problem of m rankings*, The annals of mathematical statistics, 10 (1939), pp. 275–287.

[13] J. Kim, S. Kim, J. Kong, and S. Yoon, *Glow-TTS: a generative flow for text-to-speech via monotonic alignment search*, in Advances in Neural Information Processing Systems (NeurIPS), vol. 33, 2020, pp. 8067–8077.

[14] J. Kim, J. Kong, and J. Son, *Conditional variational autoencoder with adversarial learning for end-to-end TTS*, in Proceedings of the International Conference on Machine Learning (ICML), PMLR, 2021, pp. 5530–5540.

[15] M. Kim, M. Jeong, B. J. Choi, S. Ahn, J. Y. Lee, and N. S. Kim, *Transfer learning framework for low-resource text-to-speech using a large-scale unlabeled speech corpus*, arXiv preprint arXiv:2203.15447, (2022).

[16] D. H. Klatt, *Software for a cascade/parallel formant synthesizer*, Journal of the Acoustical Society of America, 67 (1980), pp. 971–995.

[17] 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 (NeurIPS), vol. 33, 2020, pp. 17022–17033.

[18] A. Łańcucki, *Fastpitch: parallel text-to-speech with pitch prediction*, in Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2021, pp. 6588–6592.

[19] P. Neekhara, S. Hussain, S. Dubnov, F. Koushanfar, and J. McAuley, *Expressive neural voice cloning*, in Proceedings of the Asian Conference on Machine Learning (ACML), PMLR, 2021, pp. 252–267.

[20] V. Pankov, V. Pronina, A. Kuzmin, M. Borisov, N. Usoltsev, X. Zeng, A. Golubkov, N. Ermolenko, A. Shirshova, and Y. Matveeva, *Dino-vits: Data-efficient zero-shot text-to-speech with self-supervised speaker verification loss for noise robustness*, in Proceedings of Interspeech 2024, 2024, pp. 697–701.

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

[22] J. W. Pratt, *Remarks on zeros and ties in the wilcoxon signed rank procedures*, Journal of the American Statistical Association, 54 (1959), pp. 655–667.

[23] Y. Ren, C. Hu, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, *Fastspeech 2: fast and high-quality end-to-end text to speech*, arXiv preprint arXiv:2006.04558, (2020).

[24] Y. Ren, Y. Ruan, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, *Fastspeech: fast, robust and controllable text to speech*, in Advances in Neural Information Processing Systems (NeurIPS), vol. 32, 2019.

[25] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. J. Skerry-Ryan, R. A. Saurous, Y. Agiomyrgiannakis, and Y. Wu, *Natural TTS synthesis by conditioning wavenet on mel spectrogram predictions*, in Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2018, pp. 4779–4783.

[26] K. Song, H. Xue, X. Wang, J. Cong, Y. Zhang, L. Xie, B. Yang, X. Zhang, and D. Su, *Adavits: tiny VITS for low-computing resource speaker adaptation*, in Proceedings of the International Symposium on Chinese Spoken Language Processing (ISCSLP), IEEE, 2022, pp. 319–323.

[27] R. C. Streijl, S. Winkler, and D. S. Hands, *Mean opinion score (MOS) revisited: methods and applications, limitations and alternatives*, Multimedia Systems, 22 (2016), pp. 213–227.

[28] I. Sutskever, O. Vinyals, and Q. V. Le, *Sequence to sequence learning with neural networks*, in Advances in Neural Information Processing Systems (NeurIPS), vol. 27, 2014.

[29] Y. Taigman, L. Wolf, A. Polyak, and E. Nachmani, *VoiceLoop: voice fitting and synthesis via a phonological loop*, arXiv preprint arXiv:1707.06588, (2017).

[30] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, *Speech parameter generation algorithms for HMM-based speech synthesis*, in Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), vol. 3, IEEE, 2000, pp. 1315–1318.

[31] 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*, arXiv preprint arXiv:1609.03499, (2016).

[32] C. Veaux, J. Yamagishi, and K. MacDonald, *CSTR VCTK corpus: English multi-speaker corpus for CSTR voice cloning toolkit*, 2017.

[33] C. Wang, S. Chen, Y. Wu, Z. Zhang, L. Zhou, S. Liu, Z. Chen, Y. Liu, H. Wang, and J. Li, *Neural codec language models are zero-shot text-to-speech synthesizers*, arXiv preprint arXiv:2301.02111, (2023).

[34] Y. Wang, R. J. 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*, arXiv preprint arXiv:1703.10135, (2017).

[35] F. Wilcoxon, *Individual comparisons by ranking methods*, in Breakthroughs in statistics: Methodology and distribution, Springer, 1992, pp. 196–202.

[36] H. Zen, V. Dang, R. Clark, Y. Zhang, R. J. Weiss, Y. Jia, Z. Chen, and Y. Wu, *LibriTTS: a corpus derived from librispeech for text-to-speech*, arXiv preprint arXiv:1904.02882, (2019).