VoiceTuner: Self-Supervised Pre-training and Efficient Fine-tuning For Voice Generation

Anonymous ACL submission

Abstract

Voice large language models (LLMs) cast voice synthesis as a language modeling task in a discrete space, and have demonstrated significant progress to date. Despite the recent success, the current development of voice LLMs in low-resource applications is hampered by data scarcity and high computational cost. In this work, we propose VoiceTuner, with a self-supervised pre-training and efficient fine-tuning approach for low-resource voice generation. Specifically, 1) to mitigate data scarcity, we leverage large-scale unlabeled dataset and pre-train VoiceTuner-SSL without pre-defined applications, which can be fine-tuned in downstream tasks; 2) to further reduce the high training cost in complete fine-tuning, we introduce a multiscale adapter to effectively update around only 1% parameters as a plug-and-play module; and 3) to alleviate the challenges of modeling long audio tokens inherited from inefficient attention mechanism, we introduce VoiceTuner-Mamba with multiscale state space models in place of transformers. Experimental results demonstrate that VoiceTuner-SSL presents strong acoustic continuations. VoiceTuner exhibits superior quality and style similarity in three low-resource (1h, 10h, 30h) generation tasks. [1]

1 Introduction

Current voice large language models (LLMs) (Kharitonov et al., 2023; Wang et al., 2023; Zhang et al., 2023b) cast voice synthesis as a language modeling task in a discrete representation space. VALL-E (Wang et al., 2023) proposes a language model approach for text-to-speech (TTS) with audio codec tokens. UniAudio (Yang et al., 2023) introduces a multi-scale transformer to enable sub-quadratic self-attention, unlocking better performance at a reduced cost for training and generation. A line of works (Kharitonov et al., 2023; Borsos et al., 2022; Agostinelli et al., 2023) introduces the hierarchical approach that combines semantic and acoustic audio tokens to decrease supervision in model training.

Despite the success achieved, the current development of voice LLMs in low-resource scenarios is hampered by two major challenges: 1) data scarcity: most existing models rely on web-scale training data, which are lacking in low-resource scenarios; and 2) high computational cost: training voice LLMs from scratch are computationally intensive and time-consuming, and the inefficient attention mechanism in transformer further challenges model in modeling long codec sequence.

In this work, we propose VoiceTuner, with a self-supervised pre-training and efficient fine-tuning approach for low-resource voice generation. To alleviate data scarcity, we pre-train the next-token prediction model (VoiceTuner-SSL) in the large-scale unlabeled dataset, which can be fine-tuned in downstream generation tasks with reduced data and device requirements. To further reduce computational cost and avoid losing the general abilities of VoiceTuner-SSL, we introduce an efficient multiscale adapter to fine-tune only around 1% parameters in downstream applications. To alleviate the challenges of modeling long audio tokens inherited from inefficient attention mechanisms, we introduce VoiceTuner-Mamba with state space models in place of transformers, effectively reducing the quadratic complexity to linear.

VoiceTuner is pre-trained on ∼160K hours of unlabeled voice data without supervision, followed by rich or low resource (1h, 10h, and 30h) adaptation in downstream applications including zero-shot TTS, singing voice synthesis, and instruction TTS, respectively generalizing to unseen speaker, modality, and instruction. Experimental results demonstrate that VoiceTuner-SSL keeps acoustic continuations, maintaining speaker identity, emo-

[1] Audio samples are available at https://VoiceTuner.github.io

tion, and speaking speed from prompts. VoiceTuner exhibits superior audio quality and style similarity, unlocking the ability to generate voice samples in low-resource scenarios. Furthermore, VoiceTuner-Mamba with state space models is more efficient in terms of GPU memory and floating point operations (FLOPS) for modeling extremely long audio tokens. The key takeaways are as follows:

- We present VoiceTuner, with a self-supervised pre-training and fine-tuning approach to alleviate data scarcity in low-resource applications.
- We introduce a lightweight multiscale adapter to efficiently fine-tune only around 1% parameters, further reducing the computational cost.
- We investigate replacing the inefficient transformers with state space models, which reduces the complexity of modeling long audio tokens.
- Experimental results demonstrate that VoiceTuner-SSL keeps acoustic continuations, and present VoiceTuner's superior audio quality and style similarity.

2 Related Works

2.1 Generative Voice Models

Text-guided voice synthesis (text-to-speech and singing voice synthesis) typically converts input text into mel-spectrogram (e.g., Tacotron (Wang et al., 2017), FastSpeech (Ren et al., 2019)), which is then transformed to waveform using a separately trained vocoder (Kong et al., 2020; Huang et al., 2021). Recent generative models cast voice synthesis as a language modeling task to perform in-context learning: VALL-E (Wang et al., 2023) uses discrete codes derived from an off-the-shelf neural audio codec model, and regards TTS as a conditional language model. Zhang et al. (2023b) leverage back-translation and prompt-guided LLMs for high-quality TTS with limited supervision. Jiang et al. (2023) train a prosody language model with arbitrary-length speech prompts to produce expressive and controlled prosody. However, these existing voice LLMs are trained from scratch using web-scale data, and replicating this success is limited in low-resource scenarios.

2.2 State Space Models

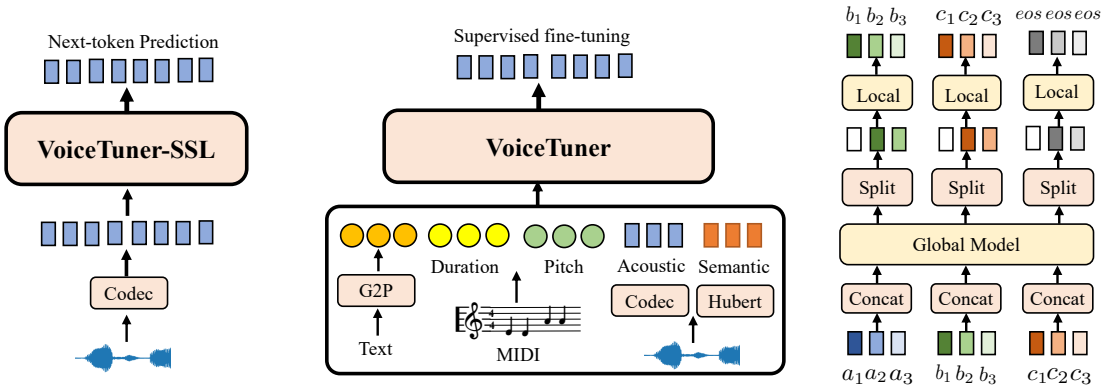
State space models are recently introduced into deep learning as state space transforming (Gu et al., 2021b,a; Smith et al., 2022). Mamba (Gu and Dao, 2023) integrates selective SSMs into a simplified end-to-end neural network architecture without attention or even MLP blocks. Vision mamba (Zhu et al., 2024) compresses the visual representation with bidirectional state space and proposes a new generic vision backbone with bidirectional Mamba blocks. VMamba (Liu et al., 2024) achieves linear complexity without sacrificing global receptive fields and introduces the cross-scan Module (CSM) to traverse the spatial domain. Inspired by these, we present the end-to-end differentiable multiscale state space models to effectively reduce the inherited attention complexity in voice LLMs.

2.3 Generative Voice Pre-training and Fine-tuning

Self-supervised learning (SSL) (Baevski et al., 2020; Hsu et al., 2021) has been shown to achieve remarkable advances in recent years, opening up a wide array of applications that leverage their power by adapting models. AudioLDM 2 (Liu et al., 2023b) leverages AudioMAE (Huang et al., 2022a) and performs self-supervised audio generation learning with a latent diffusion model conditioned on audio tokens. UniAudio (Yang et al., 2023) trains on different generative tasks to obtain prior knowledge in the inter-relationship between audio and other modalities and support new audio generation tasks after simple fine-tuning. Liu et al. (2023a) achieve better performance utilizing low-rank adaptation (LoRA), which adds the linear input projection to each self-attention layer. Vyas et al. (2023) include two-stage full fine-tuning to improve model fidelity and quality where all parameters are optimized together. In this work, we introduce a multiscale adapter for parameter-efficient adaptation, which updates only around 1% of the parameters on top as a lightweight plug-and-play module.

3 Method

In this section, we overview the discrete voice representation - acoustic tokens and then introduce the proposed generative self-supervised pre-training and follow-up fine-tuning approach, respectively with VoiceTuner-SSL and VoiceTuner. Next, we propose a lightweight, plug-and-play adapter for parameter-efficient fine-tuning. In the following, we introduce the scalable global and local architecture in Section 3.4 and provide the preliminaries of VoiceTuner-Mamba with state space models (SSM)



(a) Generative pre-training (b) Efficient fine-tuning (c) Multi-scale architecture

Figure 1: In subfigure (b), prompts can be adjusted for different tasks with a variety of conditions (speaker, emotion, prosody, and style).

block (i.e., Mamba) in Section 3.5.

3.1 Speech Representation

Audio codec models such as SoundStream (Zeghidour et al., 2021) and Encodec (Défossez et al., 2022) have recently shown that encoder-decoder architecture excels at learning acoustic information in a self-supervised manner, where the representation can be used in a variety of generative tasks.

The acoustic codec model typically consists of an audio encoder, a residual vector-quantizer (RVQ), and an audio decoder: 1) The audio encoder E consists of several convolutional blocks with a total downsampling rate of 320 and generates continuous representations at every 20-ms frame in 16kHz. 2) The residual vector-quantizer Q produces discrete representations a_q with a codebook size of K_2, using a vector quantization layer (Vasuki and Vanathi, 2006). 3) The audio decoder G reconstructs the signal $\hat{y}$, from the compressed latent representation a_q. In the end, a speech utterance y is represented as acoustic tokens with $[a_1, a_2, \ldots, a_T], a_i \in \{0, 1, \ldots, K_2 - 1\}, \forall 1 \leq i \leq T$, where T is the number of frames.

3.2 Self-supervised Pre-training

Most voice LLMs rely on web-scale training data and cast voice synthesis as a language modeling task, while the data shortage hampers its application in low-resource scenarios. To alleviate it, we leverage unlabeled corpus and pre-train LLMs (namely VoiceTuner-SSL) in a next-token prediction task without supervision, where we hypothesize that a generative model without pre-defined application can be applied to different downstream tasks, reducing data requirement in low-resource application.

VoiceTuner-SSL is pre-trained on arbitrary voice, which contains many speakers with various accents, diverse demographics, and heterogeneous recording conditions. Next, we fine-tune VoiceTuner-SSL to align speech and text modalities utilizing supervised data in downstream voice generation applications, where we find that the self-supervised pre-training stage offers a distinct gain in both rich and low-resource scenarios. We expect our VoiceTuner-SSL to keep the speaker identity, prosody, and recording conditions of the prompt and produce new content. We refer the reader to Section 5.1 for our findings.

3.3 Efficient Fine-tuning

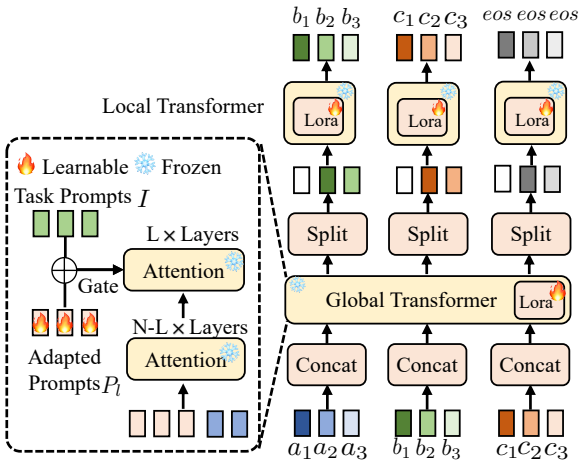


Figure 2: VoiceTuner: Efficient fine-tuning with multi-scale adapter.

Though fine-tuning voice LLMs is effective compared with training voice LLMs from scratch, a complete fine-tuning of large-scale voice LLMs still 1) is time-consuming, computation-intensive, multi-modality unsupported; and 2) can lose the

general ability of foundation model (e.g., acoustic continuations). In this section, we introduce an efficient plug-and-play module, i.e., a multiscale adapter, to update only around 1% parameters. Specifically,

- We include low-rank adaptation (LoRA) (Hu et al., 2021) in the linear input projection of each layer in attention blocks, where only the LoRA parameters are optimized.

- A set of learnable prompts with gates (Zhang et al., 2023a) are added to the input, which learn to adaptively inject new instructions (conditions) into the pre-trained model and avoid disturbing speech tokens at the beginning of training.

Suppose we have condition representation (i.e., task-specific prompts) $I \in \mathbb{R}^{K \times C}$ with length K and feature dimension C. For instruction TTS, we use pre-trained Flan-T5-XL (Raffel et al., 2020) and freeze the weights to derive condition representation; For zero-shot TTS and SVS, we use the token embedding matrix to obtain the representation of acoustic and pitch tokens from speaker and MIDI prompt, which are then pad to a fix length $K = 150$.

We initialize learnable adaption prompt $\{P_l\}_{l=1}^{L}$ for L layers, where we have each layer's prompt $P_l \in \mathbb{R}^{K \times C}$ and speech tokens $T_l \in \mathbb{R}^{M \times C}$. Then, the adaption prompt is conducted an element-wise addition with condition representation: $P_l = [P_l + I] \in \mathbb{R}^{K \times C}$.

Suppose the model is processing with the speech tokens T_l and condition P_l, The attention score related to learnable prompt is calculated as $S_l^p = \text{Attention}(T_l, P_l, P_l) = \text{Softmax}(T_l P_l^T / \sqrt{C}) P_l$, and we have S_l^t self-attention score for original speech tokens. A learnable gating factor g_l is adapted to adaptively control the importance of S_l^p in the attention with $S_l = S_l^p g_l + S_l^t$, which represents how much information the learnable prompt contributes. Initialized by zero, g_l can first eliminate the influence of under-fitted prompts and then increase its magnitude to provide more instruction semantics.

To conclude, the adaptation enjoys efficient training efficiency with only around 1% learnable parameters. As a lightweight plug-and-play module, this enables us to fine-tune voice LLMs on cheap devices.

3.4 Multiscale Architecture

VoiceTuner (denoted as θ_{AR}) predicts long sequences with end-to-end differentiable multiscale transformers similar to Yu et al. (2023); Yang et al. (2023). This enables sub-quadratic self-attention, unlocking better performance at reduced cost for both training and generation. As illustrated in Figure 1(c): 1) the token embedding matrix E_G maps integer-valued tokens $a_1, a_2, ..., c_2, c_3$ to m dimensional embeddings, following which 2) we chunk it into patches of size P of length $K = \frac{T}{P}$, 3) a large global transformer $\theta_{AR}^{\text{global}}$ module outputs patch representations $\mathbf{G_o^{1:K}} = \theta_{AR}^{\text{global}}(\mathbf{G_i^{0:K-1}})$, and 4) a relatively smaller local transformer $\theta_{AR}^{\text{local}}$ operates on a single patch containing P elements, each of which is the sum of an output from the global model and an embedding of the previous tokens, and autoregressively predict the next patch $\mathbf{L_o^{1:K}} = \theta_{AR}^{\text{local}}(\mathbf{L_i^{0:K-1}} + \mathbf{G_o^{1:K}})$.

VoiceTuner presents the improvements from scaling attention layers' depth and width without the requirement of scattered model-specific methodologies. As expected, scaling the model size (160M (base), 420M (medium), and 1.1B (large) parameter) results in better scores. We refer the reader to Section 5.5 for our findings.

3.5 State Space Model

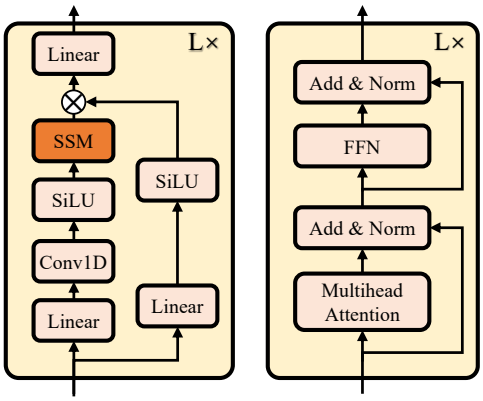


Figure 3: Left: Mamba block; Right: Transformer block. Mamba adds an SSM to the main branch.

To alleviate challenges of modeling long audio tokens in inefficient attention transformers, we introduce VoiceTuner-Mamba with state space models (Smith et al., 2022; Gu and Dao, 2023) in place of transformers, effectively reducing the quadratic complexity to linear. Illustrated in Figure 3, VoiceTuner-Mamba is built upon State Space Models (SSMs), which are considered linear time-invariant systems that map stimulation $x(t) \in \mathbb{R}^L$

to response $y(t) \in \mathbb{R}^L$ through a hidden state $h(t)$, where the parameters include $A \in \mathbb{C}^{N\times N}, B, C \in \mathbb{C}^N$:

$$\begin{aligned} h'(t) &= Ah(t) + Bx(t) \\ y(t) &= Ch(t) \end{aligned} \tag{1}$$

The S4 is the discrete version of the continuous system, which also includes a timescale parameter Δ to transform the continuous parameters A, B to discrete spaces:

$$\begin{aligned} \bar{A} &= e^{\Delta A}, \bar{B} = \left(e^{\Delta A} - I\right) A^{-1} B \\ h_t &= \bar{A}h_{t-1} + \bar{B}x_t, y_t = Ch_t \end{aligned} \tag{2}$$

Mamba (Gu and Dao, 2023) further incorporate the selective scan mechanism (S6), where the matrices $B \in \mathbb{R}^{B\times L\times N}, C \in \mathbb{R}^{B\times L\times N}, \Delta \in \mathbb{R}^{B\times L\times D}$ are derived from the input data. S6 is aware of the contextual information embedded in the input, ensuring the dynamism of weights within this mechanism. In contrast to the conventional attention computation approach, S6 enables each element in a 1-D array (e.g., sequence) to interact with any of the previously scanned samples through a compressed hidden state. VoiceTuner-Mamba refrains from utilizing position embedding bias due to the causal nature.

To compare VoiceTuner-Mamba and VoiceTuner, we developed VoiceTuner-Mamba in three distinct scales (small, medium, and large) with similar parameters. The resulting architecture is a versatile replacement for VoiceTuner in different applications.

3.6 Reconstructing High-Fidelity Waveforms

We train a unit-based neural vocoder from scratch for the acoustic unit to waveform generation. Inspired by BigVGAN (Lee et al., 2022), the synthesizer includes the generator and multi-resolution discriminator (MRD). The generator is built from a set of look-up tables (LUT) that embed the discrete representation and a series of blocks composed of transposed convolution and a residual block with dilated layers. The transposed convolutions upsample the encoded representation to match the input sample rate. Details are included in Appendix C.

4 Training and Evaluation

4.1 Dataset

For self-supervised pre-training, we utilize large-scale datasets with Librilight (Kahn et al., 2020) and WenetSpeech (Zhang et al., 2022a), where we have ∼**160K** hours of 16 kHz audio that greatly increases the domain coverage.

We fine-tuning VoiceTuner-SSL to align speech and text modalities utilizing TTS data such as LibriTTS (Zen et al., 2019), VCTK (Veaux et al., 2017) and PromptSpeech (Guo et al., 2023), resulting in rich-resource VoiceTuner. To evaluate VoiceTuner in low-resource scenarios, we construct paired data (1h, 10h, 30h) with three application tasks: instruction-guided TTS, zero-shot TTS, singing voice synthesis, respectively generalizing to unseen instruction, speaker, and modality. For text sequence, we tokenize it into the phoneme sequence with an open-source grapheme-to-phoneme conversion tool (Sun et al., 2019). We have attached detailed data configuration in Appendix A.

4.2 Evaluation Metrics

Speech intelligibility. We report word error rate (WER) to evaluate the intelligibility of speech by transcribing it using a whisper (Radford et al., 2023) ASR system following (Wang et al., 2023).

Style similarity. SIM assesses the coherence of the generated speech in relation to the speaker's characteristics, and we employ the speaker verification model WavLM-TDNN (Chen et al., 2022) to evaluate the speaker similarity. F0 Frame Error (FFE) measures the prosody similarity of synthesized and reference audio. For pitch, speaking speed, and volume attributes, we adopt a soft-margin mechanism for accuracy calculation.

Subjective evaluation. We also conduct a crowd-sourced human evaluation via Amazon Mechanical Turk, which is reported with 95% confidence intervals (CI), and analyze two aspects: style similarity (speaker, emotion, and prosody) and audio quality (clarity, high-frequency), respectively scoring SMOS and MOS. More information has been attached in Appendix D.

4.3 Model Configurations

For acoustic tokens, we train the SoundStream model with 12 quantization levels, each with a codebook of size 1024 and the same downsampling rate of 320. We take three quantization levels as the acoustic tokens, representing each frame as a flat sequence of tokens from the first, second, and third quantization layers. We trained three sets of VoiceTuner, with 160M (base), 459M (medium), and 1.1B (large) parameters. As for the unit-based vocoder, we use the modified V1 version of BigV-

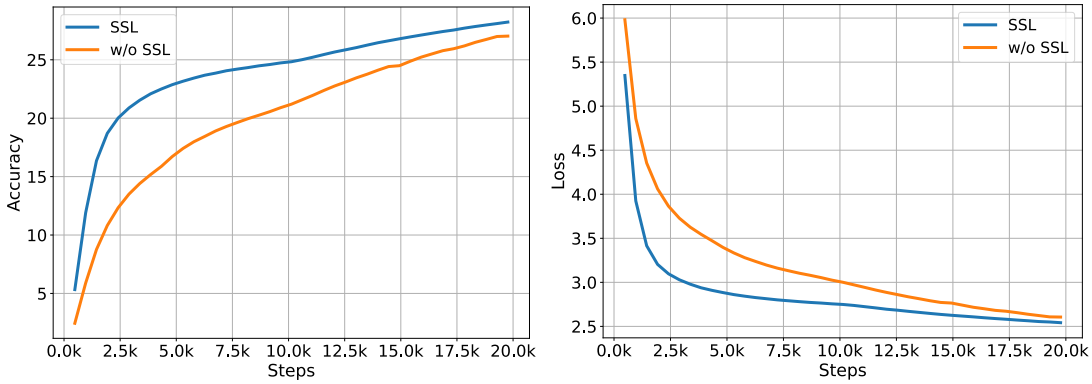


Figure 4: Loss/accuracy curves with or without self-supervised learning (SSL).

GAN. A comprehensive table of hyperparameters is available in Appendix B. Except explicitly stated, we use our 459M (medium) model for downstream evaluation.

During training, we pre-train VoiceTuner-SSL for 100K steps using 8 NVIDIA A100 GPUs with a batch size of 6000 tokens for each GPU on the publicly-available *fairseq* framework (Ott et al., 2019), and fine-tune VoiceTuner and VoiceTuner-Mamba for 10K steps using 1 NVIDIA A100 GPU. Adam optimizer is used with $\beta_1 = 0.9, \beta_2 = 0.98, \epsilon = 10^{-9}$. The unit-based vocoder is optimized with a segment size of 8192 and a learning rate of 1×10^{-4} until 500K steps using 4 NVIDIA V100 GPUs. For sampling, we employ top-p (Holtzman et al., 2019) sampling with p = 0.25.

5 Results

5.1 Self-supervised Pre-training

Model	SIM	Emotion	Style	Speed
GT	/	100	95.8	86.9
GT (voc.)	0.94	93.1	92.4	87.4
Base	0.92	90.5	78.5	63.4
Medium	0.92	91.3	81.5	65.6
Large	**0.93**	**92.7**	**83.1**	**67.1**

Table 1: Acoustic continuity of VoiceTuner-SSL.

Task	P	WER	SIM	MOS	SMOS
GT	/	3.2	/	4.35±0.05	/
GT (voc.)	/	5.6	0.93	4.23±0.07	4.20±0.05
TTS	✗	9.3	0.81	3.92±0.07	3.84±0.07
	✓	**6.7**	**0.83**	3.98±0.06	3.92±0.08
FTTS	✗	6.4	0.83	3.98±0.07	3.93±0.07
	✓	**5.9**	**0.84**	4.04±0.08	3.98±0.06

Table 2: Quality and style similarity of VoiceTuner in rich-resource TTS. FTTS: Frame-level TTS taking expanded phone as input. P: with or without pre-training.

We expect our generative foundation model VoiceTuner-SSL to keep the speaker identity, prosody, and recording conditions of the prompt and produce new content in next-token prediction. Specifically, we generate continuations of 5 seconds for each 3-second prompt, where the prompts are obtained by cropping samples from Librispeech test-clean. In the following, we run the speaker, style, emotion, and speed classifier on the sampled continuations (excluding the prompts) and report the results. We also compare the InstructSpeech with other systems, including 1) GT, the ground-truth audio; 2) GT (voc.), where we first convert the ground-truth audio into tokens and then convert them back to audio using BigVGAN;

The evaluation results are presented in Table 1, and we have the following observations: 1) VoiceTuner-SSL can preserve the speaker, style, emotion, and speaking speed in the prompt with a high recognition accuracy at a zero-shot setting, even if the model is not fine-tuned in downstream datasets; Informally, VoiceTuner-SSL is optimized in a large amount of self-supervised data, which contains many speakers with various accents and diverse demographics to improve robustness and generalization; and 2) as shown in the demo page, in a noisy environment, VoiceTuner also presents the acoustic consistency and maintain the noise conditions from the prompt.

5.2 Rich-resource Evaluation

Our proposed self-supervised pre-training and follow-up fine-tuning approach are essential for the early-stage training stability and final generation capacity. To demonstrate the rich-resource performance, we fine-tune VoiceTuner-SSL in 200-hour downstream TTS data to align speech and text modalities.

We plot the loss/accuracy curves in Figure 4 and present results in Table 2, and have the following observations: 1) the model with pre-training con-

	Instruction TTS					ZS-TTS		SVS		Subjective Evaluation	
	Gender	Speed	Pitch	Volume	WER	WER	SIM	FFE	SIM	MOS	SMOS
GT	96.6	86.9	86.9	78.9	5.1	3.2	/	/	/	4.35±0.05	/
GT (voc.)	95.8	87.4	87.0	76.0	7.1	5.6	0.93	0.01	0.95	4.23±0.07	4.20±0.05
Fine-tune with 30 hours data											
Full-s	94.1	**88.3**	**88.2**	**73.9**	16.9	18.3	0.63	/		3.94±0.06	3.89±0.08
Full-p	**94.7**	86.1	87.3	68.3	7.1	**7.2**	**0.71**	**0.31**	**0.93**	4.01±0.08	3.97±0.07
Adapter	85.1	85.1	86.7	58.8	**6.9**	7.9	0.63	0.43	0.90	3.96±0.06	3.92±0.07
Fine-tune with 10 hours data											
Full-s	90.1	76.5	85.7	61.1	68.7	/		/		/	
Full-p	**91.6**	**85.7**	85.6	**62.2**	7.6	**8.1**	**0.64**	0.47	**0.91**	3.97±0.08	3.92±0.08
Adapter	86.1	83.5	**86.3**	62.1	**7.5**	8.2	0.62	**0.44**	0.88	3.91±0.06	3.85±0.07
Fine-tune with 1 hours data											
Full-s			/			/		/		/	
Full-p	49.1	**84.5**	77.3	57.3	14.9	**8.2**	**0.66**	**0.58**	**0.83**	3.91±0.08	3.84±0.08
Adapter	**80.0**	82.9	**85.1**	**61.3**	**9.6**	8.8	0.59	0.61	0.78	3.87±0.06	3.82±0.07

Table 3: Low-resource evaluation results. Full-s: Full parameter training from scratch; Full-p: Full parameter fine-tuning from pre-trained VoiceTuner-SSL. Note that we use / to represent that the model (Full-s) cannot converges in low-resource scenarios.

verges faster and reaches lower loss bounds than the model trained from scratch; and 2) For the intelligibility of the generated speech, VoiceTuner (with pre-training) has achieved a $27\%, 7.8\%$ relatively lower WER respectively in TTS and FTTS, indicating that self-supervised pre-training provides gains with accessible speech of better quality. 3) To conclude, VoiceTuner-SSL pre-trained on an arbitrary voice corpus contains speakers with various accents, diverse demographics, and heterogeneous recording conditions, offering distinct gains in rich-resource fine-tuning.

5.3 Low-resource Evaluation

We hypothesize that a generative foundation model can be applied to different downstream tasks, reducing data requirements and computational cost, especially in low-resource scenarios. To present the capability of VoiceTuner in low-resource scenarios, we construct (1h, 10h, 30h) hours of data for three application tasks: instruction-guided TTS (ITTS), zero-shot TTS (ZS-TTS), singing voice synthesis (SVS), respectively generalizing to unseen instruction, speaker, and modality. For training efficiency, we investigate full training from scratch (Full-s), full fine-tuning from VoiceTuner-SSL (Full-p), and efficient fine-tuning with a multiscale adapter (Adapter).

The results are presented in Table 3, and we have the following observations: 1) as training data is reduced in the low-resource scenario, a distinct degradation in speech quality and similarity could be witnessed: VoiceTuner (Adapter) presents a distinct drop in TTS WER of $6.9 \rightarrow 7.5 \rightarrow 9.6$ when reducing training data from 30 to 1 hours. 2) Regarding training efficiency and computational cost: Though full parameter fine-tuning systems demonstrate better results in most cases, the multiscale adapter has still achieved the comparable results (e.g., FFE and SIM of $0.61, 0.78$ in 1-hour SVS). It indicates that the adapter enjoys high-fidelity generation with only around 1% learnable parameters, which enables us to fine-tune voice LLMs on cheap devices; 3) It is worth mentioning that in extremely low resource scenarios, VoiceTuner (Full-s) cannot converges when training from scratch. As expected, a generative model (namely VoiceTuner-SSL) without a pre-defined application can be applied to different downstream tasks, reducing data requirements in low-resource applications.

5.4 State Space Model Evaluation

Size	Params	Mem	TFLOPs	WER	SIM
VoiceTuner-Mamba: State Space Model					
B	154M	4172M	47.9	7.2	0.82
M	420M	5136M	111.4	6.4	0.83
L	1B	5141M	278.9	5.8	0.85
VoiceTuner: Transformer					
B	160M	4332M	76.3	7.8	0.81
M	459M	5259M	181.4	6.7	0.83
L	1B	5638M	408.1	5.9	0.84

Table 4: We compare VoiceTuner-Mamba and VoiceTuner among different sizes (Base, Medium, and Large). To evaluate the computational cost (the lower the better), both models predict fix-length 5s sequences and measure max-memory consumption (Mem), and total number of floating point operations (TFLOPS) during inference time.

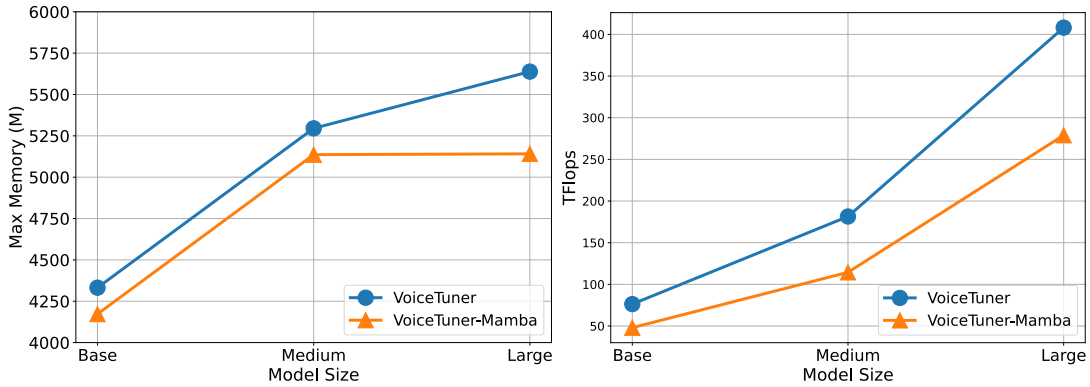


Figure 5: Memory/FLOPs curves of VoiceTuner and VoiceTuner-Mamba with different fine-tuning data.

To alleviate the challenges in modeling long audio tokens due to inefficient attention mechanisms, we propose VoiceTuner-Mamba with state space models in place of transformers. As shown in Figure 5, the efficiency curves demonstrate similar comparison results of the pure transformer backbone. As presented in Table 4, VoiceTuner-Mamba scores the lowest TFLOPs and memory of 47.9 and 4332M among base models, showing that state-space models excel at reducing the computational cost of modeling long continuous data. In contrast to the conventional attention computation approach, S6 enables each element in a 1-D array to interact with any previously scanned samples through a compressed hidden state, effectively reducing the quadratic complexity to linear.

5.5 Analysis and Ablation Studies

To verify capabilities of VoiceTuner, we conduct ablation studies on model scalability and few-shot adaptation, and discuss key findings as follows.

Tuning	Params	Gender	Speed	Pitch	WER
GT	/	96.6	86.9	86.9	5.1
Lora	8.97M	86.6	83.5	**86.3**	7.6
Adapter	12.0M	**91.6**	**85.3**	85.6	**6.9**

Table 5: Ablation studies. We obtain VoiceTuner in low-resource (10-hour) instruction TTS task and report attributes accuracy and WER.

Scalability to improve performance. As illustrated in Table 4, we report results for different model sizes, namely 160M (base), 459M (medium), and 1.1B (large) parameter models. As expected, scaling the size of VoiceTuner and VoiceTuner-Mamba results in better scores. However, this comes at the expense of longer training and inference time. Increasing the model size from 459M to 1.1B leads to additional gains of a further 40% reduction in WER for TTS tasks with a similar style.

Efficient fine-tuning with multiscale adapter. To enable few-shot learning without losing the general abilities, we fine-tune VoiceTuner in 10-hour instruction TTS data, and compare the results among different adaptation methods. Illustrated in Table 5, as a lightweight plug-and-play module, the proposed multiscale adapter enjoys superior training efficiency with only around 1% parameters in contrast to full fine-tuning, demonstrates the 9.2% WER drop and outperformed attributes accuracy (gender, speed, and pitch) compared to Lora (Hu et al., 2021). This enables us to fine-tune voice LLMs on cheap devices.

6 Conclusion

In this work, we propose VoiceTuner with a pre-training and efficient fine-tuning approach for low-resource voice generation. To mitigate the data scarcity and high computational cost for training voice LLMs, we 1) leveraged large-scale unlabeled dataset and pre-trained VoiceTuner-SSL in a next-token prediction task, which could be fine-tuned in downstream tasks with reduced data; 2) introduced an efficient multiscale adapter to fine-tune only around 1% parameters in downstream applications, further eliminating the computational cost. VoiceTuner-Mamba was proposed with a multi-scale state space model in place of transformer, alleviating the challenges of modeling long audio tokens inherited from inefficient attention mechanism. Experimental results demonstrated that VoiceTuner-SSL presented strong speech continuations. VoiceTuner exhibited superior quality and style similarity in three low-resource (1h, 10h, 30h) voice generation tasks. We envisage that our work serves as a basis for future low-resource voice synthesis studies.

7 Limitation

Although VoiceTuner is successfully applied to generate zero-shot voice signals in low-resource scenarios, it still suffers from some limitations: 1) VoiceTuner introduces a strong dependency on the quality of the audio tokenizer. 2) The model only shows in-context learning ability on voice synthesis, rather than all voice recognition and understanding tasks, and 3) a longer sequence length typically requires more computational resources, and degradation could be witnessed with decreased training data.

8 Potential Risks

VoiceTuner lowers the requirements for zero-shot voice generation even in low-resource applications, which may cause unemployment for people with related occupations, such as speech engineers and radio hosts. In addition, there is the potential for harm from non-consensual voice generation or fake media. The voices of the speakers in the recordings might be overused than they expect.

References

Andrea Agostinelli, Timo I Denk, Zalán Borsos, Jesse Engel, Mauro Verzetti, Antoine Caillon, Qingqing Huang, Aren Jansen, Adam Roberts, Marco Tagliasacchi, et al. 2023. Musiclm: Generating music from text. *arXiv preprint arXiv:2301.11325*.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*, 33:12449–12460.

Zalán Borsos, Raphaël Marinier, Damien Vincent, Eugene Kharitonov, Olivier Pietquin, Matt Sharifi, Olivier Teboul, David Grangier, Marco Tagliasacchi, and Neil Zeghidour. 2022. Audiolm: a language modeling approach to audio generation. *arXiv preprint arXiv:2209.03143*.

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.

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

Albert Gu and Tri Dao. 2023. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*.

Albert Gu, Karan Goel, and Christopher Ré. 2021a. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*.

Albert Gu, Isys Johnson, Karan Goel, Khaled Saab, Tri Dao, Atri Rudra, and Christopher Ré. 2021b. Combining recurrent, convolutional, and continuous-time models with linear state space layers. *Advances in neural information processing systems*, 34:572–585.

Zhifang Guo, Yichong Leng, Yihan Wu, Sheng Zhao, and Xu Tan. 2023. Prompttts: Controllable text-to-speech with text descriptions. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*.

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. 2021. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3451–3460.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Po-Yao Huang, Hu Xu, Juncheng Li, Alexei Baevski, Michael Auli, Wojciech Galuba, Florian Metze, and Christoph Feichtenhofer. 2022a. Masked autoencoders that listen. *Advances in Neural Information Processing Systems*, 35:28708–28720.

Rongjie Huang, Feiyang Chen, Yi Ren, Jinglin Liu, Chenye Cui, and Zhou Zhao. 2021. Multi-singer: Fast multi-singer singing voice vocoder with a large-scale corpus. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 3945–3954.

Rongjie Huang, Chenye Cui, Feiyang Chen, Yi Ren, Jinglin Liu, Zhou Zhao, Baoxing Huai, and Zhefeng Wang. 2022b. Singgan: Generative adversarial network for high-fidelity singing voice generation. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 2525–2535.

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

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.

Eugene Kharitonov, Damien Vincent, Zalán Borsos, Raphaël Marinier, Sertan Girgin, Olivier Pietquin, Matt Sharifi, Marco Tagliasacchi, and Neil Zeghidour. 2023. Speak, read and prompt: High-fidelity text-to-speech with minimal supervision. *arXiv preprint arXiv:2302.03540*.

Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. 2020. Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis. *Proc. of NeurIPS*.

Sang-gil Lee, Wei Ping, Boris Ginsburg, Bryan Catanzaro, and Sungroh Yoon. 2022. Bigvgan: A universal neural vocoder with large-scale training. *arXiv preprint arXiv:2206.04658*.

Alexander H Liu, Matt Le, Apoorv Vyas, Bowen Shi, Andros Tjandra, and Wei-Ning Hsu. 2023a. Generative pre-training for speech with flow matching. *arXiv preprint arXiv:2310.16338*.

Haohe Liu, Qiao Tian, Yi Yuan, Xubo Liu, Xinhao Mei, Qiuqiang Kong, Yuping Wang, Wenwu Wang, Yuxuan Wang, and Mark D Plumbley. 2023b. Audioldm 2: Learning holistic audio generation with self-supervised pretraining. *arXiv preprint arXiv:2308.05734*.

Jinglin Liu, Chengxi Li, Yi Ren, Feiyang Chen, and Zhou Zhao. 2022. Diffsinger: Singing voice synthesis via shallow diffusion mechanism. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Yue Liu, Yunjie Tian, Yuzhong Zhao, Hongtian Yu, Lingxi Xie, Yaowei Wang, Qixiang Ye, and Yunfan Liu. 2024. Vmamba: Visual state space model. *arXiv preprint arXiv:2401.10166*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. *arXiv preprint arXiv:1904.01038*.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *International Conference on Machine Learning*, pages 28492–28518. PMLR.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.

Yi Ren, Yangjun Ruan, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu. 2019. Fastspeech: Fast, robust and controllable text to speech. *Advances in Neural Information Processing Systems*, 32.

Jimmy TH Smith, Andrew Warrington, and Scott W Linderman. 2022. Simplified state space layers for sequence modeling. *arXiv preprint arXiv:2208.04933*.

Hao Sun, Xu Tan, Jun-Wei Gan, Hongzhi Liu, Sheng Zhao, Tao Qin, and Tie-Yan Liu. 2019. Token-level ensemble distillation for grapheme-to-phoneme conversion. *arXiv preprint arXiv:1904.03446*.

A Vasuki and PT Vanathi. 2006. A review of vector quantization techniques. *IEEE Potentials*, 25(4):39–47.

Christophe Veaux, Junichi Yamagishi, Kirsten MacDonald, et al. 2017. Cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit. *University of Edinburgh. The Centre for Speech Technology Research (CSTR)*, 6:15.

Apoorv Vyas, Bowen Shi, Matthew Le, Andros Tjandra, Yi-Chiao Wu, Baishan Guo, Jiemin Zhang, Xinyue Zhang, Robert Adkins, William Ngan, et al. 2023. Audiobox: Unified audio generation with natural language prompts. *arXiv preprint arXiv:2312.15821*.

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

Yu Wang, Xinsheng Wang, Pengcheng Zhu, Jie Wu, Hanzhao Li, Heyang Xue, Yongmao Zhang, Lei Xie, and Mengxiao Bi. 2022. Opencpop: A high-quality open source chinese popular song corpus for singing voice synthesis. *arXiv preprint arXiv:2201.07429*.

Yuxuan Wang, RJ Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, et al. 2017. Tacotron: Towards end-to-end speech synthesis. *arXiv preprint arXiv:1703.10135*.

Dongchao Yang, Jinchuan Tian, Xu Tan, Rongjie Huang, Songxiang Liu, Xuankai Chang, Jiatong Shi, Sheng Zhao, Jiang Bian, Xixin Wu, et al. 2023. Uniaudio: An audio foundation model toward universal audio generation. *arXiv preprint arXiv:2310.00704*.

Lili Yu, Dániel Simig, Colin Flaherty, Armen Aghajanyan, Luke Zettlemoyer, and Mike Lewis. 2023. Megabyte: Predicting million-byte sequences with multiscale transformers. *arXiv preprint arXiv:2305.07185*.

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*.

Binbin Zhang, Hang Lv, Pengcheng Guo, Qijie Shao, Chao Yang, Lei Xie, Xin Xu, Hui Bu, Xiaoyu Chen, Chenchen Zeng, et al. 2022a. Wenetspeech: A 10000+ hours multi-domain mandarin corpus for

speech recognition. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6182–6186. IEEE.

Lichao Zhang, Ruiqi Li, Shoutong Wang, Liqun Deng, Jinglin Liu, Yi Ren, Jinzheng He, Rongjie Huang, Jieming Zhu, Xiao Chen, et al. 2022b. M4singer: A multi-style, multi-singer and musical score provided mandarin singing corpus. *Advances in Neural Information Processing Systems*, 35:6914–6926.

Renrui Zhang, Jiaming Han, Aojun Zhou, Xiangfei Hu, Shilin Yan, Pan Lu, Hongsheng Li, Peng Gao, and Yu Qiao. 2023a. Llama-adapter: Efficient fine-tuning of language models with zero-init attention. *arXiv preprint arXiv:2303.16199*.

Ziqiang Zhang, Long Zhou, Chengyi Wang, Sanyuan Chen, Yu Wu, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, et al. 2023b. Speak foreign languages with your own voice: Cross-lingual neural codec language modeling. *arXiv preprint arXiv:2303.03926*.

Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, and Xinggang Wang. 2024. Vision mamba: Efficient visual representation learning with bidirectional state space model. *arXiv preprint arXiv:2401.09417*.

A Data

In this section, we describe details of the data usage in training and evaluating VoiceTuner.

- For self-supervised pre-training, Librilight (Kahn et al., 2020) contains 60K hours of unlabeled speech from audiobooks in English, and WenetSpeech (Zhang et al., 2022a) include 100K hours of speech in mandarin.
- For zero-shot text-to-speech, LibriTTS (Zen et al., 2019) dataset is included.
- For instruction text-to-speech, we use the dataset PromptSpeech (Guo et al., 2023).
- For singing voice synthesis, We use the female-singer OpenCPOP (Wang et al., 2022), multi-singer dataset OpenSinger (Huang et al., 2021), and M4Singer (Zhang et al., 2022b) as the singing voice data.

B Model Configurations

We list the model hyper-parameters of VoiceTuner in Table 6.

C Unit-based Vocoder

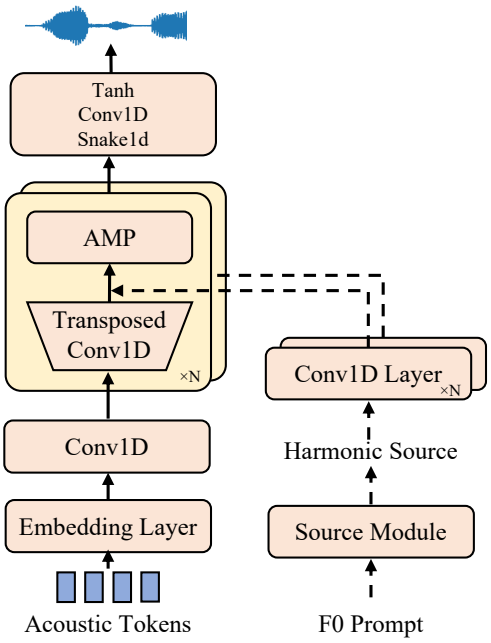


Figure 6: Overview of the unit-based vocoder. The F0 auxiliary input denoted with dotted lines is included only in singing voice synthesis.

The generator of the unit-based vocoder is built from a set of look-up tables (LUT) that embed the discrete representation, and a series of blocks composed of transposed convolution and a residual block with dilated layers. We train the enhanced vocoder with the weighted sum of the least-square adversarial loss, the feature matching loss, and the spectral regression loss on mel-spectrogram, where the training objective formulation and hyperparameters follow Kong et al. (2020); Lee et al. (2022).

For speech generation, we train the vocoder with only the discrete unit sequences as input. For singing voice generation, we further include F0-driven source excitation to stabilize long-continuous waveforms generation following (Liu et al., 2022; Huang et al., 2022b).

D Evaluation

D.1 Subjective Evaluation

For audio quality evaluation, we conduct the MOS (mean opinion score) tests and explicitly instruct the raters to "*(focus on examining the audio quality and naturalness, and ignore the differences of style (timbre, emotion, and prosody).)*". The testers present and rate the samples, and each tester is asked to evaluate the subjective naturalness on a 1-5 Likert scale.

For style similarity evaluation, we explicitly instruct the raters to "*(focus on the similarity of the style (timbre, emotion, and prosody) to the reference, and ignore the differences of content, grammar, or audio quality.)*". In the SMOS (similarity mean opinion score) tests, we paired each synthesized utterance with a ground truth utterance to evaluate how well the synthesized speech matches that of the target speaker. Each pair is rated by one rater.

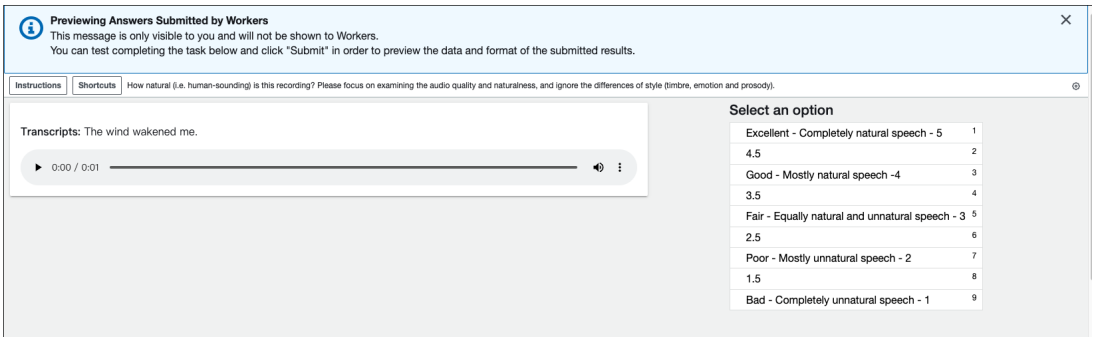
Our subjective evaluation tests are crowd-sourced and conducted by 20 native speakers via Amazon Mechanical Turk. The screenshots of instructions for testers have been shown in Figure 7. We paid $8 to participants hourly and totally spent about $600 on participant compensation. A small subset of speech samples used in the test is available at `https://VoiceTuner.github.io/`.

E Reproducibility Statement

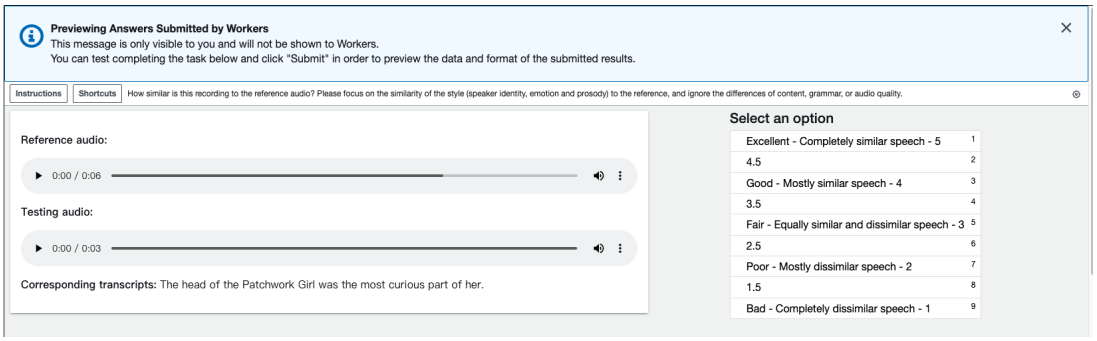
We will release our code in the future. The VoiceTuner model that we build upon is publicly available through the fairseq code repository (Ott et al., 2019). To aid reproducibility, we have included a schematic overview of hyperparameters in Table 6.

Hyperparameter		VoiceTuner/VoiceTuner-Mamba
VoiceTuner: Transformer		
Global Base	Transformer Layer Transformer Embed Dim Transformer Attention Headers Number of Parameters	16 768 12 114 M
Global Medium	Transformer Layer Transformer Embed Dim Transformer Attention Headers Number of Parameters	20 1152 16 320 M
Global Large	Transformer Layer Transformer Embed Dim Transformer Attention Headers Number of Parameters	24 1536 32 830 M
Local	Transformer Layer Transformer Embed Dim Transformer Attention Headers Number of Parameters	6 Same as global 8 46/101/303 M
VoiceTuner-Mamba: State Space Model		
Global Base	State space Layer State space Embed Dim Number of Parameters	24 768 91 M
Global Medium	State space Layer State space Embed Dim Number of Parameters	32 1152 281 M
Global Large	State space Layer State space Embed Dim Number of Parameters	48 1536 792 M
Local	State space Layer State space Embed Dim Number of Parameters	12/12/16 Same as global 63/139/245 M
BigVGAN Vocoder		
BigVGAN Vocoder	Upsample Rates Hop Size Upsample Kernel Sizes Number of Parameters	[5, 4, 2, 2, 2, 2] 320 [9, 8, 4, 4, 4, 4] 121.6M

Table 6: Hyperparameters of VoiceTuner.



(a) Screenshot of MOS testing.



(b) Screenshot of SMOS testing.

Figure 7: Screenshots of subjective evaluations.