StyleTTS-ZS: Efficient High-Quality Zero-Shot Text-to-Speech Synthesis with Distilled Time-Varying Style Diffusion

Yinghao Aaron Li* **, Xilin Jiang, Cong Han, Nima Mesgarani,**
Columbia University
Correspondence: yl4579@columbia.edu

Abstract

The rapid development of large-scale text-to-speech (TTS) models has led to significant advancements in modeling diverse speaker prosody and voices. However, these models often face issues such as slow inference speeds, reliance on complex pre-trained neural codec representations, and difficulties in achieving naturalness and high similarity to reference speakers. To address these challenges, this work introduces StyleTTS-ZS, an efficient zero-shot TTS model that leverages distilled time-varying style diffusion to capture diverse speaker identities and prosodies. We propose a novel approach that represents human speech using input text and fixed-length time-varying discrete style codes to capture diverse prosodic variations, trained adversarially with multi-modal discriminators. A diffusion model is then built to sample this time-varying style code for efficient latent diffusion. Using classifier-free guidance, StyleTTS-ZS achieves high similarity to the reference speaker in the style diffusion process. Furthermore, to expedite sampling, the style diffusion model is distilled with perceptual loss using only 10k samples, maintaining speech quality and similarity while reducing inference speed by 90%. Our model surpasses previous state-of-the-art large-scale zero-shot TTS models in both naturalness and similarity, offering a 10-20× faster sampling speed, making it an attractive alternative for efficient large-scale zero-shot TTS systems. The audio demo, code and models are available at `https://styletts-zs.github.io/`.

1 Introduction

Recent advancements in text-to-speech (TTS) technology have achieved remarkable progress, bringing TTS systems close to, or even surpassing, human-level performance on various benchmark datasets (Tan et al., 2024; Shen et al., 2024; Li et al., 2024a). With studio-level TTS capabilities nearly perfected, there is a growing demand for more sophisticated tasks such as diverse and personalizable zero-shot speaker adaptation (Casanova et al., 2022). These tasks present a significant challenge due to the need to replicate the unique characteristics and prosodic variations of a vast array of speakers without extensive training data for each individual. Although there have been rapid developments in zero-shot adaptation, driven by large-scale modeling techniques in large language models (LLMs) (Jiang et al., 2023b; Wang et al., 2023a; Peng et al., 2024; Kim et al., 2024a; Chen et al., 2024a), high-quality discrete audio codecs (Zeghidour et al., 2021; Défossez et al., 2022; Kumar et al., 2024), and diffusion-based models (Shen et al., 2024; Ju et al., 2024; Le et al., 2024; Lee et al., 2024; Yang et al., 2024), current models face crucial limitations. Many large-scale speech synthesis models rely on auto-regressive modeling (Jiang et al., 2023b; Wang et al., 2023a,c; Jiang et al., 2023a; Peng et al., 2024; Kim et al., 2024a; Chen et al., 2024a; Meng et al., 2024), which results in slower scaling of inference speed as the length of the target speech increases. Alternatively, diffusion-based models are used for building large-scale speech synthesis models (Shen et al., 2024; Le et al., 2024; Ju et al., 2024; Lee et al., 2024; Yang et al., 2024; Eskimez et al., 2024). However, since these models require iterative refinement to produce high-quality results, they also suffer from efficiency issues. Moreover, these models often depend on pre-trained neural codecs not specifically designed for TTS tasks (Wang et al., 2023a,c; Shen et al., 2024), limiting their ability to naturally model diverse human speech, which encompasses a wide range of speaking styles that can be challenging to control with existing codecs.

In this work, we introduce StyleTTS-ZS, an innovative approach to diverse speech synthesis that aims to address these limitations. Our model decomposes human speech into a global style vector

*Work done in part during an internship at Descript.

Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 4725–4744
April 29 - May 4, 2025

derived from a speaker prompt and prompt-aligned text embeddings that encapsulate the timbre and acoustic features of the speech. Additionally, it includes a fixed-length time-varying style vector that encodes the diverse prosodic variations, such as pitch and duration changes over time. By carefully designing the bottleneck for the style vector space with vector quantization (Van Den Oord et al., 2017) and multimodal adversarial training (Janiczek et al., 2024), we can reconstruct speech with high fidelity. We then train a diffusion model (Ho et al., 2020) to sample the time-varying style vector, effectively modeling the diversity of prosodic variations conditioned on the speaker prompt. This results in efficient latent diffusion, as the latent variable is a fixed-length style vector. The simplicity and efficiency of our latent space enable us to distill the teacher diffusion model into a student model using only 10k samples. This distillation maintains diversity and similarity to the prompt speaker while reducing inference to one step. Our evaluation results demonstrate the effectiveness of StyleTTS-ZS. When trained on the small-scale LibriTTS dataset (Zen et al., 2019), our model surpasses several public zero-shot TTS baseline models. Furthermore, when trained on the large-scale LibriLight dataset (Kahn et al., 2020), comprising 60k hours of data, our model performs comparably to previous large-scale state-of-the-art (SOTA) TTS models in similarity and even surpasses them in naturalness for unseen speakers on the LibriSpeech dataset using only a 3-second reference speaker prompt. Remarkably, we achieve this with nearly 10-20 times faster inference speeds compared to previous SOTA models, showcasing its real-time applicability.

2 Related Work

Zero-Shot TTS Synthesis. Zero-shot TTS synthesizes speech in unseen voices using reference speech, offering adaptability without extra training. Traditional models train on small datasets, utilizing pre-trained speaker embeddings or speaker encoders (Casanova et al., 2022, 2021; Wu et al., 2022; Lee et al., 2022; Li et al., 2024a; Min et al., 2021; Li et al., 2022; Choi et al., 2022). Recent large-scale methods focus on in-context learning with reference prompts (Wang et al., 2023a), employing either autoregressive models like large language models to predict speech tokens (Shen et al., 2024; Le et al., 2024; Ju et al., 2024; Lee et al., 2024; Yang et al., 2024; Eskimez et al., 2024) or non-autoregressive diffusion techniques for higher-quality speech (Jiang et al., 2023b; Wang et al., 2023a,c; Jiang et al., 2023a; Peng et al., 2024; Kim et al., 2024a; Chen et al., 2024a; Meng et al., 2024; Yang et al., 2024; Lee et al., 2024). Our method combines encoder techniques with in-context learning by modeling speech as prompt-aligned text embeddings while using diffusion-based models to predict the global prosody.

Efficient High-Quality Speech Synthesis. Autoregressive models produce diverse speech but suffer from slow inference (Wang et al., 2023a; Song et al., 2024). Non-autoregressive models (Ren et al., 2020) are faster but often miss fine details. Adversarial training (Kim et al., 2021) and diffusion models (Popov et al., 2021) enhance quality but add inference time. Speed-ups, like diffusion model distillation (Huang et al., 2022b; Ye et al., 2023, 2024; Guan et al., 2024), sacrifice quality due to trade-offs. StyleTTS-ZS minimizes these issues by focusing diffusion modeling only on prosody, reducing the diffusion model's burden. It distills the model in one step, outperforming a very recent efficient TTS models like FlashSpeech (Ye et al., 2024) while keeping similar speed.

3 StyleTTS-ZS

StyleTTS-ZS consists of four modules: acoustic synthesizer, prosody autoencoder, time-varying style diffusion, and multimodal discriminators. We detail these four modules in the following sections with an overview of our framework in Figure 1 and implementation details in Appendix D and C.

3.1 Acoustic Synthesizer

The role of the acoustic synthesizer is to reconstruct input speech $\boldsymbol{x}$ using its text transcription $\boldsymbol{t}$ and a speech prompt $\boldsymbol{x}'$ from the same speaker into the reconstructed speech $\hat{\boldsymbol{x}}$. This process starts with extracting pitch p, energy n, and duration d from the input speech $\boldsymbol{x}$. The joint prompt-text encoder T then encodes the phoneme text $\boldsymbol{t}$ and the prompt speech $\boldsymbol{x}'$ into prompt-aligned text embeddings $\boldsymbol{h}_{\text{text}} = T(\boldsymbol{t}, \boldsymbol{x}')$ and a global style $\boldsymbol{s}$. The speech is then reconstructed as $\hat{\boldsymbol{x}} = G(\boldsymbol{h}_{\text{text}}, p, n, d, \boldsymbol{s})$.

We use the same pitch extractor F, duration extractor A, and decoder G as in (Li et al., 2024a). Unlike previous works that rely solely on global speaker embeddings or style vectors (Min et al., 2021; Casanova et al., 2022; Li et al., 2022, 2024a) or prompt-aligned text embeddings (Huang et al.,

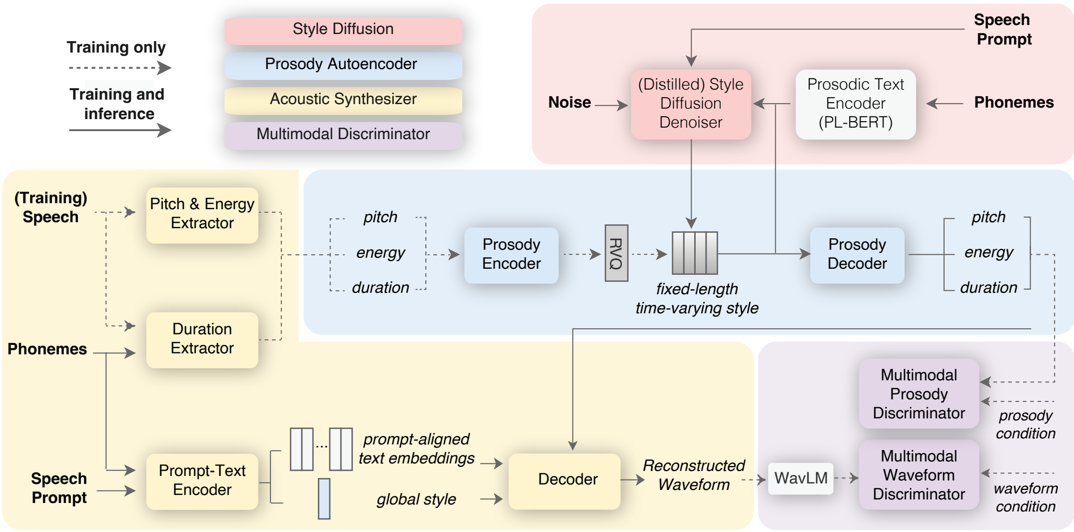


Figure 1: Overview of StyleTTS-ZS architecture. During training, the model uses ground truth speech to extract prosodic features and encode text and style with prompt speech. The prosody encoder compresses these features into a fixed-length time-varying style vector, which is regularized and decoded back by the prosody decoder. The style diffusion denoiser uses this vector for diffusion model training, and the decoder reconstructs speech using prosodic features, text embeddings, and global style, with multimodal discriminators assessing the output. Bold indicates system input, where speech prompts and phonemes are used for both style diffusion and acoustic synthesizer.

2022a; Kim et al., 2024b), we combine both approaches and jointly encode the speech prompt and text to produce both prompt-aligned text embeddings and a style vector (Figure 2c). Our prompt-text encoder is similar to (Kim et al., 2024b) but uses conformer blocks (Gulati et al., 2020) instead of transformer blocks (Vaswani et al., 2017) to better model the speech. Moreover, instead of discarding the output portion from the speech input, we apply average pooling and use the pooled results as the global style. Utilizing both prompt-aligned text embeddings and global style vectors enhances speaker similarity, as demonstrated in Section 4.3.

When training on large-scale datasets with thousands of speakers, we observed that the mel-spectrogram reconstruction loss alone was insufficient for achieving high-fidelity voice similarity due to the limited capacity of the acoustic synthesizer, which does not use transformers to optimize the inference speed. To address this, we introduced an additional reconstruction loss that aligns with the intermediate features of speaker embedding models (Wang et al., 2023b). Since our model operates directly in the waveform domain and generates waveforms end-to-end without relying on any codec decoder or vocoder, we can compute and match these intermediate features directly. This significantly enhances speaker similarity, as shown in Section 4.3. Moreover, the AdaIN-based decoder from (Li et al., 2024a) is easier to train than the attention-based module in the prompt-text encoder, causing the model to over-rely on the global style vector. Consequently, the prompt-aligned text embeddings become too similar to plain text embeddings. To mitigate this, we apply a 20% dropout to the global style vector during training, which compels the prompt-text encoder to focus more on aligning text with the prompt and producing a global style. This strategy enhances reconstruction quality by ensuring the prompt-text encoder actively contributes to acoustic synthesis.

3.2 Vector Quantized Prosody Autoencoder

While the acoustic synthesizer can reconstruct speech from prompt-aligned text embeddings, phoneme duration, pitch, energy, and a global style vector with high fidelity, we lack ground truth for these prosodic features during inference. Predicting these features from text alone is challenging due to their variability and diversity, especially in large-scale datasets with various speakers. Prosodic features are crucial for both speech naturalness and speaker similarity; unnatural prosody can make speech sound robotic, while uncharacteristic prosody produces dissimilar voices despite having the same timbre as the prompt speaker. Recent works in large-scale TTS model this variability on large datasets using methods from large language models (LLMs) (Jiang et al., 2023b,a) and large diffusion models (Ju et al., 2024). However, these methods are inefficient as the computation required to infer prosody is proportional to the length of the target speech. We take an innovative approach to compressing these prosodic features into a fixed-length time-varying style vector via a prosody autoencoder, allowing efficient diffusion sampling for diverse prosody of human speech.

The prosody encoder (Figure 2a) processes stacked pitch, energy, and duration inputs, mapping them into a fixed-length vector. We represent duration using upsampled positional embeddings PE$(\cdot)$, where for each $i \in \{1, \ldots, N\}$, the positional embedding PE(i) is repeated d_i times. This representation efficiently distinguishes each phoneme's duration without complicating the latent space by taking additional text embeddings as input. The stacked prosody representations are fed into conformer blocks to extract variable-length prosody representations $\boldsymbol{h}_{\text{vl}}$. To compress these into a fixed-length vector, we use cross-attention Attn(k, q, v) with $\boldsymbol{h}_{\text{vl}}$ as the query and value, and learnable fixed-length positional embeddings $\boldsymbol{h}_{\text{pe}}$ as the key. This yields $\boldsymbol{h}_{\text{style}} = \text{Attn}(\boldsymbol{h}_{\text{pe}}, \boldsymbol{h}_{\text{vl}}, \boldsymbol{h}_{\text{vl}})$, named as *time-varying style* to differentiate it from previous works that use global style vectors to represent speech styles. The prosody decoder PD$(\cdot)$ (Figure 2b) then uses $\boldsymbol{h}_{\text{style}}$ to decode the duration $\hat{d}$, pitch $\hat{p}$, and energy $\hat{n}$ conditioned on the PL-BERT (Li et al., 2023b) phoneme embeddings from $\boldsymbol{t}$.

We noticed that this method achieves almost perfect prosody reconstruction with minimal perceptual difference from the input, even with a vector length of $K = 50$ for up to 30 seconds of speech without adversarial training. However, this leads to a latent space that is overly detailed, making diffusion modeling and one-step distillation difficult. To address this, we apply residual vector quantization (RVQ) (Zeghidour et al., 2021), simplifying the latent space by quantizing it. This reduces details in the latent space and simplifies the diffusion model's task at the cost of reconstruction accuracy of the autoencoder, which can be mitigated using adversarial training with multimodal discriminators. We use 9 codebooks with 1024 codes each and project the time-varying style with $d = 512$ into a lower space with $d = 8$ for efficient quantization following (Kumar et al., 2024), achieving a balance between diffusion model difficulty and prosody reconstruction fidelity (see Appendix A.2).

To further shift the burden of the diffusion model to the prosody decoder, we randomly mask and truncate the input pitch, energy, and duration fed to the prosody encoder. This technique encourages the prosody decoder to learn to reconstruct prosodic features from partial input, particularly beneficial for zero-shot TTS where the input prompt is short.

3.3 Distilled Time-Varying Style Diffusion

We deterministically sample the latent $\boldsymbol{h}_{\text{style}}$ (denote as $\boldsymbol{h}$) using DDIM (Song et al., 2020b):

$$d\boldsymbol{h} = \left[f(\boldsymbol{h}, \tau) - \frac{1}{2} g^2(\tau) \nabla_{\boldsymbol{h}} \log p_\tau(\boldsymbol{h}|\boldsymbol{t}, \boldsymbol{x}') \right] d\tau,$$
$$\boldsymbol{h}(1) \sim \mathcal{N}(0, \sigma_1^2 I), \tag{1}$$

where $f(\boldsymbol{h}, \tau) := \frac{d}{d\tau} \log \alpha_\tau$ is the drift coefficient, $g(\cdot, \tau) := \frac{d}{d\tau} \sigma_\tau^2 (1 - 2 \log (\alpha_\tau))$ is the diffusion coefficient, σ_τ is the noise level, α_τ is the schedule for σ_τ, and $\nabla_{\boldsymbol{h}} \log p_\tau(\boldsymbol{h}|\mathcal{C})$ is the score function of probability distribution of $\boldsymbol{h}$ at time τ conditioned on $\mathcal{C} = \{\boldsymbol{t}, \boldsymbol{x}'\}$, the PL-BERT embeddings $\boldsymbol{t}$ and speech prompt $\boldsymbol{x}'$, estimated using a denoiser $K(\cdot\,; \sigma_\tau, \mathcal{C})$ with architecture in Figure 2d:

$$\nabla_{\boldsymbol{h}} \log p_\tau(\boldsymbol{h}|\mathcal{C}) = \frac{\alpha_\tau K(\boldsymbol{h}; \sigma_\tau, \mathcal{C}) - \boldsymbol{h}(\tau)}{\sigma_\tau}. \tag{2}$$

We train the denoiser using the velocity formulation (Salimans and Ho, 2022):

$$\mathcal{L}_{\text{diff}} = \mathbb{E}_{\boldsymbol{x}, \boldsymbol{x}', \boldsymbol{t}, \tau \sim \mathcal{U}([0,1]), \boldsymbol{\xi} \sim \mathcal{N}(0, I)} [\| K(\alpha_\tau E(\boldsymbol{x}) +$$
$$\sigma_\tau \boldsymbol{\xi}; \sigma_\tau, \mathcal{C}) - \boldsymbol{v}(\sigma_\tau, E(\boldsymbol{x})) \|_1], \tag{3}$$

where $E(\cdot) : \mathcal{X} \to \mathcal{H}$ denotes combined pitch, energy and duration extractor and prosody encoder that maps speech $\boldsymbol{x} \in \mathcal{X}$ to latent $\boldsymbol{h} \in \mathcal{H}$. The velocity $\boldsymbol{v}$ is defined as $\boldsymbol{v}(\sigma_\tau, x) := \alpha_\tau \boldsymbol{\xi} - \sigma_\tau x$, with an angular scheduler $\alpha_\tau := \cos(\phi_\tau)$ and $\sigma_\tau := \sin(\phi_\tau)$ for $\phi_\tau = \frac{\pi}{2} \tau$.

We apply classifier-free guidance (CFG) (Ho and Salimans, 2022) using both speech prompt $\boldsymbol{x}'$ and text $\boldsymbol{t}$ as a condition. The modified denoiser with CFG is:

$$\tilde{K}(\cdot\,; \omega, \sigma_n, \mathcal{C}) := K(\cdot\,; \sigma_n, \emptyset) +$$
$$\omega \cdot (K(\cdot\,; \sigma_n, \mathcal{C}) - K(\cdot\,; \sigma_n, \emptyset)), \tag{4}$$

where ω is the guidance scale and $\emptyset$ indicates null condition embeddings. We randomly dropped out the condition $\boldsymbol{x}'$ with rate of 0.1 during training and fixed $\omega = 5$ during inference.

Equation 1 can be viewed as a neural ODE following a trajectory that maps a Gaussian noise

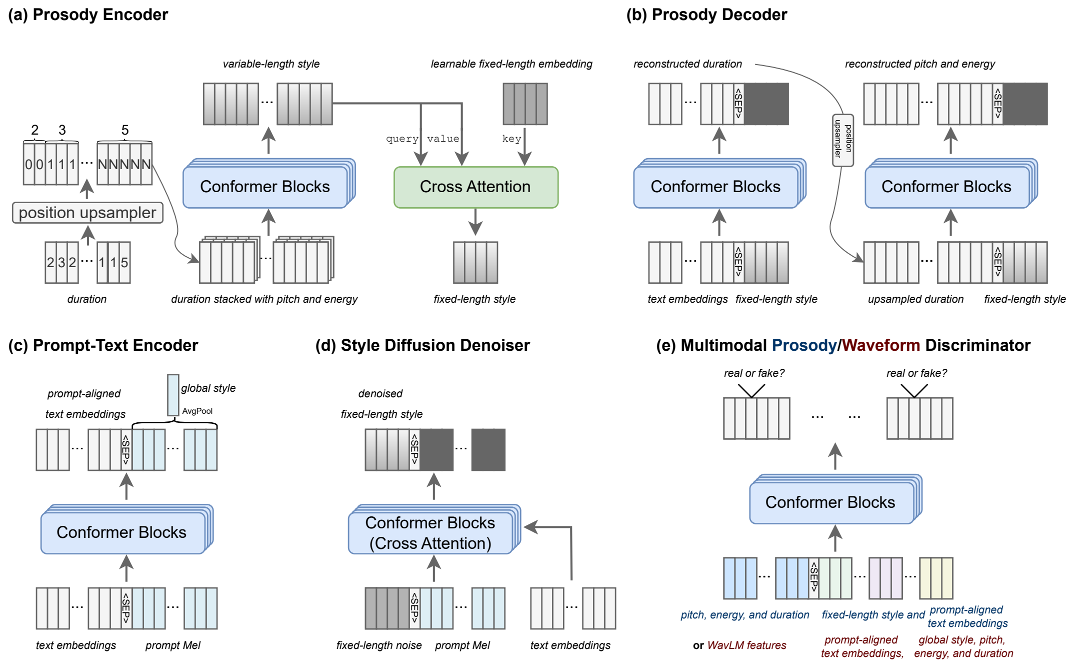


Figure 2: Architectures for newly proposed components in StyleTTS-ZS. For (b) and (d), the dark part of the output means this part is discarded for output, and only the grey part is used.

$\xi \in \mathcal{N}(0, I)$ to the time-varying style $\boldsymbol{h} \in \mathcal{H}$ conditioned on $\boldsymbol{x}'$ and $\boldsymbol{t}$. We denote this map as $h(\cdot\,; \omega, \mathcal{C}) : \mathcal{N} \to \mathcal{H}$. We solve this ODE using the deterministic solver DDIM (Song et al., 2020a) for later distillation that repeatedly applies the followings for $n \in \{L, L-1, \ldots, 0\}$:

$$\boldsymbol{h}_{\sigma_{n-1}} = \alpha_{n-1}\boldsymbol{v}_n + \sigma_{n-1}\tilde{\boldsymbol{v}}_n, \tag{5}$$

where $\boldsymbol{v}_n := \alpha_n \boldsymbol{v}_n - \sigma_n \tilde{K}(\boldsymbol{h}_{\sigma_n}; \omega, \sigma_n, \mathcal{C})$ and $\tilde{\boldsymbol{v}}_n = \sigma_n \boldsymbol{v}_n + \alpha_n \tilde{K}(\boldsymbol{h}_{\sigma_n}; \omega, \sigma_n, \mathcal{C})$ for $L = 100$.

We can train a student network $H(\cdot\,; \omega, \mathcal{C})$ to approximate h as it is a deterministic map. Since obtaining samples to train H through equation 5 can be expensive especially with large integration steps $L = 100$ due to the need to accurately reflect the effects of CFG, we initialize our student network using a pre-trained network H' that predicts prosody decoder output from the text $\boldsymbol{t}$ and a speech prompt $\boldsymbol{x}'$. This student initialization allows sample reduction to as small as 10k samples as shown in Appendix A.3, since the model has already learned a deterministic map from the speech prompt and text to the time-varying style as we match the initial student network and target distilled diffusion sampler in the output space. The student network can be pre-trained during the style diffusion training. We use perpetual loss for distillation (Liu et al., 2023), for which our perpetual metric is the prosody decoder's output. The distillation loss is defined as:

$$\begin{aligned}\mathcal{L}_{\text{dist}} = &\mathbb{E}_{\substack{\boldsymbol{x}',\boldsymbol{t},\boldsymbol{\xi}\sim\mathcal{N}(0,I),\\ \omega\in\mathcal{U}([1,15])}}[|\text{PD}\left(H(\boldsymbol{\xi}; \omega, \mathcal{C})\right) \\ &- \text{PD}\left(h(\boldsymbol{\xi}; \omega, \mathcal{C})\right)|_1].\end{aligned} \tag{6}$$

We show this simulation-based approach for distillation is superior to other simulation-free methods that use bootstrapping, such as consistency distillation (Song et al., 2023) and adversarial diffusion distillation (Sauer et al., 2023) in Appendix A.3. This is because our latent space is a 50×512 vector and can be sampled fairly fast and distilled with 10k samples, which took a few hours to obtain on 8 NVIDIA RTX 3090 GPUs.

3.4 Multimodal Discriminators

One observation we made is the trade-off between latent space complexity, reconstruction error, and the difficulty of diffusion model training and distillation (see Appendix A.2). A simpler latent space makes the diffusion model easier to train and distill but increases the reconstruction error for the autoencoder. To achieve efficient latent diffusion and high-fidelity distillation, we opted to simplify the latent space for the time-varying style and optimize the autoencoder for high-quality reconstruction. We introduce multimodal discriminators that evaluate not only the decoder output to determine

Table 1: Comparative mean opinion scores of naturalness (CMOS-N) and similarity (COMS-S) for StyleTTS-ZS (LL) relative to other models (negative scores indicate StyleTTS-SZ (LL) is better; one asterisk * indicates $p < 0.05$ and two asterisks ** indicate $p < 0.01$), predicted MOS (UT-MOS), speaker embedding similarity (SIM), word error rate (WER), coefficient of variation for pitch and energy (CV_{p+n}) and real-time factor (RTF)[3] in comparison to other recent large-scale models and StyleTTS-ZS (LT).

Model	Training Set	CMOS-N	CMOS-S	UT-MOS ↑	WER ↓	SIM ↑	CV_{p+n} ↑	RTF ↓
Ground Truth	—	0.44^{*}	-0.77^{**}	4.17	0.34	0.67	1.49	—
Vall-E	LibriLight	-1.07^{**}	-0.65^{**}	3.31	4.97	0.47	0.94	0.62 †
NaturalSpeech 2	MLS	-0.57^{**}	-0.19^{*}	3.78	1.25	0.55	1.07	0.37 ‡
VoiceCraft	GigaSpeech	-0.84^{**}	-0.11^{*}	3.58	3.73	0.54	0.79	1.24 ‡
FlashSpeech	MLS	-0.42^{*}	-0.52^{**}	3.98	1.47	0.50	0.50	**0.02** ‡
NaturalSpeech 3	LibriLight	-0.28^{*}	0.01	4.09	1.81	**0.66**	1.23	0.30 ‡
StyleTTS-ZS (LT)	LibriTTS	-0.21^{*}	-0.31^{*}	**4.24**	0.90	0.47	1.06	0.03 ‡
StyleTTS-ZS (LL)	LibriLight	**0.00**	**0.00**	4.16	**0.79**	0.56	**1.67**	0.03 ‡

if the sample is real or fake but also consider the decoder input as an additional modality, which has been shown to improve speech quality for zero-shot speech synthesis (Janiczek et al., 2024).

Specifically, we use two multimodal discriminators (Figure 2e): one for the waveform decoder and one for the prosody decoder. The waveform discriminator takes the WavLM (Chen et al., 2022) features of the decoder's output following the idea of SLM discriminator in (Li et al., 2024a) that has been subsequently demonstrated effectively (Li et al., 2023c; Ye et al., 2024) and conditions on all the inputs to the decoder (prompt-aligned text embeddings, global style, pitch, energy, and duration). The prosody discriminator takes the prosody decoder's output and conditions on all the inputs to the prosody decoder, including the time-varying style and PL-BERT text embeddings. This approach significantly enhances the naturalness and similarity of the reconstructed speech, as demonstrated in Section 4.3. In addition to the multimodal discriminator, we also have the multi-period discriminator (MPD) (Kong et al., 2020) and multi-resolution STFT discriminator (MRD) (Kumar et al., 2024) for the waveform decoder.

4 Experiments

4.1 Model Training

We trained our model on the LibriTTS and LibriLight datasets. First, we used the LibriTTS dataset (Zen et al., 2019), with 585 hours of speech from 1,185 speakers, excluding utterances shorter than 1 second or longer than 30 seconds. The model was trained on the *train-clean-100*, *train-clean-360*, and *train-other-500* subsets. We then trained on the LibriLight dataset (Kahn et al., 2020), which consists of 57,706 hours of audio from 7,439 speakers, and used the methods from (Kang et al., 2024) to transcribe the text. All datasets were resampled to 24 kHz, and texts were converted into phonemes using Phonemizer (Bernard and Titeux, 2021). We truncated input speech randomly to the smallest batch length and used it as prompts. Training was done for 30 epochs on LibriTTS and 1 million steps on LibriLight. The style diffusion denoiser was distilled using 10k samples, and the student model was trained for 10 epochs. We trained our model using the AdamW optimizer (Loshchilov and Hutter, 2018) with $\beta_1 = 0$, $\beta_2 = 0.99$, weight decay $\lambda = 10^{-4}$, learning rate $\gamma = 10^{-4}$, and a batch size of 32 samples on four NVIDIA L40 GPUs.

4.2 Evaluations

We employed two metrics in our experiments: Mean Opinion Score of Naturalness (MOS-N) for human-likeness, and Mean Opinion Score of Similarity (MOS-S) for similarity to the prompt speaker. These evaluations were conducted by native English speakers from the U.S. on Amazon Mechanical Turk. All evaluators reported normal hearing and provided informed consent. We conducted two experiments with different groups of baseline models: one for small-scale models and another for large-scale models. For small-scale models, we compared our model to three high-performing public models: XTTS-v2 (Casanova et al., 2022), StyleTTS 2 (Li et al., 2024a), and HierSpeech++ (Lee et al., 2023) on the LibriTTS dataset. Each synthesized speech set was rated by 10 evaluators on a 1-5 scale, with 0.5 increments. We randomized the model order and kept their labels hidden, similar to the MUSHRA approach (Li et al., 2021,

Table 2: Comparison of MOS with 95% confidence intervals (CI), word error rate (WER) and real-time factor (RTF) for public models trained on LibriTTS. StyleTTS-ZS uses LT model, and StyleTTS 2 uses 5 steps for style diffusion.

Model	MOS-N (CI) ↑	MOS-S (CI) ↑	WER ↓	RTF ↓
Ground Truth	4.67 ($\pm$ 0.11)	4.32 ($\pm$ 0.10)	0.34	—
StyleTTS-ZS	**4.54** (**$\pm$ 0.11**)	**4.33** (**$\pm$ 0.11**)	**0.90**	**0.0320**
StyleTTS 2	4.23 ($\pm$ 0.11)	3.42 ($\pm$ 0.09)	1.61	0.0671
HierSpeech++	3.54 ($\pm$ 0.12)	4.27 ($\pm$ 0.12)	7.82	0.1969
XTTSv2	3.68 ($\pm$ 0.09)	3.74 ($\pm$ 0.10)	6.17	0.3861

2022). We tested 40 samples from the LibriSpeech (Panayotov et al., 2015) *test-clean* subset with 3-second refernece speech, following (Wang et al., 2023a). Official checkpoints trained on LibriTTS were used for all baseline models (see Appendix B.1 for more information). For large-scale experiments, since most state-of-the-art large-scale models are not publicly available, we compared our model to audio samples obtained from the authors or official demo pages using comparative MOS (CMOS) tests, as raters can ignore subtle differences in MOS experiments, making it difficult to estimate accurate performance from limited samples. Raters compared pairs of samples and rated whether the second was better or worse (or more or less similar to the prompt speaker) than the first on a -6 to 6 scale, with 1-point increments. We included five recent models: Vall-E, NaturalSpeech 2, NaturalSpeech 3, FlashSpeech, and VoiceCraft. For Vall-E, NaturalSpeech 2/3 and FlashSpeech, we obtained 40 samples from the authors with 3-second of prompt speech of unseen speakers in LibriSpeech *test-clean* subset and used these samples for evaluations. Since the model of VoiceCraft is publicly available, we synthesized the samples using the same 40 prompts and texts.

In addition to subjective evaluations, we followed (Shen et al., 2024; Ju et al., 2024; Ye et al., 2024) for objective evaluations of sound quality using predicted MOS (UT-MOS) (Saeki et al., 2022), robustness using word error rate (WER) from a pre-trained ASR model [1] and similarity to the reference speaker (SIM) by cosine similarity from a pre-trained speaker verification model [2]. Additionally, we measured prosody similarity by computing the Pearson correlation coefficients of acoustic features associated with emotions and speech duration between the prompt and the synthesized speech, following (Li et al., 2022).

As shown in Table 1, our model trained on the large dataset has outperformed previous state-of-the-art (SOTA) large-scale TTS models in multiple metrics: human rated naturalness (CMOS-N), predicted sound quality (UT-MOS), similarity (CMOS-S), expressiveness (CV), inference time (RTF), and robustness (as indicated by WER). We note that we achieve competitive performance in SIM with most models except NaturalSpeech 3 despite having a statistically insignificant CMOS-S compared to it. This may be due to our model's adversarial training with multimodal discriminators, which enhances speaker likeness from a human perception perspective, whereas other models use a pre-trained neural codec that aligns more with neural network perceptions but not necessarily human perceptions (Ju et al., 2024). Although the current SOTA NaturalSpeech 3 has achieved ground-truth level performance in terms of similarity measured by speaker verification models, it still falls short of robustness and naturalness where our model excels. Notably, our model has demonstrated similar performance in terms of human-perceived similarity as NaturalSpeech 3 and has achieved similarly superior perceived similarity than ground truth. In addition, our model is 10× faster than NaturalSpeech 3.

Since our model does not use iterative refinement methods, it is among the fastest large-scale TTS models, second only to FlashSpeech in efficiency, while significantly surpassing it in both naturalness and similarity. Additionally, our model exhibits greater robustness than all other models, as indicated by the WER scores. The expressiveness of our model, shown by the pitch and energy standard deviation, indicates that it closely matches the ground truth in terms of speech variation and expressiveness. We also include an breakdown of time taken for each module in Appendix 4.4.

Our model also outperforms other public models on small-scale data with only 585 hours of audio, as shown in Table 2. Moreover, when comparing our model trained on larger data (StyleTTS-ZS LL), both CMOS-N and CMOS-S scores are significantly higher than the model trained on smaller data (StyleTTS-ZS LT), despite using the same amount of parameters. This demonstrates our model's scalability and capability to handle larger datasets ef-

[1] https://huggingface.co/facebook/hubert-large-ls960-ft

[2] https://github.com/microsoft/UniSpeech/tree/main/downstreams/speaker_verification

[3] †: device unknown and results are taken from the original paper. ‡: RTF was computed on an NVIDIA V100 GPU.

Table 3: Ablation study on LibriTTS for verifying the effectiveness of each proposed component. Significant results ($p < 0.05$) are marked by an asterisk (*). For w/o distillation, the RTF is 0.28.

Model	CMOS-N	CMOS-S	UT-MOS	SIM	WER
StyleTTS-ZS	0	0	4.24	0.47	0.90
w/o PATE	-0.24^*	-0.18^*	3.98	0.38	1.14
w/o global style	-0.31^*	-0.47^*	3.54	0.34	1.45
w/o SEFM Loss	-0.02	-0.23^*	4.31	0.40	1.01

Model	CMOS-N	CMOS-S	UT-MOS	SIM	WER
StyleTTS-ZS	0	0	4.24	0.47	0.90
w/o distillation	-0.02	0.06	4.12	0.46	0.90
w/o MMWD	-0.24^*	-0.29^*	3.97	0.42	1.22
w/o MMPD	-0.58^*	-0.32^*	4.19	0.45	0.96

fectively. In Table 5, we see that our model has outperformed other zero-shot TTS models in most of the acoustic characteristics associated with emotions, demonstrating its ability to reproduce the speech style of prompt speech.

4.3 Ablation Study

To verify the effectiveness of each proposed component, we conducted ablation studies on LibriTTS using two subjective metrics, CMOS-N for naturalness and CMOS-S for similarity, and evaluated UT-MOS, speaker embedding similarity (SIM) and word error rate (WER) for robustness. We used the same texts and prompt speech as in the other MOS experiments and tested the following variations:

- *w/o PATE*: Using only global styles in acoustic synthesizer without prompt-aligned text embeddings (PATE). Global styles are computed along with the prompt, but the text embeddings are computed with a prompt of value all 0.
- *w/o global style*: Using only prompt-aligned embeddings in the acoustic synthesizer without global styles. All AdaIN layers in the decoder were replaced with instance normalization.
- *w/o SEFM Loss*: No speaker embedding feature matching (SEFM) loss as in eq. 14.
- *w/o distillation*: Using the original diffusion model instead of distilled one for inference.
- *w/o MMWD*: No multimodal waveform discriminator (MMWD) for acoustic synthesizer.
- *w/o MMPD*: No multimodal prosody discriminator (MMPD) for the prosody autoencoder.

As shown in Table 3, both prompt-aligned text embeddings and global style vectors are crucial for high-quality speech synthesis with high fidelity to the prompt speaker, with global style being more important likely because the AdaIN-based decoder benefits significantly from the global style, as demonstrated in (Li et al., 2022). Without the speaker embedding feature matching loss, there is a significant decrease in speaker similarity, though UT-MOS is slightly higher because the model does not have to follow the sound quality of the prompt strictly. Using the distilled instead of the original diffusion model has minimal impact on perceived naturalness and similarity with even a boost of predicted MOS due to mode shrinkage during distillation as the model learns sample the mode of the distributions. However, it significantly reduces inference speed by nearly 90%, proving the effectiveness and efficiency of our distillation design. Lastly, removing either the multimodal waveform discriminator or multimodal prosody discriminator significantly decreases perceived naturalness and similarity, with the multimodal prosody discriminator having a more substantial impact. This is because the acoustic synthesizer still benefits from MPD and MRD during training, while the prosody autoencoder only relies on ℓ_1 loss without adversarial training. However, since UT-MOS primarily focuses on the acoustic aspects of speech for naturalness while largely ignores the prosody naturalness, the predicted MOS is unaffected. A similar effect is observed with SIM, which largely focuses on global characteristics such as timbre, and hence prosody has little effect on this metric.

4.4 Processing Time Analysis

We analyzed the processing time of each module using the same 40 samples as those used for computing the real-time factor (RTF). The percentage of time taken by each module is shown in Table 4. The results indicate that the acoustic decoder is the most time-consuming component, suggesting a potential area for improvement. Future work could focus on reducing the acoustic synthesis time by employing ultra-fast acoustic decoders, such as those utilizing inverse short-time Fourier transform (iSTFT), as demonstrated in Voco (Siuzdak, 2023) and HiFT-NET (Li et al., 2023a).

Table 4: The average percentage of processing time for each module for synthesizing varying lengths of speech.

Module	% Time
Prompt-Text Encoder	6.1%
Distilled Style Diffusion Model	12.3 %
Prosody Decoder	9.9 %
Acoustic Decoder	71.7 %

5 Conclusions

We introduced StyleTTS-ZS, a highly efficient zero-shot TTS system that matches previous state-of-the-art models while being 10-20 times faster. With distilled time-varying style diffusion, it captures diverse speaker identities and prosodies, showing strong scalability on large datasets like LibriLight. The model's speed make it ideal for real-time applications, such as virtual assistants and customized dialog generation, especially when integrated with large language models as demonstrated by StyleTalker (Li et al., 2024b) where style-based TTS is integrated into spoken dialog systems. This advancement also benefits audiobook narration, accessibility tools, and media content creation.

6 Limitations

Although our model outperforms previous state-of-the-art models, it still has not achieved human-level performance for zero-shot TTS with unseen speakers, as models trained on small-scale datasets for seen speakers have achieved (Tan et al., 2024; Li et al., 2024a). Moreover, our model prioritizes speed over quality, meaning the acoustic synthesizer does not benefit from the latest generative modeling developments with iterative refinements that could further enhance speaker fidelity to the prompt speaker. Since recent works have demonstrated that diffusion-based models with iterative sampling can achieve close-to-human similarity to the prompt speaker (Eskimez et al., 2024; Chen et al., 2024b; Wang et al., 2024), it is worth investigating the potential of replacing our GAN-based acoustic synthesizer with diffusion-based one while preserving the superior inference speed through further distillation. Additionally, the models were trained on English audiobook reading datasets rather than audio from more diverse, real-world environments and other languages. Future research should explore higher-quality generation methods to improve quality without compromising speed, expand training to include more diverse datasets and languages to enhance generalizability, and further refine prosody modeling to achieve even more natural and expressive speech synthesis. Still, our findings indicate that StyleTTS-ZS is a robust and efficient solution for zero-shot TTS synthesis, with promising potential for applications on large-scale real-world data and future research directions.

7 Ethical Concerns

The capabilities of our model can bring potential negative impacts. The ability to generate high-quality, recognizable speech could be misused for voice spoofing, posing risks to personal and financial security. There is also the potential for creating convincing deepfake audio, which can be used maliciously in misinformation campaigns, fraud, or defamation. To mitigate these risks, we recommend controlled access to the model, with strict licensing agreements that require users to obtain consent from individuals whose voices are being cloned. Establishing ethical guidelines and usage policies is essential to prevent misuse. Additionally, encouraging collaboration in the research community to develop and improve deepfake detection technologies and incorporating mechanisms to watermark or trace synthetic audio to ensure accountability and traceability are necessary steps.

8 Acknowledgments

We thank Gavin Mischler and Ashley Thomas for assessing the quality of synthesized samples and providing feedback on the quality of models during the development stage of this work. We also thank Kundan Kumar for his feedback on the manuscript. This work was done in part during an internship at Descript Inc. (Y.A. Li) and was funded by the National Institute of Health (NIHNIDCD) and a grant from Marie-Josee and Henry R. Kravis.

References

Mathieu Bernard and Hadrien Titeux. 2021. Phonemizer: Text to Phones Transcription for Multiple Languages in Python. *Journal of Open Source Software*, 6(68):3958.

Edresson Casanova, Christopher Shulby, Eren Gölge, Nicolas Michael Müller, Frederico Santos De Oliveira, Arnaldo Candido Junior, Anderson da Silva Soares, Sandra Maria Aluisio, and Moacir Antonelli Ponti. 2021. Sc-glowtts: An efficient zero-shot multi-speaker text-to-speech model. *arXiv preprint arXiv:2104.05557*.

Edresson Casanova, Julian Weber, Christopher D Shulby, Arnaldo Candido Junior, Eren Gölge, and Moacir A Ponti. 2022. Yourtts: Towards zero-shot multi-speaker tts and zero-shot voice conversion for everyone. In *International Conference on Machine Learning*, pages 2709–2720. PMLR.

Sanyuan Chen, Shujie Liu, Long Zhou, Yanqing Liu, Xu Tan, Jinyu Li, Sheng Zhao, Yao Qian, and Furu Wei. 2024a. Vall-e 2: Neural codec language models are human parity zero-shot text to speech synthesizers. *arXiv preprint arXiv:2406.05370*.

Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, et al. 2022. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1505–1518.

Yushen Chen, Zhikang Niu, Ziyang Ma, Keqi Deng, Chunhui Wang, Jian Zhao, Kai Yu, and Xie Chen. 2024b. F5-tts: A fairytaler that fakes fluent and faithful speech with flow matching. *arXiv preprint arXiv:2410.06885*.

Hyeong-Seok Choi, Jinhyeok Yang, Juheon Lee, and Hyeongju Kim. 2022. Nansy++: Unified voice synthesis with neural analysis and synthesis. *arXiv preprint arXiv:2211.09407*.

Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. 2022. High fidelity neural audio compression. *arXiv preprint arXiv:2210.13438*.

Sefik Emre Eskimez, Xiaofei Wang, Manthan Thakker, Canrun Li, Chung-Hsien Tsai, Zhen Xiao, Hemin Yang, Zirun Zhu, Min Tang, Xu Tan, et al. 2024. E2 tts: Embarrassingly easy fully non-autoregressive zero-shot tts. *arXiv preprint arXiv:2406.18009*.

Wenhao Guan, Qi Su, Haodong Zhou, Shiyu Miao, Xingjia Xie, Lin Li, and Qingyang Hong. 2024. Reflow-tts: A rectified flow model for high-fidelity text-to-speech. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 10501–10505. IEEE.

Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, et al. 2020. Conformer: Convolution-augmented transformer for speech recognition. *arXiv preprint arXiv:2005.08100*.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851.

Jonathan Ho and Tim Salimans. 2022. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*.

Rongjie Huang, Yi Ren, Jinglin Liu, Chenye Cui, and Zhou Zhao. 2022a. Generspeech: Towards style transfer for generalizable out-of-domain text-to-speech. *Advances in Neural Information Processing Systems*, 35:10970–10983.

Rongjie Huang, Zhou Zhao, Huadai Liu, Jinglin Liu, Chenye Cui, and Yi Ren. 2022b. Prodiff: Progressive fast diffusion model for high-quality text-to-speech. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 2595–2605.

John Janiczek, Dading Chong, Dongyang Dai, Arlo Faria, Chao Wang, Tao Wang, and Yuzong Liu. 2024. Multi-modal adversarial training for zero-shot voice cloning. *arXiv preprint arXiv:2408.15916*.

Ziyue Jiang, Jinglin Liu, Yi Ren, Jinzheng He, Chen Zhang, Zhenhui Ye, Pengfei Wei, Chunfeng Wang, Xiang Yin, Zejun Ma, et al. 2023a. Mega-tts 2: Zero-shot text-to-speech with arbitrary length speech prompts. *arXiv preprint arXiv:2307.07218*.

Ziyue Jiang, Yi Ren, Zhenhui Ye, Jinglin Liu, Chen Zhang, Qian Yang, Shengpeng Ji, Rongjie Huang, Chunfeng Wang, Xiang Yin, et al. 2023b. Mega-tts: Zero-shot text-to-speech at scale with intrinsic inductive bias. *arXiv preprint arXiv:2306.03509*.

Zeqian Ju, Yuancheng Wang, Kai Shen, Xu Tan, Detai Xin, Dongchao Yang, Yanqing Liu, Yichong Leng, Kaitao Song, Siliang Tang, et al. 2024. Natural-speech 3: Zero-shot speech synthesis with factorized codec and diffusion models. *arXiv preprint arXiv:2403.03100*.

Jacob Kahn, Morgane Riviere, Weiyi Zheng, Evgeny Kharitonov, Qiantong Xu, Pierre-Emmanuel Mazaré, Julien Karadayi, Vitaliy Liptchinsky, Ronan Collobert, Christian Fuegen, et al. 2020. Libri-light: A benchmark for asr with limited or no supervision. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7669–7673. IEEE.

Wei Kang, Xiaoyu Yang, Zengwei Yao, Fangjun Kuang, Yifan Yang, Liyong Guo, Long Lin, and Daniel Povey. 2024. Libriheavy: a 50,000 hours asr corpus with punctuation casing and context. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 10991–10995. IEEE.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. 2022. Elucidating the design space of diffusion-based generative models. *Advances in Neural Information Processing Systems*, 35:26565–26577.

Jaehyeon Kim, Jungil Kong, and Juhee Son. 2021. Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech. In *International Conference on Machine Learning*, pages 5530–5540. PMLR.

Jaehyeon Kim, Keon Lee, Seungjun Chung, and Jaewoong Cho. 2024a. Clam-tts: Improving neural codec language model for zero-shot text-to-speech. *arXiv preprint arXiv:2404.02781*.

Sungwon Kim, Kevin Shih, Joao Felipe Santos, Evelina Bakhturina, Mikyas Desta, Rafael Valle, Sungroh Yoon, Bryan Catanzaro, et al. 2024b. P-flow: A fast and data-efficient zero-shot tts through speech prompting. *Advances in Neural Information Processing Systems*, 36.

Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. 2020. Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis. *Advances in neural information processing systems*, 33:17022–17033.

Rithesh Kumar, Prem Seetharaman, Alejandro Luebs, Ishaan Kumar, and Kundan Kumar. 2024. High-fidelity audio compression with improved rvqgan. *Advances in Neural Information Processing Systems*, 36.

Matthew Le, Apoorv Vyas, Bowen Shi, Brian Karrer, Leda Sari, Rashel Moritz, Mary Williamson, Vimal Manohar, Yossi Adi, Jay Mahadeokar, et al. 2024. Voicebox: Text-guided multilingual universal speech generation at scale. *Advances in neural information processing systems*, 36.

Keon Lee, Dong Won Kim, Jaehyeon Kim, and Jaewoong Cho. 2024. Ditto-tts: Efficient and scalable zero-shot text-to-speech with diffusion transformer. *arXiv preprint arXiv:2406.11427*.

Sang-Hoon Lee, Ha-Yeong Choi, Seung-Bin Kim, and Seong-Whan Lee. 2023. Hierspeech++: Bridging the gap between semantic and acoustic representation of speech by hierarchical variational inference for zero-shot speech synthesis. *arXiv preprint arXiv:2311.12454*.

Sang-Hoon Lee, Seung-Bin Kim, Ji-Hyun Lee, Eunwoo Song, Min-Jae Hwang, and Seong-Whan Lee. 2022. Hierspeech: Bridging the gap between text and speech by hierarchical variational inference using self-supervised representations for speech synthesis. *Advances in Neural Information Processing Systems*, 35:16624–16636.

Yinghao Aaron Li, Cong Han, Xilin Jiang, and Nima Mesgarani. 2023a. Hiftnet: A fast high-quality neural vocoder with harmonic-plus-noise filter and inverse short time fourier transform. *arXiv preprint arXiv:2309.09493*.

Yinghao Aaron Li, Cong Han, Xilin Jiang, and Nima Mesgarani. 2023b. Phoneme-level bert for enhanced prosody of text-to-speech with grapheme predictions. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Yinghao Aaron Li, Cong Han, and Nima Mesgarani. 2022. Styletts: A style-based generative model for natural and diverse text-to-speech synthesis. *arXiv preprint arXiv:2205.15439*.

Yinghao Aaron Li, Cong Han, and Nima Mesgarani. 2023c. Slmgan: Exploiting speech language model representations for unsupervised zero-shot voice conversion in gans. In *2023 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 1–5. IEEE.

Yinghao Aaron Li, Cong Han, Vinay Raghavan, Gavin Mischler, and Nima Mesgarani. 2024a. Styletts 2: Towards human-level text-to-speech through style diffusion and adversarial training with large speech language models. *Advances in Neural Information Processing Systems*, 36.

Yinghao Aaron Li, Jiang Xilin, Jordan Darefsky, Ge Zhu, and Nima Mesgarani. 2024b. Style-talker: Finetuning audio language model and style-based text-to-speech model for fast spoken dialogue generation. *First Conference on Language Modeling*.

Yinghao Aaron Li, Ali Zare, and Nima Mesgarani. 2021. StarGANv2-VC: A Diverse, Unsupervised, Non-parallel Framework for Natural-Sounding Voice Conversion. *arXiv preprint arXiv:2107.10394*.

Xingchao Liu, Xiwen Zhang, Jianzhu Ma, Jian Peng, et al. 2023. Instaflow: One step is enough for high-quality diffusion-based text-to-image generation. In *The Twelfth International Conference on Learning Representations*.

Ilya Loshchilov and Frank Hutter. 2018. Fixing Weight Decay Regularization in Adam.

Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. 2023. Latent consistency models: Synthesizing high-resolution images with few-step inference. *arXiv preprint arXiv:2310.04378*.

Lingwei Meng, Long Zhou, Shujie Liu, Sanyuan Chen, Bing Han, Shujie Hu, Yanqing Liu, Jinyu Li, Sheng Zhao, Xixin Wu, et al. 2024. Autoregressive speech synthesis without vector quantization. *arXiv preprint arXiv:2407.08551*.

Dongchan Min, Dong Bok Lee, Eunho Yang, and Sung Ju Hwang. 2021. Meta-stylespeech: Multi-speaker adaptive text-to-speech generation. In *International Conference on Machine Learning*, pages 7748–7759. PMLR.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5206–5210. IEEE.

Puyuan Peng, Po-Yao Huang, Daniel Li, Abdelrahman Mohamed, and David Harwath. 2024. Voicecraft: Zero-shot speech editing and text-to-speech in the wild. *arXiv preprint arXiv:2403.16973*.

Vadim Popov, Ivan Vovk, Vladimir Gogoryan, Tasnima Sadekova, and Mikhail Kudinov. 2021. Grad-tts: A diffusion probabilistic model for text-to-speech. In *International Conference on Machine Learning*, pages 8599–8608. PMLR.

Yi Ren, Chenxu Hu, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu. 2020. Fastspeech 2: Fast and high-quality end-to-end text to speech. *arXiv preprint arXiv:2006.04558*.

Takaaki Saeki, Detai Xin, Wataru Nakata, Tomoki Koriyama, Shinnosuke Takamichi, and Hiroshi Saruwatari. 2022. Utmos: Utokyo-sarulab system for voicemos challenge 2022. *arXiv preprint arXiv:2204.02152*.

Tim Salimans and Jonathan Ho. 2022. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*.

Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. 2023. Adversarial diffusion distillation. *arXiv preprint arXiv:2311.17042*.

Kai Shen, Zeqian Ju, Xu Tan, Eric Liu, Yichong Leng, Lei He, Tao Qin, Jiang Bian, et al. 2024. Naturalspeech 2: Latent diffusion models are natural and zero-shot speech and singing synthesizers. In *The Twelfth International Conference on Learning Representations*.

Hubert Siuzdak. 2023. Vocos: Closing the gap between time-domain and fourier-based neural vocoders for high-quality audio synthesis. *arXiv preprint arXiv:2306.00814*.

Jiaming Song, Chenlin Meng, and Stefano Ermon. 2020a. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*.

Yakun Song, Zhuo Chen, Xiaofei Wang, Ziyang Ma, and Xie Chen. 2024. Ella-v: Stable neural codec language modeling with alignment-guided sequence reordering. *arXiv preprint arXiv:2401.07333*.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. 2023. Consistency models. *arXiv preprint arXiv:2303.01469*.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. 2020b. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*.

Xu Tan, Jiawei Chen, Haohe Liu, Jian Cong, Chen Zhang, Yanqing Liu, Xi Wang, Yichong Leng, Yuanhao Yi, Lei He, et al. 2024. Naturalspeech: End-to-end text-to-speech synthesis with human-level quality. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Aaron Van Den Oord, Oriol Vinyals, et al. 2017. Neural discrete representation learning. *Advances in neural information processing systems*, 30.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Chengyi Wang, Sanyuan Chen, Yu Wu, Ziqiang Zhang, Long Zhou, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, et al. 2023a. Neural codec language models are zero-shot text to speech synthesizers. *arXiv preprint arXiv:2301.02111*.

Hongji Wang, Chengdong Liang, Shuai Wang, Zhengyang Chen, Binbin Zhang, Xu Xiang, Yanlei Deng, and Yanmin Qian. 2023b. Wespeaker: A research and production oriented speaker embedding learning toolkit. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Xiaofei Wang, Manthan Thakker, Zhuo Chen, Naoyuki Kanda, Sefik Emre Eskimez, Sanyuan Chen, Min Tang, Shujie Liu, Jinyu Li, and Takuya Yoshioka. 2023c. Speechx: Neural codec language model as a versatile speech transformer. *arXiv preprint arXiv:2308.06873*.

Yuancheng Wang, Haoyue Zhan, Liwei Liu, Ruihong Zeng, Haotian Guo, Jiachen Zheng, Qiang Zhang, Shunsi Zhang, and Zhizheng Wu. 2024. Maskgct: Zero-shot text-to-speech with masked generative codec transformer. *arXiv preprint arXiv:2409.00750*.

Yihan Wu, Xu Tan, Bohan Li, Lei He, Sheng Zhao, Ruihua Song, Tao Qin, and Tie-Yan Liu. 2022. Adaspeech 4: Adaptive text to speech in zero-shot scenarios. *arXiv preprint arXiv:2204.00436*.

Dongchao Yang, Rongjie Huang, Yuanyuan Wang, Haohan Guo, Dading Chong, Songxiang Liu, Xixin Wu, and Helen Meng. 2024. Simplespeech 2: Towards simple and efficient text-to-speech with flow-based scalar latent transformer diffusion models. *arXiv preprint arXiv:2408.13893*.

Zhen Ye, Zeqian Ju, Haohe Liu, Xu Tan, Jianyi Chen, Yiwen Lu, Peiwen Sun, Jiahao Pan, Weizhen Bian, Shulin He, et al. 2024. Flashspeech: Efficient zero-shot speech synthesis. *arXiv preprint arXiv:2404.14700*.

Zhen Ye, Wei Xue, Xu Tan, Jie Chen, Qifeng Liu, and Yike Guo. 2023. Comospeech: One-step speech and singing voice synthesis via consistency model. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 1831–1839.

Neil Zeghidour, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi. 2021. Soundstream: An end-to-end neural audio codec. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:495–507.

Heiga Zen, Viet Dang, Rob Clark, Yu Zhang, Ron J Weiss, Ye Jia, Zhifeng Chen, and Yonghui Wu. 2019. Libritts: A corpus derived from librispeech for text-to-speech. *arXiv preprint arXiv:1904.02882*.

A Additional Results

A.1 Other Applications

Speech Editing. By training our prosody decoder with masked prosody, we enable speech editing. To edit a specific part of the speech, we mask the prosody of that segment and decode the prosody conditioned on the new text and input speech as a prompt. This approach retains the original prosody as much as possible while generating new prosody for the edited segment. Re-synthesizing the speech using the edited prosody and the input speech prompt allows for seamless speech editing.

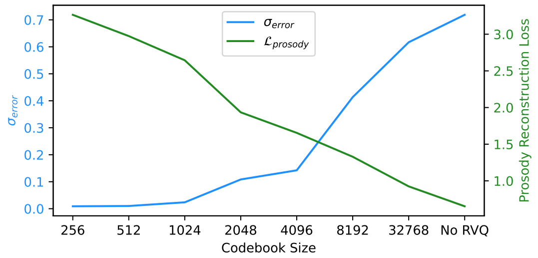
Zero-Shot Voice Conversion. Our model also supports zero-shot voice conversion conditioned on the text of the input speech, which can be accurately obtained through modern ASR models. We extract the duration, energy, and pitch from the input speech, and the energy and pitch from the prompt. By computing the median of both input and prompt pitch and energy and re-normalizing the input pitch and energy to match the prompt's median values, we ensure consistency. Using the ground truth duration, rescaled pitch, and energy, we can re-synthesize the speech with the prompt speech, achieving effective zero-shot voice conversion. We provide samples on our audio demo page.

A.2 Prosody Autoencoder Bottleneck

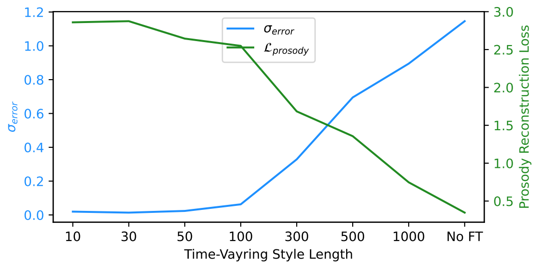
We examine the trade-off between the reconstruction error of the prosody autoencoder and the diffusion modeling difficulty by varying the codebook size with the length $K = 50$ of time-varying style and varying length with a fixed codebook size of 1024. We measure the reconstruction error by the aggregated prosody loss $\mathcal{L}_{prosody} = \mathcal{L}_{dur} + \mathcal{L}_{f0} + \mathcal{L}_{n}$ as defined in C.2. We use the normalized standard deviation error as a measure for diffusion modeling complexity:

$$\sigma_{\text{error}} = \frac{\|\sigma_{\text{data}} - \sigma_{\text{diff}}\|_1}{\sigma_{\text{data}}}, \tag{7}$$

where σ_{data} is the standard deviation of the data distribution $\mathcal{D}$ and σ_{diff} is that of diffusion samples. When the diffusion model converges but the error in standard deviation is large, it indicates that the diffusion model is not powerful enough to sample from $\mathcal{D}$ starting from a unit standard deviation distribution $\mathcal{N}(0, 1)$, an observation made in (Karras et al., 2022).



(a) Effects of codebook size with $K = 50$.



(b) Effects of K with codebooks size of 1024.

Figure 3: Effects of bottleneck complexity of prosody autoencoder and diffusion denoiser's performance. (a) Fixed time-varying style length $K = 50$ and varying codebook size. (b) Fixed codebook size of 1024 and varying style length K.

As shown in Figure 3 (a), higher codebook size results in lower reconstruction error but higher σ_{error}, with reconstruction error being the lowest while σ_{erorr} being the highest when no RVQ is used. This finding justifies the importance of using RVQ in our prosody autoencoder design. Similarly, with a fixed codebook size, the higher the length of the time-varying style, the lower the reconstruction error but the higher the σ_{error}. When varying length style vector is used as in the "No FT" case, the prosody encoder is the same as just applying RVQ to the input prosody, and thus, the diffusion complexity is proportional to the input size. This finding motivates us to apply adversarial training with multimodal discriminators for our prosody autoencoder training in order to shift the burden from the diffusion model to the prosody decoder for efficient sample generation.

A.3 Diffusion Distillation

We compare our simulation-based distillation with perceptual loss to simulation-free bootstrapping methods such as consistency distillation (Song et al., 2023) and adversarial diffusion distillation (Sauer et al., 2023) with varying sample sizes and whether the student network is initialized with a pre-trained model that predicts the ground truth

Table 5: Comparison of Pearson correlation coefficients of acoustic features associated with emotions between prompt and synthesized speech.

Model	Pitch mean	Pitch standard deviation	Energy mean	Energy standard deviation	Harmonics-to-noise ratio	Speaking rate	Jitter	Shimmer
Ground Truth	0.86	0.35	0.56	0.63	0.68	0.38	0.36	0.62
NaturalSpeech 2	0.88	0.41	0.81	0.40	0.83	0.03	0.57	0.74
FlashSpeech	0.89	0.02	0.83	0.23	0.67	0.03	0.64	0.49
VoiceCraft	**0.96**	**0.69**	0.63	0.33	0.74	0.10	0.72	0.76
XTTSv2	0.93	0.39	0.70	0.26	0.73	0.45	0.84	0.63
StyleTTS-ZS (LT)	0.94	0.53	0.74	0.32	**0.84**	0.26	0.76	0.77
StyleTTS-ZS (LL)	**0.96**	0.53	**0.88**	**0.70**	0.81	**0.46**	**0.88**	**0.81**

time-varying style conditioned on the input text $\boldsymbol{t}$ and prompt $\boldsymbol{x}'$. We use the ℓ_1 distance of the decoded duration ($\mathcal{L}_{\text{dur}}$), pitch ($\mathcal{L}_{f0}$) and energy ($\mathcal{L}_n$) between teacher and student models from the same input noise conditioned on unseen text and speaker prompts as the metric to measure the distillation performance. We tested the model performance using the guidance $\omega = 5$ as during our inference process.

For the consistency distillation (CD) baseline, we follow Algorithm 1 in (Luo et al., 2023) for guided distillation. We use the DDIM solver with noise schedule specified in section 3.3, ℓ_1 norm for distance, EMA rate $\mu = 0.999943$ as in (Song et al., 2023) and guidance scale range $\omega_{\text{min}} = 1, \omega_{\text{max}} = 15$ as in equation 6. For the adversarial diffusion distillation (ADD) baseline, we use the architecture of multimodal prosody discriminator as the discriminator that takes the denoised time-varying style as input and text embeddings $\boldsymbol{t}$ and prompt $\boldsymbol{x}'$ as conditions. We set $\tau_n = 200$ as our simulation-based method uses 100 steps and the student step $N = 4$ as in (Sauer et al., 2023). We initialized the student network with the teacher network for both CD and ADD baselines as in (Sauer et al., 2023). For the case of the simulation-based approach without pre-trained initialization, we also initialized our student network with the teacher network for fair comparison. We trained our model for 100k steps with a batch size of 32.

As shown in Table 6, our simulation-based approach with only 10k samples achieves better results than both CD and ADD baselines and much lower perceptual discrepancies compared to student models without the pre-trained initialization. This is likely because the classifier-free guidance is not easily approximated with bootstrapping-based methods, as their effects depend on fine-grained trajectories with small step sizes. These bootstrapping-based methods are useful for distilling large-scale diffusion models that are expensive to run sampling processes to get samples directly. However, since our latent is a fixed-size 50×512 vector, it is straightforward to run simulations and obtain enough samples to cover the latent space, particularly with our novel initialization approach with pre-trained networks that directly predict the ground truth latent.

B Evaluation Details

B.1 Baseline Models and Samples

In this section, we provide brief introductions to each baseline model and our methods to obtain samples to conduct our evaluations.

- **Vall-E**: Vall-E (Wang et al., 2023a), a previous SOTA model trained on 60k hours of LibriLight data, is the first large-scale zero-shot TTS model using language modeling techniques. Since it is not publicly available, we obtained 40 samples from the authors along with text transcriptions and 3-second prompts to synthesize speech for comparison. These samples were also used to calculate speaker embedding similarity (SIM) and word error rate (WER).

- **NaturalSpeech 2**: NaturalSpeech 2 (Shen et al., 2024), trained on the 44k-hour Multilingual LibriSpeech (MLS) dataset, excels in high-fidelity zero-shot TTS. We obtained 40 samples with the same texts and 3-second prompts as Vall-E for comparison. SIM and WER were computed from these samples.

- **FlashSpeech**: FlashSpeech (Ye et al., 2024), the current SOTA TTS model with fast inference, was trained on the MLS dataset. We obtained

Table 6: Comparison of $\mathcal{L}_{\text{dur}}$, $\mathcal{L}_{f0}$ and $\mathcal{L}_n$ with various distillation methods and sample size. The method used in our experiments is highlighted in italics.

Method	Sample size	$\mathcal{L}_{\text{dur}}$	$\mathcal{L}_{f0}$	$\mathcal{L}_n$
Consistency distillation (Song et al., 2023)	—	0.34	0.83	0.36
Adversarial diffusion distillation (Sauer et al., 2023)	—	0.29	0.77	0.24
Simulation-based (w/ pre-trained initialization)	1k	0.37	1.07	0.34
Simulation-based (w/ pre-trained initialization)	5k	0.29	0.93	0.29
Simulation-based (w/ pre-trained initialization)	*10k*	*0.18*	*0.64*	*0.18*
Simulation-based (w/ pre-trained initialization)	30k	0.16	0.52	0.13
Simulation-based (w/ pre-trained initialization)	50k	0.13	**0.43**	0.09
Simulation-based (w/ pre-trained initialization)	100k	**0.11**	0.44	**0.07**
Simulation-based (w/o pre-trained initialization)	1k	0.65	1.74	0.59
Simulation-based (w/o pre-trained initialization)	5k	0.58	1.63	0.52
Simulation-based (w/o pre-trained initialization)	10k	0.53	1.35	0.44
Simulation-based (w/o pre-trained initialization)	30k	0.44	1.02	0.29
Simulation-based (w/o pre-trained initialization)	50k	0.38	0.81	0.24
Simulation-based (w/o pre-trained initialization)	100k	0.24	0.67	0.18

40 samples from the authors, using the same texts and 3-second prompts as Vall-E and NaturalSpeech 2. SIM and WER were computed for comparison.

- **NaturalSpeech 3**: NaturalSpeech 3 (Ju et al., 2024), trained on LibriLight with factorized codec and diffusion models, achieves near-human performance in speaker similarity. We used 40 samples from the authors, matching the texts and 3-second prompts provided by Vall-E and NaturalSpeech 2 for comparison. SIM and WER were also computed.

- **VoiceCraft**: VoiceCraft (Peng et al., 2024), an autoregressive model trained on the GigaSpeech dataset, shows high speaker similarity for real-world prompts. We synthesized 40 samples using the same texts and 3-second prompts as Vall-E and others. The model is publicly available at `https://github.com/jasonppy/VoiceCraft`.

- **XTTSv2**: XTTSv2 (Casanova et al., 2022), trained on various public datasets, is a strong baseline for zero-shot TTS. We synthesized 40 samples with the same texts and 3-second prompts as used by other models. The model is publicly available at `https://huggingface.co/coqui/XTTS-v2`.

- **StyleTTS 2**: StyleTTS 2 (Li et al., 2024a), a SOTA model for fast and high-quality zero-shot TTS, was evaluated with 40 samples synthesized using the same texts and 3-second prompts as others. The model and LibriTTS checkpoint are available at `https://github.com/yl4579/StyleTTS2` and `https://huggingface.co/yl4579/StyleTTS2-LibriTTS/tree/main`.

- **HierSpeech++**: HierSpeech++ (Lee et al., 2023) is a strong baseline trained on limited data, achieving high speaker similarity. We synthesized 40 samples using the same texts and 3-second prompts for comparison with the *LibriTTS-train-960* checkpoint. The model is publicly available at `https://github.com/sh-lee-prml/HierSpeechpp`.

B.2 Subjective Evaluations

To ensure high-quality evaluation from MTurk, we followed (Li et al., 2024a) by enabling the following filters on MTurk:

- HIT Approval Rate (%) for all Requesters' HITS: `greater than 95`.

- Location: `is UNITED STATES (US)`.

- Number of HITs Approved: `greater than 50`.

We provided the following instructions for rating the naturalness and similarity of our MOS experiments following (Li et al., 2024a):

- Naturalness:

```
Some of them may be synthesized while
others may be spoken by an American
audiobook narrator.

Rate how natural each audio clip
```

```
sounds on a scale of 1 to 5 with 1
indicating completely unnatural
speech (bad) and 5 completely natural
speech (excellent).

Here, naturalness includes whether you
feel the speech is spoken by a native
American English speaker from a human
source.
```

- Similarity:

```
Rate whether the two audio clips could
have been produced by the same speaker
or not on a scale of 1 to 5 with 1
indicating completely different
speakers and 5 indicating exactly the
same speaker.

Some samples may sound somewhat
degraded/distorted; for this question,
please try to listen beyond the
distortion of the speech and
concentraten on identifying the voice
(including the person's accent and
speaking habits, if possible).
```

An example survey used for our MOS evaluation can be found at https://survey.alchemer.com/s3/7858362/styletts-new-MOS-521 [4].

For CMOS experiments, since all the models have high-quality synthesis results, we removed the instruction regarding audio distortion and used the following instructions instead:

- Naturalness:

```
Some of them may be synthesized while
others may be spoken by an American
audiobook narrator.

Rate how natural is B compared to A on
a scale of -6 to 6, with 6 indicating
that B is much better than A.

Here, naturalness includes whether
you feel the speech is spoken by a
native American English speaker from
a human source.
```

[4] An example survey used for our CMOS evaluation is available at https://survey.alchemer.com/s3/7854889/CMOS-stylettsz-flashspeech-librispeech-0519.

- Similarity:

```
Rate how similar is the speaker in B
to the reference voice, compared to A.
Here, "similar" means that you feel
that the recording and the reference
voice are produced by the same speaker.
```

We ensured the survey quality by applying additional attention checking following (Li et al., 2024a). In the MOS assessment, we utilized the average score given by a participant to ground truth audios, unbeknownst to the participants, to ascertain their attentiveness. We excluded ratings from those whose average score for the ground truth did not rank in the top three among all five models. In the CMOS evaluation, we checked the consistency of the rater's scores: if the score's sign (indicating whether A was better or worse than B) differed for over half the sample set, the rater was disqualified. Four raters were eliminated through this process in all of our experiments.

For the CMOS experiments with (Wang et al., 2023a; Shen et al., 2024; Ye et al., 2024; Peng et al., 2024) and ground truth, since we have 40 samples, we divided it into two batches with 20 samples each. We recruited 10 raters for the first batch, which consisted of 20 pairs of samples, and obtained 200 ratings from this batch. We then launched a second batch with another 20 pairs of samples and obtained another 200 ratings.

The raters were paid \$3 for completing the 10-minute CMOS survey and \$8 for completing the 25-minute MOS survey on MTurk.

C Training Objectives

In this section, we provide detailed training objectives for our acoustic synthesizer and prosody autoencoder, as the training objective for style diffusion is provided in Section 3.3.

C.1 Acoustic Synthesizer Training

In this section, we follow the notation of (Li et al., 2024a) where the decoder $G(\cdot)$ takes upsampled text embeddings $\boldsymbol{h}_{\text{text}} \cdot \boldsymbol{a}$, pitch $p_{\boldsymbol{x}}$, energy $n_{\boldsymbol{x}}$, and global style $\boldsymbol{s}$ instead of five inputs as in Section 3.1 for consistency with (Li et al., 2024a). In section 3.1, we use duration $d_{\boldsymbol{x}}$ as a compact representation for the phoneme-speech alignment $\boldsymbol{a}$, which is what is being used in the actual implementation. $\boldsymbol{a}$ can be obtained by repeating the value 1 for d_i times at ℓ_{i-1}, where ℓ_i is the end position of the i^{th} phoneme $\boldsymbol{t}_i$ calculated by summing d_k for $k \in \{1, \ldots, i\}$, and d_i is the duration of $\boldsymbol{t}_i$.

Mel-spectrogram reconstruction. The decoder is trained on waveform $\boldsymbol{y}$, its corresponding mel-spectrogram $\boldsymbol{x}$, a speech prompt mel-spectrogram $\boldsymbol{x}'$ which is a small chunk of $\boldsymbol{x}$, and the text $\boldsymbol{t}$, using L_1 reconstruction loss as

$$\mathcal{L}_{\text{mel}} = \mathbb{E}_{\boldsymbol{x},\boldsymbol{t}}\left[\left\|\boldsymbol{x} - M\left(G\left(\boldsymbol{h}_{\text{text}} \cdot \boldsymbol{a}_{\text{algn}}, \boldsymbol{s}, p_{\boldsymbol{x}}, n_{\boldsymbol{x}}\right)\right)\right\|_1\right]. \tag{8}$$

Here, $\boldsymbol{h}_{\text{text}}, \boldsymbol{s} = T(\boldsymbol{t}, \boldsymbol{x}')$ is the encoded phoneme representation aligned with the prompt $\boldsymbol{x}'$ and global style of $\boldsymbol{x}'$, and the attention alignment is denoted by $\boldsymbol{a}_{\text{algn}} = A(\boldsymbol{x}, \boldsymbol{t})$. $p_{\boldsymbol{x}}$ is the pitch F0 and $n_{\boldsymbol{x}}$ indicates energy of $\boldsymbol{x}$, which are extracted by a pitch extractor $p_{\boldsymbol{x}} = F(\boldsymbol{x})$, and $M(\cdot)$ represents mel-spectrogram transformation. Following (Li et al., 2022), half of the time, raw attention output from A is used as alignment, allowing back-propagation through the text aligner. For another 50% of the time, a monotonic version of $\boldsymbol{a}_{\text{algn}}$ is utilized via dynamic programming algorithms (see Appendix A in (Li et al., 2022)).

TMA objectives. We follow (Li et al., 2022) and use the original sequence-to-sequence ASR loss function $\mathcal{L}_{\text{s2s}}$ to fine-tune the pre-trained text aligner, preserving the attention alignment during end-to-end training:

$$\mathcal{L}_{\text{s2s}} = \mathbb{E}_{\boldsymbol{x},\boldsymbol{t}}\left[\sum_{i=1}^{N} \mathbf{CE}(\boldsymbol{t}_i, \hat{\boldsymbol{t}}_i)\right], \tag{9}$$

where N is the number of phonemes in $\boldsymbol{t}$, $\boldsymbol{t}_i$ is the i-th phoneme token of $\boldsymbol{t}$, $\hat{\boldsymbol{t}}_i$ is the i-th predicted phoneme token, and $\mathbf{CE}(\cdot)$ denotes the cross-entropy loss function.

Additionally, we apply the monotonic loss $\mathcal{L}_{\text{mono}}$ to ensure that soft attention approximates its non-differentiable monotonic version:

$$\mathcal{L}_{\text{mono}} = \mathbb{E}_{\boldsymbol{x},\boldsymbol{t}}\left[\left\|\boldsymbol{a}_{\text{algn}} - \boldsymbol{a}_{\text{hard}}\right\|_1\right], \tag{10}$$

where $\boldsymbol{a}_{\text{hard}}$ is the monotonic version of $\boldsymbol{a}_{\text{algn}}$ obtained through dynamic programming algorithms (see Appendix A in (Li et al., 2022) for more details).

Adversarial objectives. Two adversarial loss functions, originally used in HifiGAN (Kong et al., 2020), are employed to enhance the sound quality of the reconstructed waveforms: the LSGAN loss function $\mathcal{L}_{\text{adv}}$ for adversarial training and the feature-matching loss $\mathcal{L}_{\text{fm}}$.

$$\mathcal{L}_{\text{adv}}(G; D) = \mathbb{E}_{\boldsymbol{t},\boldsymbol{x}}\left[\left(D\left(\left(G\left(\boldsymbol{h}_{\text{text}} \cdot \boldsymbol{a}_{\text{algn}}, \boldsymbol{s}, p_{\boldsymbol{x}}, n_{\boldsymbol{x}}\right)\right); \mathcal{C}\right) - 1\right)^2\right] \tag{11}$$

$$\mathcal{L}_{\text{adv}}(D; G) = \mathbb{E}_{\boldsymbol{t},\boldsymbol{x}}\left[\left(D\left(\left(G\left(\boldsymbol{h}_{\text{text}} \cdot \boldsymbol{a}_{\text{algn}}, \boldsymbol{s}, p_{\boldsymbol{x}}, n_{\boldsymbol{x}}\right)\right)\right); \mathcal{C}\right)^2\right] + \mathbb{E}_{\boldsymbol{y}}\left[\left(D(\boldsymbol{y}; \mathcal{C}) - 1\right)^2\right], \tag{12}$$

$$\mathcal{L}_{\text{fm}} = \mathbb{E}_{\boldsymbol{y},\boldsymbol{t},\boldsymbol{x}}\left[\sum_{l=1}^{\Lambda} \frac{1}{N_l}\left\|D^l(\boldsymbol{y}; \mathcal{C}) - D^l\left(\hat{\boldsymbol{y}}; \mathcal{C}\right)\right\|_1\right], \tag{13}$$

where $\hat{\boldsymbol{y}} = G\left(\boldsymbol{h}_{\text{text}} \cdot \boldsymbol{a}_{\text{algn}}, \boldsymbol{s}, p_{\boldsymbol{x}}, n_{\boldsymbol{x}}\right)$ is the generated waveform and D represents both MPD, MRD, and multimodal waveform discriminator (MMWD). $\mathcal{C}$ denotes the conditional input to MMWD (see Section 3.4) and is $\emptyset$ for MPD and MRD. Λ is the total number of layers in D, and D^l denotes the output feature map of l-th layer with N_l number of features.

Speaker Embedding Feature Matching Loss. We compute the intermediate features of a ResNet-based speaker embedding model (Wang et al., 2023b) V for the following loss:

$$\mathcal{L}_{\text{fm}} = \mathbb{E}_{\boldsymbol{y},\boldsymbol{t},\boldsymbol{x}}\left[\sum_{l=1}^{\Lambda} \frac{1}{N_l}\left\|V^l(\boldsymbol{y}) - V^l\left(\hat{\boldsymbol{y}}\right)\right\|_1\right], \tag{14}$$

$\hat{\boldsymbol{y}} = G\left(\boldsymbol{h}_{\text{text}} \cdot \boldsymbol{a}_{\text{algn}}, \boldsymbol{s}, p_{\boldsymbol{x}}, n_{\boldsymbol{x}}\right)$ is the generated waveform, Λ is the total number of layers in V, V^l denotes the output feature map of l-th layer with N_l number of features.

Acoustic synthesizer full objectives. Our full objective functions in acoustic synthesizer training can be summarized as follows with hyperparameters λ_{s2s} and λ_{mono}:

$$\min_{G,A,T,F} \mathcal{L}_{\text{mel}} + \lambda_{\text{s2s}}\mathcal{L}_{\text{s2s}} + \lambda_{\text{mono}}\mathcal{L}_{\text{mono}} + \mathcal{L}_{\text{adv}}(G; D) + \mathcal{L}_{\text{fm}} \tag{15}$$

$$\min_{D} \mathcal{L}_{\text{adv}}(D; G) \tag{16}$$

Following (Li et al., 2022), we set $\lambda_{\text{s2s}} = 0.2$ and $\lambda_{\text{mono}} = 5$.

C.2 Prosody Autoencoder Training

Duration reconstruction. We employ the L-1 loss to reconstruct the duration:

$$\mathcal{L}_{\text{dur}} = \mathbb{E}_{\boldsymbol{x}}\left[\left\|d_{\boldsymbol{x}} - \hat{d}_{\boldsymbol{x}}\right\|_1\right], \tag{17}$$

where $\hat{d}_{\boldsymbol{x}} = \text{PD}_d(\text{PAE}(d_{\boldsymbol{x}}, p_{\boldsymbol{x}}, n_{\boldsymbol{x}}), \boldsymbol{t})$ is the decoded duration conditioned on $\boldsymbol{t}$ from the duration decoder $\text{PD}_d(\cdot)$ after being encoded by the prosody encoder $\text{PAE}(\cdot)$.

Prosody reconstruction. We use $\mathcal{L}_{f_0}$ and $\mathcal{L}_n$, which are F0 and energy reconstruction loss, respectively:

$$\mathcal{L}_{f0} = \mathbb{E}_{\boldsymbol{x}}\left[\|p_{\boldsymbol{x}} - \hat{p}_{\boldsymbol{x}}\|_1\right] \tag{18}$$

$$\mathcal{L}_{n} = \mathbb{E}_{\boldsymbol{x}}\left[\|n_{\boldsymbol{x}} - \hat{n}_{\boldsymbol{x}}\|_1\right] \tag{19}$$

where $\hat{p}_{\boldsymbol{x}}, \hat{n}_{\boldsymbol{x}} = \text{PD}_p(\text{PAE}(d_{\boldsymbol{x}}, p_{\boldsymbol{x}}, n_{\boldsymbol{x}}), \boldsymbol{t})$ are the decoded pitch and energy of $\boldsymbol{x}$ conditioned on $\boldsymbol{t}$ from the pitch and energy decoder $\text{PD}_p(\cdot)$.

Adversarial objectives. We employed similar adversarial objectives as during the acoustic synthesizer training:

$$\mathcal{L}_{\text{adv}}(G;D) = \mathbb{E}_{\boldsymbol{t},\boldsymbol{x}}\left[\left(D\left(\left(G\left(\mathcal{P}\right)\right);\mathcal{C}\right) - 1\right)^2\right], \tag{20}$$

$$\begin{aligned}\mathcal{L}_{\text{adv}}(D;G) = \; &\mathbb{E}_{\boldsymbol{t},\boldsymbol{x}}\left[\left(D\left(\left(G\left(\mathcal{P}\right)\right)\right);\mathcal{C}\right)^2\right] \\ &+ \mathbb{E}_{\boldsymbol{y}}\left[\left(D(\mathcal{P};\mathcal{C}) - 1\right)^2\right],\end{aligned} \tag{21}$$

$$\begin{aligned}&\mathcal{L}_{\text{fm}} \\ &= \mathbb{E}_{\boldsymbol{y},\boldsymbol{t},\boldsymbol{x}}\left[\sum_{l=1}^{\Lambda}\frac{1}{N_l}\left\|D^l(\mathcal{P};\mathcal{C}) - D^l\left(G\left(\mathcal{P}\right);\mathcal{C}\right)\right\|_1\right],\end{aligned} \tag{22}$$

where D represents the multimodal prosody discriminator (MMPD) for duration and pitch and energy and G represents the combined prosody encoder and decoder $\text{PD}(\text{PAE}(\cdot), \boldsymbol{t})$. $\mathcal{C}$ denotes the conditional input to MMPD and $\mathcal{P}$ denotes either duration, pitch or energy. Λ is the total number of layers in D, and D^l denotes the output feature map of l-th layer with N_l number of features.

Prosody autoencoder training full objectives. Our full objective functions in joint training can be summarized as follows with hyperparameters λ_{dur}, λ_{f0}, and λ_n:

$$\begin{aligned}\min_{G,D}\; &\mathcal{L}_{\text{dur}} + \lambda_{\text{f0}}\mathcal{L}_{\text{f0}} + \lambda_{\text{n}}\mathcal{L}_{\text{n}} \\ &+ \mathcal{L}_{\text{adv}}(G;D) + \mathcal{L}_{\text{fm}}\end{aligned} \tag{23}$$

$$\min_{D}\; \mathcal{L}_{\text{adv}}(D;G) \tag{24}$$

where G represents both prosody encoder and decoder. Following (Li et al., 2024a), we set $\lambda_{f0} = 0.1$ and $\lambda_n = 1$.

D Model Architectures

This section provides a detailed outline of StyleTTS-ZS architecture. We keep the same architecture for the waveform decoder, duration extractor (or text aligner in (Li et al., 2024a)), and pitch extractor as in (Li et al., 2024a). The prosodic text encoder is a pre-trained PL-BERT (Li et al., 2023b) available at https://github.com/yl4579/PL-BERT. Additionally, we adopt the same acoustic discriminators as in (Kumar et al., 2024). This section focuses primarily on our new proposed components as outlined in Figure 2.

Table 7: Prompt-text encoder architecture. N represents the input phoneme length of the mel-spectrogram and T represents the prompt length, $\boldsymbol{t}$ represents the input phonemes with shape $1 \times N$, $\boldsymbol{x}'$ is the speech prompt mel-spectrogram with shape $80 \times T$.

Submodule	Input	Layer	Output Shape
Embedding	$\boldsymbol{t}$	Phoneme Embedding	$512 \times N$
	$\boldsymbol{x}'$	Linear 80×512	$512 \times T$
	–	Concat	$512 \times (N+T)$
Conformer Block $(\times 1)$	–	# of head: 8, head features: 64, kernel size: 31, feedforward dimension: 1024	$512 \times (N+T)$
Conformer Block $(\times 5)$	–	# of head: 8, head features: 64, kernel size: 15, feedforward dimension: 1024	$512 \times (N+T)$
Output	–	Split w.r.t. $\boldsymbol{t}$ and $\boldsymbol{x}'$	$512 \times N$ $512 \times T$
	–	Adaptive Average Pool w.r.t. $\boldsymbol{x}'$	512×1
	–	Linear 512×512 w.r.t. $\boldsymbol{t}$	$512 \times T$

Table 8: Prosody encoder architecture. T represents the length of pitch p and energy n with shape $1 \times T$, while N represents the length of duration d with shape $1 \times N$. k_l represents position inputs from $\{1, \dots, l\}$.

Submodule	Input	Layer	Output Shape
Input	d, k_T	Position Upsampling with d and Embedding	$512 \times T$
	p, n	Concat	$514 \times T$
	–	Linear 514×512	$512 \times T$
Conformer Block $(\times 1)$	–	# of heads: 8, head features: 64, kernel size: 31, feedforward dimension: 1024	$512 \times T$
Conformer Block $(\times 5)$ (output denoted as $\boldsymbol{h}_{vl}$)	–	# of heads: 8, head features: 64, kernel size: 15, feedforward dimension: 1024	$512 \times T$
Output	k_{50}	Embedding	512×50
	$\boldsymbol{h}_{vl}$	8-Head Cross Attention with 64 head features	512×50

Table 9: Prosody decoder architecture. $\boldsymbol{t}$ represents the input text embeddings from PL-BERT with size $512 \times N$ where N is the text length. $\boldsymbol{h}$ represents the time-varying style output from the prosody encoder with size 512×50. $k = 2$ for pitch and energy decoder and $k = 1$ for duration decoder.

Submodule	Input	Layer	Output Shape
Input	$\boldsymbol{t}, \boldsymbol{h}$	Concat	$512 \times (N+50)$
Conformer Block $(\times 6)$	–	# of heads: 8, head features: 64, kernel size: 31, feedforward dimension: 1024	$512 \times (N+50)$
Output	–	Truncate w.r.t. $\boldsymbol{t}$	$512 \times N$
	–	Linear $512 \times k$	$k \times N$

Table 10: Style diffusion denoiser architecture. We used the same architecture for the distilled student model. $\boldsymbol{t}$ represents the input text embeddings from PL-BERT with size $512 \times N$ where N is the text length and $\boldsymbol{x}'$ is the prompt speech with size $80 \times T$. $\boldsymbol{\xi}$ represents the noisy input with the size 512×50. σ represents either the noise level (for the teacher diffusion model) or the guidance scale (for the distilled student model). During pre-training of student model, $\sigma = 0$ and $\boldsymbol{\xi} = 0$.

Submodule	Input	Layer	Output Shape
Input	$\boldsymbol{x}'$	Linear 80×512	512 $\times N$
	$\boldsymbol{t}, \boldsymbol{\xi}$	Concat as output $\boldsymbol{x}$	512 $\times (T + N + 50)$
Embedding	$\boldsymbol{\sigma}$	Sinusoidal Embedding	512 $\times 1$
	–	Repeat	512 $\times (T + N + 50)$
	$\boldsymbol{x}$	Addition	512 $\times (T + N + 50)$
Conformer Block $(\times 2)$	–	# of head: 8, head features: 64, kernel size: 31, feedforward dimension: 1024	512 $\times (N + T + 50)$
Conformer Block $(\times 10)$	–	# of head: 8, head features: 64, kernel size: 15, feedforward dimension: 1024	512 $\times (N + T + 50)$
Output	–	Truncate w.r.t. $\boldsymbol{\xi}$	512×50

Table 11: Multimodal discriminator architecture. $\boldsymbol{x}$ represents the decoder output with shape $d \times N$ and $\mathcal{C}$ represents conditions for the discriminator with shape $k \times T$.

Submodule	Input	Layer	Output Shape
Input	$\boldsymbol{x}$	Linear $d \times 512$	$512 \times N$
	$\mathcal{C}$	Linear $k \times 512$	$512 \times T$
	–	Concat	$512 \times (T + N)$
Conformer Block $(\times 6)$	–	# of heads: 8, head features: 64, kernel size: 15, feedforward dimension: 1024	512 $\times (T + N)$
Output	–	Linear 1024×1	$1 \times (T + N)$