Neural Codec Language Models are Zero-Shot Text to Speech Synthesizers

Chengyi Wang* **Sanyuan Chen*** **Yu Wu*** **Ziqiang Zhang** **Long Zhou** **Shujie Liu**
Zhuo Chen **Yanqing Liu** **Huaming Wang** **Jinyu Li** **Lei He** **Sheng Zhao** **Furu Wei**
Microsoft
https://github.com/microsoft/unilm

Abstract

We introduce a language modeling approach for text to speech synthesis (TTS). Specifically, we train a *neural codec language model* (called VALL-E) using discrete codes derived from an off-the-shelf neural audio codec model, and regard TTS as a conditional language modeling task rather than continuous signal regression as in previous work. During the pre-training stage, we scale up the TTS training data to 60K hours of English speech which is hundreds of times larger than existing systems. VALL-E emerges *in-context learning* capabilities and can be used to synthesize high-quality personalized speech with only a 3-second enrolled recording of an unseen speaker as an acoustic prompt. Experiment results show that VALL-E significantly outperforms the state-of-the-art zero-shot TTS system in terms of speech naturalness and speaker similarity. In addition, we find VALL-E could preserve the speaker's emotion and acoustic environment of the acoustic prompt in synthesis. See https://aka.ms/valle for demos of our work.

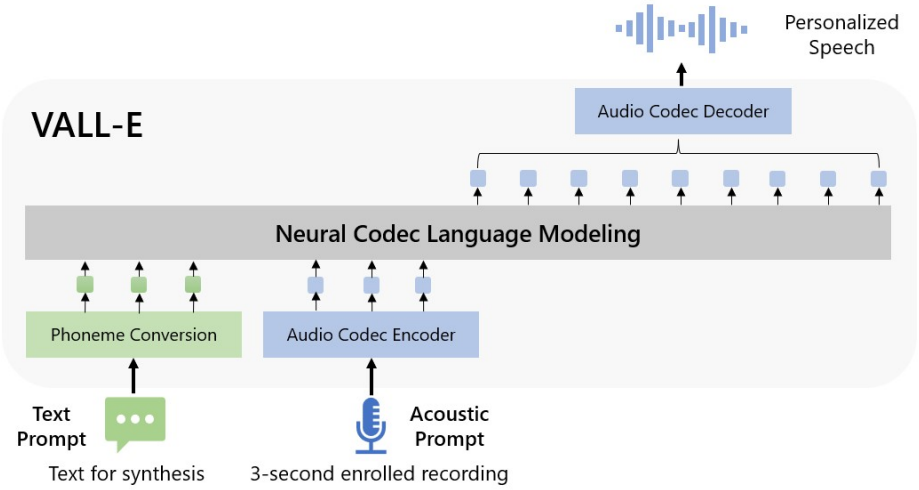


Figure 1: The overview of VALL-E. Unlike the previous pipeline (e.g., phoneme → mel-spectrogram → waveform), the pipeline of VALL-E is phoneme → discrete code → waveform. VALL-E generates the discrete audio codec codes based on phoneme and acoustic code prompts, corresponding to the target content and the speaker's voice. VALL-E directly enables various speech synthesis applications, such as zero-shot TTS, speech editing, and content creation combined with other generative AI models like GPT-3 [Brown et al., 2020].

*These authors contributed equally to this work. Correspondence: {yuwu1,shujliu,fuwei}@microsoft.com

1 Introduction

The last decade has yielded dramatic breakthroughs in speech synthesis through the development of neural networks and end-to-end modeling. Currently, cascaded text to speech (TTS) systems [Shen et al., 2018, Ren et al., 2019, Li et al., 2019] usually leverage a pipeline with an acoustic model and a vocoder using mel spectrograms as the intermediate representations. While advanced TTS systems can synthesize high-quality speech from single or multiple speakers [Liu et al., 2022, Kim et al., 2021], it still requires high-quality clean data from the recording studio. Large-scale data crawled from the Internet cannot meet the requirement, and always lead to performance degradation. Because the training data is relatively small, current TTS systems still suffer from poor generalization. Speaker similarity and speech naturalness decline dramatically for unseen speakers in the zero-shot scenario. To tackle the zero-shot TTS problem, existing work leverages speaker adaptation [Chen et al., 2019, Wang et al., 2020] and speaker encoding [Arik et al., 2018, Casanova et al., 2022b] methods, requiring additional fine-tuning, complex pre-designed features, or heavy structure engineering.

Instead of designing a complex and specific network for this problem, the ultimate solution is to train a model with large and diverse data as much as possible, motivated by success in the field of text synthesis [Brown et al., 2020, Chowdhery et al., 2022]. Recent years have witnessed notable performance improvement for data increase in the text language model, from 16GB of uncompressed text [Devlin et al., 2019], to 160GB [Liu et al., 2019], to 570GB [Brown et al., 2020], and finally, around 1TB [Chowdhery et al., 2022]. Transferring this success to the field of speech synthesis, we introduce VALL-E, the first language model based TTS framework leveraging the large, diverse, and multi-speaker speech data. As shown in Figure 1, to synthesize personalized speech (e.g., zero-shot TTS), VALL-E generates the corresponding acoustic tokens conditioned on the acoustic tokens of the 3-second enrolled recording and the phoneme prompt, which constrain the speaker and content information respectively. Finally, the generated acoustic tokens are used to synthesize the final waveform with the corresponding neural codec decoder [Défossez et al., 2022]. The discrete acoustic tokens derived from an audio codec model enable us to treat TTS as conditional codec language modeling, and advanced prompting-based large-model techniques (as in GPTs [Brown et al., 2020]) can be leveraged for the TTS tasks. The acoustic tokens also allow us to generate diverse synthesized results in TTS by using different sampling strategies during inference.

We train VALL-E with LibriLight [Kahn et al., 2020], a corpus consisting of 60K hours of English speech with over 7000 unique speakers. The original data is audio-only, so we employ a speech recognition model to generate the transcriptions. Compared to previous TTS training datasets, such as LibriTTS [Zen et al., 2019], our data contain more noisy speech and inaccurate transcriptions but provide diverse speakers and prosodies. We believe the proposed approach is robust to the noise and generalize well by leveraging large data. It is worth noting that existing TTS systems are always trained with dozens of hours of single-speaker data or hundreds of hours of multi-speaker data, which is over hundreds of times smaller than VALL-E. Table 1 summarizes the innovation of VALL-E, a language model approach for TTS, using audio codec codes as intermediate representations, leveraging large and diverse data, leading to strong in-context learning capabilities.

Table 1: A comparison between VALL-E and current cascaded TTS systems.

	Current Systems	**VALL-E**
Intermediate representation	mel spectrogram	audio codec code
Objective function	continuous signal regression	language model
Training data	$\leq$ 600 hours	60K hours
In-context learning	✗	✓

We evaluate VALL-E on LibriSpeech [Panayotov et al., 2015] and VCTK [Veaux et al., 2016] datasets, where all test speakers are unseen in the training corpus. VALL-E significantly outperforms the state-of-the-art zero-shot TTS system [Casanova et al., 2022b] in terms of speech naturalness and speaker similarity, with +0.12 comparative mean option score (CMOS) and +0.93 similarity mean option score (SMOS) improvement on LibriSpeech. VALL-E also beats the baseline on VCTK with +0.11 SMOS and +0.23 CMOS improvements. It even achieves a +0.04 CMOS score against ground truth, showing the synthesized speech of unseen speakers is as natural as human recordings on VCTK. Moreover, the qualitative analysis shows that VALL-E is able to synthesize diverse outputs with the

same text and target speaker, which could benefit pseudo-data creation for the speech recognition task. We also find that VALL-E could keep the acoustic environment (e.g., reverberation) and emotion (e.g. anger) of the acoustic prompt.

In summary, we make the following contributions.

- We propose VALL-E, the first TTS framework with strong in-context learning capabilities as GPT-3, which treats TTS as a language model task with audio codec codes as an intermediate representation to replace the traditional mel spectrogram. It has in-context learning capability and enables prompt-based approaches for zero-shot TTS, which does not require additional structure engineering, pre-designed acoustic features, and fine-tuning as in previous work.
- We build a generalized TTS system in the speaker dimension by leveraging a huge amount of semi-supervised data, suggesting that simple scaling up semi-supervised data has been underestimated for TTS.
- VALL-E is able to provide diverse outputs with the same input text and keep the acoustic environment and speaker's emotion of the acoustic prompt.
- We verify that VALL-E synthesizes natural speech with high speaker similarity by prompting in the zero-shot scenario. Evaluation results show that VALL-E significantly outperforms the state-of-the-art zero-shot TTS system on LibriSpeech and VCTK.

We encourage the reader to listen to our samples on the demo page `https://aka.ms/valle`.

2 Related Work

Zero-Shot TTS: Current TTS methods can be categorized into cascaded and end-to-end methods. Cascaded TTS systems [Shen et al., 2018, Ren et al., 2019, Li et al., 2019] usually leverage a pipeline with an acoustic model and a vocoder using mel spectrograms as the intermediate representations. To tackle the drawbacks of the vocoder, end-to-end TTS models [Kim et al., 2021, Liu et al., 2022] are proposed to jointly optimize the acoustic model and vocoder. In real scenarios, it is highly desirable to customize a TTS system to an arbitrary voice with rare enrolled recordings. Therefore, there is growing interest in the zero-shot multi-speaker TTS techniques, and most of work is done in the context of cascaded TTS systems. As the pioneers, Arik et al. [2018] proposes speaker adaptation and speaker encoding approaches. In the line of speaker adaptation, the following work [Chen et al., 2019, Wang et al., 2020, Chen et al., 2021] tries to improve the adaptation efficiency with less target speaker data and speaker-specific parameters. Huang et al. [2022] applies meta-learning on speaker adaptation, which only requires 5-shot to build a well-performed system. In parallel, speaker encoding-based methods achieved great progress in recent years. A speaker encoding based system contains a speaker encoder and a TTS component, where the speaker encoder could be pre-trained on the speaker verification task [Jia et al., 2018]. In Jia et al. [2018] and Arik et al. [2018], the experiments show that the model is able to generate high-quality outputs with 3 seconds enrolled recordings for in-domain speakers. To improve the quality of unseen speakers, advanced speaker embedding models [Cai et al., 2018] can be employed, but it is still undesirable according to Tan et al. [2021]. Another way is to design advanced but complex speaker encoder [Wu et al., 2022]. Diffusion model based TTS [Popov et al., 2021, Kim et al., 2022] is also extended to zero-shot TTS [Kang et al., 2022] and achieved good results. Compared to previous work [Ren et al., 2019, Du et al., 2022], our work follows the line of cascaded TTS but first uses audio codec code as intermediate representations. It is the first one that has strong in-context learning capabilities as GPT-3, which does not require fine-tuning, pre-designed features, or a complex speaker encoder.

Spoken generative pre-trained models: Self-supervised learning is widely investigated in the field of speech understanding [Baevski et al., 2020b, Hsu et al., 2021, Chen et al., 2022] and speech-to-speech generation [Lakhotia et al., 2021, Borsos et al., 2022]. In the context of speech-to-speech generation, a hot topic is how to synthesize speech in a textless setting. GSLM [Lakhotia et al., 2021] proposes to synthesize speech based on HuBERT codes [Hsu et al., 2021], and Polyak et al. [2021] improves the performance by combining HuBERT codes with codes of VQVAE and a speaker encoder. AudioLM [Borsos et al., 2022] follows a similar way but use audio codecs [Zeghidour et al., 2022] to synthesize speech, together with semantic codes. It should be noted that AudioLM is able to synthesize speech based on audio codecs without training an additional vocoder such as HifiGAN [Kong et al., 2020]. AudioLM is a speech-to-speech model, whereas VALL-E is a TTS model, so

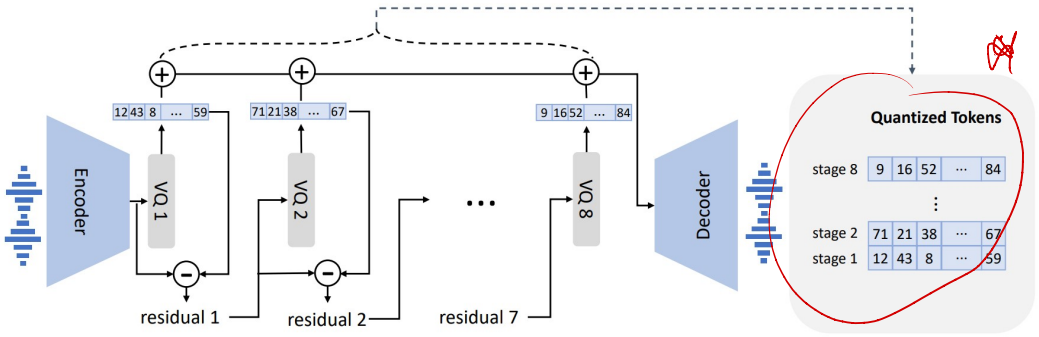


Figure 2: The neural audio codec model revisit. Because RVQ is employed, the first quantizer plays the most important role in reconstruction, and the impact from others gradually decreases.

we can explicitly control the content in speech synthesis. Another direction is to apply pre-training to the neural TTS. Chung et al. [2018] pre-trains speech decoder in TTS through autoregressive mel-spectrogram prediction. In Ao et al. [2022], the authors propose a unified-modal encoder-decoder framework SpeechT5, which can leverage unlabeled speech and text data to pre-train all components of TTS model. Tjandra et al. [2019] quantizes unlabeled speech into discrete tokens by a VQVAE model [van den Oord et al., 2017], and train a model with the token-to-speech sequence. They demonstrate that the pre-trained model only requires a small amount of real data for fine-tuning. Bai et al. [2022] proposes mask and reconstruction on mel spectrogram and showing better performance on speech editing and synthesis. Previous TTS pre-training work leverages less than 1K hours of data, whereas VALL-E is pre-trained with 60K hours of data. Furthermore, VALL-E is the first to use audio codec codes as intermediate representations, and emerge in-context learning capability in zero-shot TTS.

3 Background: Speech Quantization

Since audio is typically stored as a sequence of 16-bit integer values, a generative model is required to output $2^{16} = 65,536$ probabilities per timestep to synthesize the raw audio. In addition, the audio sample rate exceeding ten thousand leads to an extraordinarily long sequence length, making it more intractable for raw audio synthesis. To this end, speech quantization is required to compress integer values and sequence length. μ-law transformation can quantize each timestep to 256 values and reconstruct high-quality raw audio. It is widely used in speech generative models, such as WaveNet [van den Oord et al., 2016], but the inference speed is still slow since the sequence length is not reduced. Recently, vector quantization is widely applied in self-supervised speech models for feature extraction, such as vq-wav2vec [Baevski et al., 2020a] and HuBERT [Hsu et al., 2021]. The following work [Lakhotia et al., 2021, Du et al., 2022] shows the codes from self-supervised models can also reconstruct content, and the inference speed is faster than WaveNet. However, the speaker identity has been discarded and the reconstruction quality is low [Borsos et al., 2022]. AudioLM [Borsos et al., 2022] trains speech-to-speech language models on both k-means tokens from a self-supervised model and acoustic tokens from a neural codec model, leading to high-quality speech-to-speech generation.

In this paper, we follow AudioLM [Borsos et al., 2022] to leverage neural codec models to represent speech in discrete tokens. To compress audio for network transmission, codec models are able to encode waveform into discrete acoustic codes and reconstruct high-quality waveform even if the speaker is unseen in training. Compared to traditional audio codec approaches, the neural-based codec is significantly better at low bitrates, and we believe the quantized tokens contain sufficient information about the speaker and recording conditions. Compared to other quantization methods, the audio codec shows the following advantages: 1) It contains abundant speaker information and acoustic information, which could maintain speaker identity in reconstruction compared to HuBERT codes [Hsu et al., 2021]. 2) There is an off-the-shelf codec decoder to convert discrete tokens into a waveform, without the additional efforts on vocoder training like VQ-based methods that operated on spectrum [Du et al., 2022]. 3) It could reduce the length of time steps for efficiency to address the problem in μ-law transformation [van den Oord et al., 2016].

We adopt a pre-trained neural audio codec model, EnCodec [Défossez et al., 2022], as our tokenizer. EnCodec is a convolutional encoder-decoder model, whose input and output are both 24 kHz audio across variable bitrates. The encoder produces embeddings at 75 Hz for input waveforms at 24 kHz, which is a 320-fold reduction in the sampling rate. Each embedding is modeled by a residual vector quantization (RVQ), in which we choose eight hierarchy quantizers with 1024 entries each as shown in Figure 2. This configuration corresponds to EnCodec at 6K bitrates for 24 kHz audio reconstruction. In this setting, given a 10-second waveform, the discrete representation is a matrix with 750×8 entries, where $750 = \frac{24{,}000 \times 10}{320}$ is the downsampled time step and 8 is the number of quantizers. It is fine to choose other bitrate settings. A larger bitrate corresponds to more quantizers and better reconstruction quality. For example, if we choose EnCodecc at 12K bitrates, there are 16 quantizers are needed and the 10-second waveform corresponds to a matrix with 750×16 entries. With the discrete codes from all quantizers, the convolutional decoder of EnCodec generates real-valued embeddings and reconstructs the waveform at 24 kHz.

4 VALL-E

token vs modality

4.1 Problem Formulation: Regarding TTS as Conditional Codec Language Modeling

Given a dataset $\mathcal{D} = \{\mathbf{x}_i, \mathbf{y}_i\}$, where $\mathbf{y}$ is an audio sample and $\mathbf{x} = \{x_0, x_1, \ldots, x_L\}$ is its corresponding phoneme transcription, we use a pre-trained neural codec model to encode each audio sample into discrete acoustic codes, denoted as $\text{Encodec}(\mathbf{y}) = \mathbf{C}^{T \times 8}$, where $\mathbf{C}$ represents the two-dimensional acoustic code matrix, and T is the downsampled utterance length. The row vector of each acoustic code matrix $\mathbf{c}_{t,:}$ represents the eight codes for frame t and the column vector of each acoustic code matrix $\mathbf{c}_{:,j}$ represents the code sequence from the j-th codebook, where $j \in \{1, \ldots, 8\}$. After quantization, the neural codec decoder is able to reconstruct the waveform, denoted as $\text{Decodec}(\mathbf{C}) \approx \hat{\mathbf{y}}$.

Zero-shot TTS requires the model to synthesize high-quality speech for unseen speakers. In this work, we regard zero-shot TTS as a conditional codec language modeling task. We train a neural language model to generate an acoustic code matrix $\mathbf{C}$ conditioned on a phoneme sequence $\mathbf{x}$ and an acoustic prompt matrix $\tilde{\mathbf{C}}^{T' \times 8}$ with the optimization objective of $\max p(\mathbf{C}|\mathbf{x}, \tilde{\mathbf{C}})$. Here, $\tilde{\mathbf{C}}$ is obtained by the same neural codec with an enrolled recording as the input. We expect the neural language model learns to extract the content and speaker information from the phoneme sequence and the acoustic prompt, respectively. During inference, given a phoneme sequence and a 3-second enrolled recording of the unseen speaker, the acoustic code matrix with corresponding content and speaker's voice is firstly estimated by the trained language model. Then the neural codec decoder synthesizes the high-quality speech.

4.2 Training: Conditional Codec Language Modeling

The neural speech codec model allows us to operate on discrete audio representations. Due to residual quantization in the neural codec model, the tokens have a hierarchical structure: tokens from previous quantizers recover acoustic properties like speaker identity, while the consecutive quantizers learn fine acoustic details. Each quantizer is trained to model the residual from the previous quantizers. Motivated by this, we design two conditional language models in a hierarchical manner.

For the discrete tokens from the first quantizer $\mathbf{c}_{:,\mathbf{1}}$, we train an autoregressive (AR) decoder-only language model. It is conditioned on the phoneme sequence $\mathbf{x}$ and the acoustic prompt $\tilde{\mathbf{C}}_{:,1}$, formulated as

$$p(\mathbf{c}_{:,1}|\mathbf{x}, \tilde{\mathbf{C}}_{:,1}; \theta_{AR}) = \prod_{t=0}^{T} p(\mathbf{c}_{t,1}|\mathbf{c}_{<t,1}, \tilde{\mathbf{c}}_{:,1}, \mathbf{x}; \theta_{AR}) \tag{1}$$

Since VALL-E is a decoder-only LM, the concatenation of $\tilde{\mathbf{c}}_{:,\mathbf{1}}$ and $\mathbf{c}_{:,1}$ is a whole sequence, and we do not distinguish them or insert a specific token in training. Only $\mathbf{c}_{:,1}$ is predicted while the prefix $\tilde{\mathbf{c}}_{:,1}$ is given during inference.

For the discrete tokens from the second to the last quantizers, $\mathbf{c}_{:,j\in[2,8]}$, we train a non-autoregressive (NAR) language model. Since the tokens can not access each other in a NAR manner, to constrain the speaker identity, the acoustic prompt matrix $\tilde{C}$ is used as an acoustic prompt. Thus, the model is

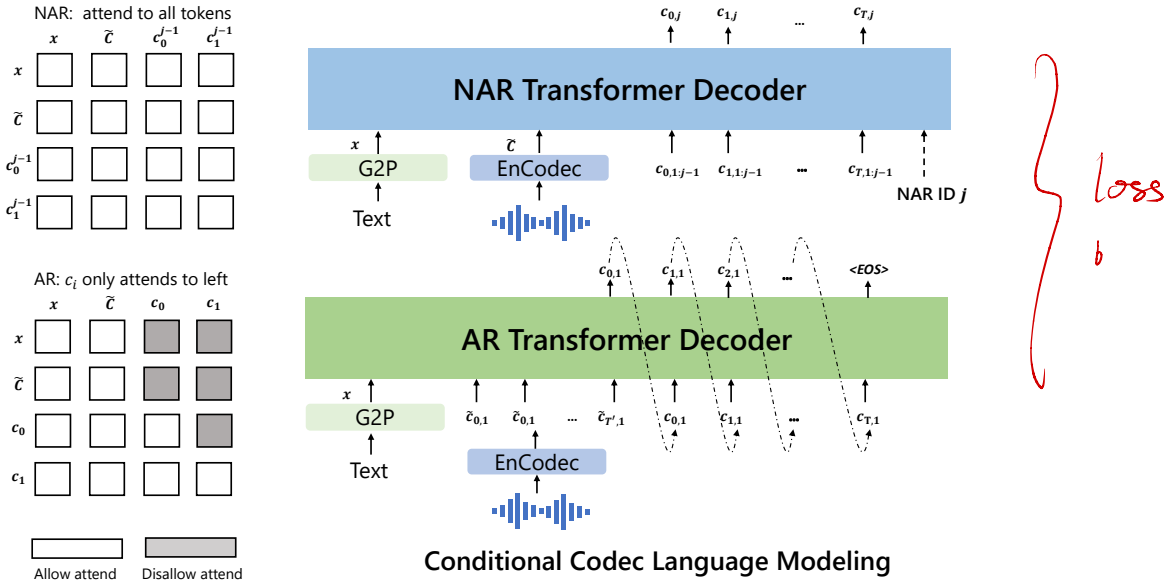


Figure 3: The structure of the conditional codec language modeling, which is built in a hierarchical manner. In practice, the NAR decoder will be called seven times to generate codes in seven quantizers.

conditioned on the phoneme sequence $\mathbf{x}$, the acoustic prompt $\tilde{\mathbf{C}}$ and the predicted acoustic tokens belong to the previous codebooks $\mathbf{C}_{:,<j}$:

$$p(\mathbf{C}_{:,2:8}|\mathbf{x}, \tilde{\mathbf{C}}; \theta_{NAR}) = \prod_{j=2}^{8} p(\mathbf{c}_{:,j}|\mathbf{C}_{:,<j}, \mathbf{x}, \tilde{\mathbf{C}}; \theta_{NAR}) \tag{2}$$

The combination of the AR model and the NAR model provides a good trade-off between speech quality and inference speed. On the one hand, the rate of the generated speech should be consistent with the enrolled recording, and it is hard to train a length predictor for different speakers since their speaking speed may be very diverse. In this case, the AR model is a more natural choice with its flexibility for acoustic sequence length prediction. On the other hand, for the consecutive stages, as the number of output slots follows the sequence length of the first stage, NAR can reduce the time complexity from $\mathcal{O}(T)$ to $\mathcal{O}(1)$. Overall, the prediction of $\mathbf{C}$ can be modeled as:

$$p(\mathbf{C}|\mathbf{x}, \tilde{\mathbf{C}}; \theta) = p(\mathbf{c}_{:,1}|\tilde{\mathbf{C}}_{:,1}, \mathbf{X}; \theta_{AR}) \prod_{j=2}^{8} p(\mathbf{c}_{:,\mathbf{j}}|\mathbf{c}_{:,<j}, \mathbf{x}, \tilde{\mathbf{C}}; \theta_{NAR}) \tag{3}$$

4.2.1 Autoregressive Codec Language Modeling

The autoregressive language model generates the tokens from the first quantizer. It comprises a phoneme embedding W_x, an acoustic embedding W_a, a transformer decoder, and a prediction layer. In order to generate speech with specific content, we use the phoneme sequence as the phoneme prompt of the language model. Thus, the model input is the concatenation of $\mathbf{x}$ and $\mathbf{c}_{:,1}$, and two special <EOS> tokens are appended after each of them. We compute sinuous position embedding separately for prompt and input tokens. For the causal transformer model, each token $c_{t,1}$ can attend to $(\mathbf{x}, \mathbf{c}_{<t,1})$ as illustrated in the left part of Figure 3. The model is optimized to maximize the probability of the next token in the first codebook. We share the parameters of the output projection layer with the parameters of the acoustic embedding W_a.

In the AR model, we do not explicitly extract an audio clip as the prompt in training. The training process is pure casual language model training. In this way, any prefix sequence $\mathbf{c}_{<t,1}$ is treated as a prompt for the latter part of the sequence $\mathbf{c}_{\geq t,1}$. During inference, given an enrolled recording, we should concatenate the phoneme sequence of the enrolled recording and the phoneme sequence for synthesis together. Meanwhile, the acoustic token sequence of the enrolled recording is used as the

prefix in AR decoding, as formulated in equation 1. We will study the superiority of this setting in the experiment.

4.2.2 Non-Autoregressive Codec Language Modeling

When we obtain the first quantizer codes by the AR model, we employ a non-autoregressive (NAR) model to generate codes of the other seven quantizers. The NAR model has a similar architecture to the AR model, except that it contains eight separate acoustic embedding layers. In each training step, we randomly sample a training stage $i \in [2, 8]$. The model is trained to maximize the acoustic tokens from the i-th quantizer codebook. The acoustic tokens from stage 1 to stage $i-1$ are embedded and summed up as model input:

$$e_{c_{t,j}} = W_a^j \odot c_{t,j} \tag{4}$$

$$\mathbf{e_{c_t}} = \sum_{j=1}^{i-1} e_{c_t,j} \tag{5}$$

where $\odot$ indicates index selection. The phoneme sequence is also regarded as the prompt of the language model. Besides, to clone the unique voice of the given speaker, we also use the acoustic tokens from the enrolled speech as the acoustic prompt. Specifically, we first tokenize the enrolled speech with the neural codec model as $\tilde{\mathbf{C}}^{T\times 8}$. The embedded representations from all of the eight codebooks are summed up as the acoustic prompt $\mathbf{e_{\tilde{c}_t}} = \sum_{j=1}^{8} e_{\tilde{c}_{t,j}}$. To predict the acoustic tokens from the i-th codebook, the transformer input is the concatenation of $(\mathbf{e_x}, \mathbf{e_{\tilde{c}}}, \mathbf{e_{c_{:,<i}}})$. The positional embeddings are also computed separately for prompts and the acoustic sequence. The current stage i is injected into the network with Adaptive Layer Normalization [Xu et al., 2019] operator, i.e., $\text{AdaLN}(h, i) = a_i \text{LayerNorm}(h) + b_i$, where h is the intermediate activations, a_i and b_i are obtained from a linear projection of the stage embedding. Unlike AR, the NAR model allows each token to attend to all the input tokens in the self-attention layer. We also share the parameters of the acoustic embedding layer and the output prediction layer, which means the weights of the j-th prediction layer are the same as the $(j+1)$-th acoustic embedding layer.

4.3 Inference: In-Context Learning via Prompting

In-context learning is a surprising ability of the text-based language model, which is able to predict labels for unseen inputs without additional parameter updates. For TTS, if the model can synthesize high-quality speech for unseen speakers without fine-tuning, the model is believed to have in-context learning capability. However, the in-context learning capability of existing TTS systems is not strong, because they either require additional fine-tuning or degrade dramatically for unseen speakers.

For language models, prompting is necessary to enable in-context learning in the zero-shot scenario. We design prompts and inference as follows. We first convert the text into a phoneme sequence and encode the enrolled recording into an acoustic matrix, forming the phoneme prompt and acoustic prompt. Both prompts are used in the AR and NAR models. For the AR model, we use sampling-based decoding conditioned on the prompts since we observe that beam search may lead the LM into an infinity loop. Furthermore, the sampling-based method could significantly increase the diversity of the output. For the NAR model, we use greedy decoding to choose the token with the highest probability. Finally, we use the neural codec decoder to generate the waveform conditioned on the eight code sequences. The acoustic prompt may or may not semantically relate to the speech to be synthesized, resulting in two cases:

VALL-E: Our main interest is to generate given content for unseen speakers. The model is given a text sentence, a segment of enrolled speech, and its corresponding transcription. We prepend the transcription phoneme of the enrolled speech to the phoneme sequence of the given sentence as the phoneme prompt, and use the first layer acoustic token of the enrolled speech $\tilde{c}_{:,1}$ as an acoustic prefix. With the phoneme prompt and the acoustic prefix, VALL-E generates the acoustic tokens for the given text cloning the voice of this speaker.

VALL-E-continual: In this setting, we use the whole transcription and the first 3 seconds of the utterance as the phoneme and acoustic prompts respectively, and ask the model to generate the continuations. The inference process is the same as setting VALL-E, except that the enrolled speech and the generated speech are semantically continuous.

5 Experiment

5.1 Experiment Setup

Dataset: We use LibriLight [Kahn et al., 2020] as the training data which contains 60K hours of unlabelled speech from audiobooks in English. The number of distinct speakers is around 7000 in LibriLight. We train a hybrid DNN-HMM ASR model on 960 hours labeled LibriSpeech following Kaldi recipe [Povey et al., 2011]. Once the hybrid model is trained, unlabeled speech data is decoded and transduced to the best phoneme-level alignment paths where the frameshift is 30ms. The EnCodec model [Défossez et al., 2022] is used to generate the acoustic code matrix for the 60K hours of data.

Model: Both the AR model and the NAR model have the same transformer architecture with 12 layers, 16 attention heads, an embedding dimension of 1024, a feed-forward layer dimension of 4096, and a dropout of 0.1. The average length of the waveform in LibriLight is 60 seconds. During training, we randomly crop the waveform to a random length between 10 seconds and 20 seconds. Its corresponding phoneme alignments are used as the phoneme prompt. We remove the consecutive repetitions in the force-aligned phoneme sequence. For the NAR acoustic prompt tokens, we select a random segment waveform of 3 seconds from the same utterance.

The models are trained using 16 NVIDIA TESLA V100 32GB GPUs with a batch size of 6k acoustic tokens per GPU for 800k steps. We optimize the models with the AdamW optimizer, warm up the learning rate for the first 32k updates to a peak of 5×10^{-4}, and then linear decay it.

Baseline: We choose the SOTA zero-shot TTS model YourTTS [Casanova et al., 2022b] as the baseline, which is trained on a combined dataset of VCTK [Veaux et al., 2016], LibriTTS [Zen et al., 2019], and TTS-Portuguese [Casanova et al., 2022a]. We use their released checkpoint*.

Automatic metrics: We employ the SOTA speaker verification model, WavLM-TDNN [Chen et al., 2022], to evaluate the speaker similarity between prompt (the decompressed enrolled speech) and synthesized speech. WavLM-TDNN achieved the top rank at the VoxSRC Challenge 2021 and 2022 leaderboards. It reached an average Equal Error Rate (EER) of 0.383, 0.480, and 0.986 on Vox1-O, Vox1-E, and Vox1-H respectively. The similarity score predicted by WavLM-TDNN is in the range of $[-1, 1]$, where a larger value indicates a higher similarity of input samples.

We also evaluate the synthesis robustness of our model. Neural TTS systems suffer from the robustness issue, which sometimes has deletion, insertion, and replacement errors due to wrong attention alignments. We perform ASR on the generated audio and calculate the word error rate (WER) with respect to the original transcriptions. In this experiment, we employ the HuBERT-Large [Hsu et al., 2021] model fine-tuned on LibriSpeech 960h as the ASR model, which is a CTC-based model without language model fusion.

Human evaluation: We calculate the comparative mean option score (CMOS) and similarity mean option score (SMOS) by crowdsourcing, where 12 and 6 native speakers are invited as CMOS and SMOS contributors. The scale of SMOS is from 1 to 5 with 0.5-point increments. CMOS ranges from -3 (the new system is much worse than baseline) to 3 (the new system is much better than baseline) with intervals of 1. CMOS is an indicator of speech naturalness, and SMOS measures whether the speech is similar to the original speaker's voice.

5.2 LibriSpeech Evaluation

We first use LibriSpeech [Panayotov et al., 2015] for zero-shot TTS evaluation, since there is no speaker overlap between LibriLight training data and LibriSpeech test-clean data. Following Borsos et al. [2022], we use the samples from LibriSpeech test-clean with lengths between 4 and 10 seconds, resulting in a 2.2 hours subset. For each sample synthesis, VALL-E randomly choose another utterance of the same speaker and crop a 3-seconds speech segment as the enrolled speech. Each experiment runs three times and the average score is reported. VALL-E-continual uses the first 3 seconds of the ground-truth speech as enrolled speech.

Table 2 shows the objective evaluation results. We first compute the WER score and the speaker similarity score of the ground truth speech as the upper bound. To compare the speaker similarity, we use speech pairs from the same speaker in the test set. Compared with the YourTTS baseline, our

*https://github.com/Edresson/YourTTS

Table 2: Evaluation results on audio generation. YourTTS and VALL-E are text-to-speech models using phonemes as inputs, while GSLM and AudioLM are speech-to-speech models using latent code as inputs. The WER result of AudioLM is obtained by a Conformer Transducer model [Borsos et al., 2022]. Since AudioLM* is not open-source, we cannot evaluate its speaker score with our tool.

model	WER	SPK
GroundTruth	2.2	0.754
Speech-to-Speech Systems		
GSLM	12.4	0.126
AudioLM*	6.0	-
TTS Systems		
YourTTS	7.7	0.337
VALL-E	5.9	**0.580**
VALL-E-continual	**3.8**	0.508

model is significantly better in both robustness and speaker similarity, showing that our generated speech is highly faithful to the given text and the given enrolled speech. Furthermore, the word error rate can be further reduced in VALL-E-continual setting, because the acoustic tokens for the first 3 seconds are extracted from the ground truth. We also compare the robustness with other speech-to-speech LM-based generation models, GSLM and AudioLM, which use audio latent codes as input. GSLM uses HuBERT code as input and reconstructs the waveform with the Tacotron2 [Shen et al., 2018] model and the WaveGlow [Prenger et al., 2019] vocoder. We run their open-sourced code using the released model and evaluate the results. Since the HuBERT codes discard the speaker identity, it achieves a poor speaker score. For the AudioLM, we list their WER score reported in their paper, which is obtained by a Conformer Transducer model. The experiment results show that VALL-E is better than other speech-to-speech LM-based generative systems in terms of robustness. One major reason is VALL-E trained with pseudo-phoneme instead of HuBERT/w2v-BERT codes, which enjoys better alignment quality with the input text.

We randomly sample one utterance for each speaker in LibriSpeech test-clean for the human evaluation, resulting in 40 test cases. Table 3 shows the human evaluation results. VALL-E is very closed to ground truth in terms of SMOS, indicating the synthesized speech is similar to the given unseen speaker in testing. It significantly outperforms the baseline with +0.93 SMOS, demonstrating the effectiveness of VALL-E in zero-shot scenarios. Regarding naturalness, VALL-E beats the baseline with +0.12 CMOS, indicating the proposed method could synthesize more natural and realistic speech against baselines.

Table 3: Human evaluation with 40 speakers on LibriSpeech test-clean with 3-second enrolled recording for each.

	SMOS	CMOS (v.s. VALL-E)
YourTTS	$3.45_{\pm 0.09}$	-0.12
VALL-E	$4.38_{\pm 0.10}$	0.00
GroundTruth	$4.5_{\pm 0.10}$	+0.17

Ablation study: In this section, we perform detailed ablation experiments. We first study the NAR model. We train three NAR models with different numbers of prompts. The setting **NAR-no prompt** is trained without any prompts. The setting **NAR-phn prompt** is trained with only phoneme sequence as prompt and the setting **NAR-2 prompts** uses both phoneme prompt and acoustic token prompt as conditions. In evaluation, we use the ground-truth first-level acoustic tokens as the model input and compute the WER and speaker similarity scores. The results are listed in Table 4. Results show that the model without any prompts performs poorly on both ASR and speaker similarity evaluations, even though the acoustic input token is ground truth. When adding the phoneme prompt, the WER is reduced by a large margin from 19.6 to 3.0. It shows the phoneme prompt mainly contributes to the content of the generation. In the **NAR-2 prompts**, the model can learn speaker information from the acoustic token prompt and thus improve the speaker evaluation quality.

Table 4: Ablation study of the NAR model. The inputs of the NAR models are the ground-truth for the ablation study.

	NAR-no prompt	**NAR-phn prompt**	**NAR-2 prompts**
WER	19.6	3.0	**2.8**
SPK	0.518	0.541	**0.732**

We further conduct the ablation experiments on the AR model. In these experiments, we always use the **NAR-2 prompts** setting as the NAR model. In Table 5, we can see that when we remove the acoustic prompt (w/o acoustic prompt), it can only obtain a speaker similarity score of 0.236, showing the prompt is extremely crucial for speaker identity. Even if the NAR model could see the prompt, the prompt for the AR model also contributes a lot to speaker similarity.

Table 5: Ablation study of the AR model.

	WER	SPK
VALL-E	**5.9**	**0.585**
w/o acoustic prompt	**5.9**	0.236

5.3 VCTK Evaluation

We evaluate our model on VCTK consisting of 108 speakers, where none of the speakers are observed during training. Since YourTTS has seen 97 speakers in VCTK as training, we evaluate YourTTS performance on the full 107 speakers and 11 unseen speakers, respectively. For each speaker, we randomly selected three utterances of 3s/5s/10s as the prompts and the text of another utterance as the text prompt.

Table 6: Automatic evaluation of speaker similarity with 108 speakers on VCTK. *YourTTS has observed 97 speakers during training, while VALL-E observed none of them.

	3s prompt	5s prompt	10s prompt
108 full speakers			
YourTTS*	0.357	0.377	0.394
VALL-E	0.382	0.423	**0.484**
GroundTruth	0.546	0.591	0.620
11 unseen speakers			
YourTTS	0.331	0.337	0.344
VALL-E	0.389	0.380	**0.414**
GroundTruth	0.528	0.556	0.586

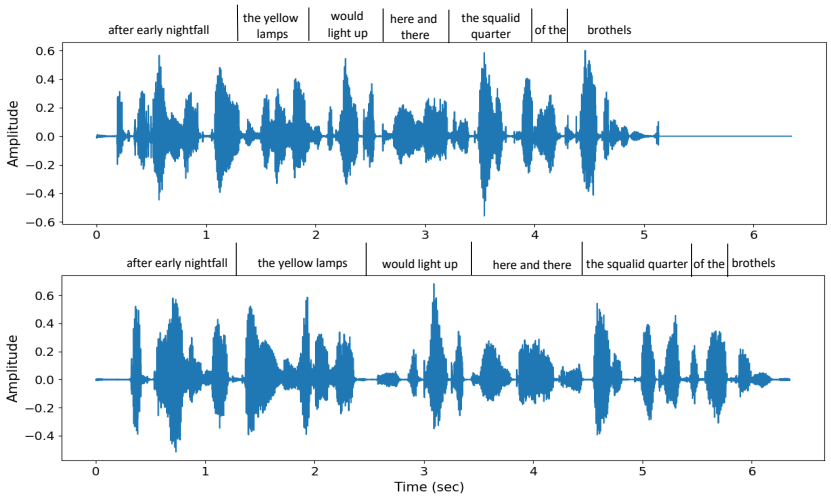
We first evaluate two models with the speaker verification metric as described before. From Table 6, we can see that VALL-E outperforms the baseline even if the baseline has seen 97 speakers in training, indicating our model is able to synthesize speech with higher speaker similarity. When we compare with the baseline in a fair setting (11 speakers), the performance gap becomes larger, especially when only 3s prompts are available. By comparing different lengths of the prompt, we can see our model is able to generate more similar speech when the prompt becomes longer, which is consistent with our intuition.

We sample 60 speakers for human evaluation, one utterance for each, where 11 are unseen speakers, and 49 speakers have been seen for YourTTS. VALL-E do not see any of the 60 speakers. During model synthesis, each speaker has a 3-second enrolled recording. Table 7 shows a comparison of our method against baseline and ground truth. The comparison of SMOS shows that VALL-E has better speaker similarity than the baseline, even if the baseline has seen some of the speakers in training. The side-by-side CMOS evaluation shows that VALL-E is +0.23 over YourTTS, indicating a significantly better performance on speaking of naturalness. Furthermore, VALL-E achieves +0.04 CMOS over ground-truth, demonstrating no statistically significant difference from human recordings on this dataset. Compared to the evaluation results on LibriSpeech, VALL-E shows a better CMOS

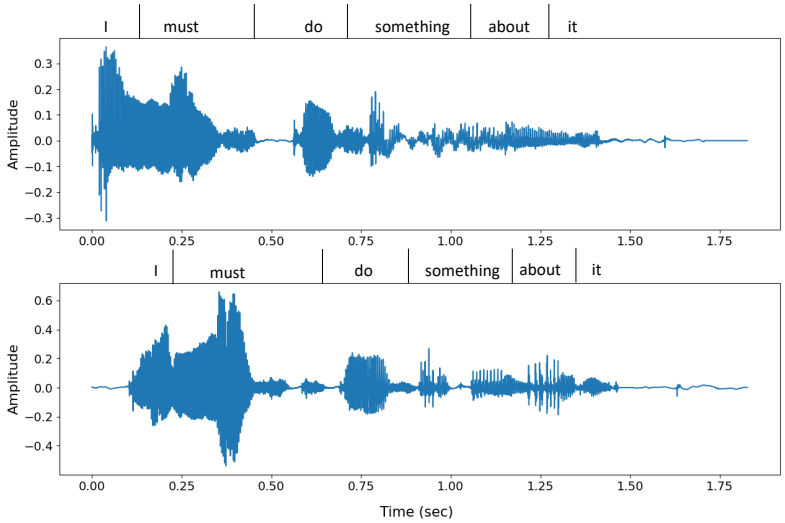
Table 7: Human evaluation with 60 speakers on VCTK with 3-second enrolled recording for each.

	SMOS	CMOS (v.s. VALL-E)
YourTTS*	$3.70_{\pm 0.09}$	-0.23
VALL-E	$3.81_{\pm 0.09}$	0.00
GroundTruth	$4.29_{\pm 0.09}$	-0.04

score in the comparison with ground truth, which is mainly because the average sentence length is shorter and some of the ground truth utterances also have noisy environments in VCTK. In terms of speaker similarity, VCTK is more challenging as it contains speakers with various accents while the training data and LibriSpeech test data do not contain various accent speakers.



(a) A LibriSpeech sample: After early nightfall, the yellow lamp would light up here and there the squalid quarter of the brothels.



(b) A VCTK sample: I must do something about it.

Figure 4: Diversity analysis of VALL-E. Each utterance is synthesized two times with different random seeds. We can observe substantial diversity of the two outputs regarding the same input.

5.4 Qualitative Analysis

Diversity: Previous TTS systems have a strong one-one mapping between input text and output waveform, because mel spectrum generation is based on reconstruction for each step without randomness. Since VALL-E uses the sampling-based method to generate discrete tokens, its output is diverse for the same input text due to the randomness in inference. Given a sentence and an enrolled recording, we run the inference process twice and visualize its waveform in Figure 4. In Figure 4(a), we observe the two samples have different lengths and phrase durations, where the first has a faster speech rate. In Figure 4(b), we observe that the accents of the two samples are different. The second output emphasizes the word “must" with a larger amplitude whereas the first output does not. We leave more samples on our demo page.

The diversity is important for some downstream scenarios. For example, speech recognition always benefits from diverse inputs with different speakers and acoustic environments, which cannot be met by the previous TTS system. Considering the diversity feature of VALL-E, it is an ideal candidate to generate pseudo-data for speech recognition.

Acoustic environment maintenance: Another interesting finding is the acoustic environment consistency between the acoustic prompt and the generation. When the acoustic prompt has reverberation, VALL-E could synthesize speech with reverberation as well, whereas the baseline outputs clean speech. Our explanation is that VALL-E is trained on a large-scale dataset consisting of more acoustic conditions than the data used by the baseline, so VALL-E could learn the acoustic consistency instead of a clean environment only during training. We show consistency on our demo page.

Speaker’s emotion maintenance: Emotional TTS is a classic subtopic of speech synthesis, which synthesizes speech with a required emotion. Traditional methods [Lei et al., 2021] always train a model on a supervised emotional TTS dataset, where the speech corresponds to a transcription and an emotion label. We find that VALL-E can preserve the emotion in the prompt at a zero-shot setting. We select acoustic prompts from EmoV-DB [Adigwe et al., 2018], a dataset containing speech with five emotions, VALL-E is able to keep the same emotion of the prompt in speech synthesis, even if the model is not fine-tuned on an emotional TTS dataset. We put audio samples on our demo page.

6 Conclusion, Limitations, and Future Work

We introduced VALL-E, a language model approach for TTS with audio codec codes as intermediate representations. We pre-train VALL-E with 60K hours of speech data, and show the in-context learning capability in zero-shot scenarios. We achieve new state-of-the-art zero-shot TTS results on LibriSpeech and VCTK. Furthermore, VALL-E could keep the acoustic environment and speaker’s emotion in synthesis, and provide diverse outputs in different sampling-based decoding processes.

Despite making significant progress, VALL-E still suffers from several issues.

Synthesis robustness: We observe that some words may be unclear, missed, or duplicated in speech synthesis. It is mainly because the phoneme-to-acoustic language part is an autoregressive model, in which disordered attention alignments exist and no constraints to solving the issue. The phenomenon is also observed in vanilla Transformer-based TTS, which was addressed by applying non-autoregressive models or modifying the attention mechanism in modeling. In the future, we would like to leverage these techniques to solve the issue.

Data coverage: Even if we use 60K hours of data for training, it still cannot cover everyone’s voice, especially accent speakers. The worse result on VCTK than LibriSpeech also implies insufficient coverage of accent speakers. Moreover, the diversity of speaking styles is not enough, as LibriLight is an audiobook dataset, in which most utterances are in reading style. In the future, we will further scale up the training data to improve the model performance across prosody, speaking style, and speaker similarity perspectives. We believe the zero-shot TTS task could be almost solved through our approach with model and data scale-up.

Model Structure: Now, we use two models to predict codes of different quantizers. A promising direction is to predict them with a large universal model. Another interesting direction is using full NAR models to speed up model inference in the framework.

Broader impacts: Since VALL-E could synthesize speech that maintains speaker identity, it may carry potential risks in misuse of the model, such as spoofing voice identification or impersonating

a specific speaker. To mitigate such risks, it is possible to build a detection model to discriminate whether an audio clip was synthesized by VALL-E. We will also put Microsoft AI Principles[*] into practice when further developing the models.

References

Adaeze Adigwe, Noé Tits, Kevin El Haddad, Sarah Ostadabbas, and Thierry Dutoit. The emotional voices database: Towards controlling the emotion dimension in voice generation systems. *arXiv preprint arXiv:1806.09514*, 2018.

Junyi Ao, Rui Wang, Long Zhou, Chengyi Wang, Shuo Ren, Yu Wu, Shujie Liu, Tom Ko, Qing Li, Yu Zhang, et al. Speecht5: Unified-modal encoder-decoder pre-training for spoken language processing. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5723–5738, 2022.

Sercan Ömer Arik, Jitong Chen, Kainan Peng, Wei Ping, and Yanqi Zhou. Neural voice cloning with a few samples. In *NeurIPS*, pages 10040–10050, 2018.

Alexei Baevski, Steffen Schneider, and Michael Auli. vq-wav2vec: Self-supervised learning of discrete speech representations. In *ICLM*, 2020a.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. *NeurIPS*, 33:12449–12460, 2020b.

He Bai, Renjie Zheng, Junkun Chen, Mingbo Ma, Xintong Li, and Liang Huang. A^3t: Alignment-aware acoustic and text pretraining for speech synthesis and editing. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 1399–1411. PMLR, 2022.

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

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *NeurIPS*, 2020.

Weicheng Cai, Jinkun Chen, and Ming Li. Exploring the encoding layer and loss function in end-to-end speaker and language recognition system. In *Odyssey 2018: The Speaker and Language Recognition Workshop, 26-29 June 2018, Les Sables d'Olonne, France*, pages 74–81. ISCA, 2018.

Edresson Casanova, Arnaldo Cândido Júnior, Christopher Shulby, Frederico Santos de Oliveira, João Paulo Ramos Teixeira, Moacir Antonelli Ponti, and Sandra M. Aluísio. Tts-portuguese corpus: a corpus for speech synthesis in brazilian portuguese. *Lang. Resour. Evaluation*, 56(3):1043–1055, 2022a.

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

Mingjian Chen, Xu Tan, Bohan Li, Yanqing Liu, Tao Qin, Sheng Zhao, and Tie-Yan Liu. Adaspeech: Adaptive text to speech for custom voice. In *ICLR*, 2021.

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

[*] https://www.microsoft.com/ai/responsible-ai

Yutian Chen, Yannis M. Assael, Brendan Shillingford, David Budden, Scott E. Reed, Heiga Zen, Quan Wang, Luis C. Cobo, Andrew Trask, Ben Laurie, Çaglar Gülçehre, Aäron van den Oord, Oriol Vinyals, and Nando de Freitas. Sample efficient adaptive text-to-speech. In *ICLR ,*, 2019.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. Palm: Scaling language modeling with pathways. *CoRR*, abs/2204.02311, 2022.

Yu-An Chung, Yuxuan Wang, Wei-Ning Hsu, Yu Zhang, and R. J. Skerry-Ryan. Semi-supervised training for improving data efficiency in end-to-end speech synthesis. In *ICASSP*, pages 6940–6944. IEEE, 2018.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, pages 4171–4186, 2019.

Chenpeng Du, Yiwei Guo, Xie Chen, and Kai Yu. VQTTS: high-fidelity text-to-speech synthesis with self-supervised VQ acoustic feature. In *Interspeech 2022, 23rd Annual Conference of the International Speech Communication Association, Incheon, Korea, 18-22 September 2022*, pages 1596–1600. ISCA, 2022. doi: 10.21437/Interspeech.2022-489.

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

Sung-Feng Huang, Chyi-Jiunn Lin, Da-Rong Liu, Yi-Chen Chen, and Hung-yi Lee. Meta-tts: Meta-learning for few-shot speaker adaptive text-to-speech. *IEEE ACM Trans. Audio Speech Lang. Process.*, 30:1558–1571, 2022.

Ye Jia, Yu Zhang, Ron J. Weiss, Quan Wang, Jonathan Shen, Fei Ren, Zhifeng Chen, Patrick Nguyen, Ruoming Pang, Ignacio Lopez-Moreno, and Yonghui Wu. Transfer learning from speaker verification to multispeaker text-to-speech synthesis. In *NeurIPS*, pages 4485–4495, 2018.

Jacob Kahn, Morgane Rivière, Weiyi Zheng, Evgeny Kharitonov, Qiantong Xu, Pierre-Emmanuel Mazaré, Julien Karadayi, Vitaliy Liptchinsky, Ronan Collobert, Christian Fuegen, et al. Libri-light: A benchmark for asr with limited or no supervision. In *ICASSP*, pages 7669–7673. IEEE, 2020.

Minki Kang, Dongchan Min, and Sung Ju Hwang. Any-speaker adaptive text-to-speech synthesis with diffusion models. *CoRR*, abs/2211.09383, 2022. doi: 10.48550/arXiv.2211.09383.

Heeseung Kim, Sungwon Kim, and Sungroh Yoon. Guided-tts: A diffusion model for text-to-speech via classifier guidance. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato, editors, *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 11119–11133. PMLR, 2022.

Jaehyeon Kim, Jungil Kong, and Juhee Son. Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech. In *ICML*, volume 139 of *Proceedings of Machine Learning Research*, pages 5530–5540. PMLR, 2021.

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

Kushal Lakhotia, Evgeny Kharitonov, Wei-Ning Hsu, Yossi Adi, Adam Polyak, Benjamin Bolte, Tu Anh Nguyen, Jade Copet, Alexei Baevski, Adelrahman Mohamed, and Emmanuel Dupoux. Generative spoken language modeling from raw audio. *CoRR*, abs/2102.01192, 2021.

Yi Lei, Shan Yang, and Lei Xie. Fine-grained emotion strength transfer, control and prediction for emotional speech synthesis. In *2021 IEEE Spoken Language Technology Workshop (SLT)*, pages 423–430. IEEE, 2021.

Naihan Li, Shujie Liu, Yanqing Liu, Sheng Zhao, and Ming Liu. Neural speech synthesis with transformer network. In *AAAI*, pages 6706–6713. AAAI, 2019.

Yanqing Liu, Ruiqing Xue, Lei He, Xu Tan, and Sheng Zhao. Delightfultts 2: End-to-end speech synthesis with adversarial vector-quantized auto-encoders. In *Interspeech 2022, 23rd Annual Conference of the International Speech Communication Association, Incheon, Korea, 18-22 September 2022*, pages 1581–1585. ISCA, 2022. doi: 10.21437/Interspeech.2022-277.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: an asr corpus based on public domain audio books. In *ICASSP*, pages 5206–5210. IEEE, 2015.

Adam Polyak, Yossi Adi, Jade Copet, Eugene Kharitonov, Kushal Lakhotia, Wei-Ning Hsu, Abdelrahman Mohamed, and Emmanuel Dupoux. Speech resynthesis from discrete disentangled self-supervised representations. In *Interspeech*, pages 3615–3619. ISCA, 2021.

Vadim Popov, Ivan Vovk, Vladimir Gogoryan, Tasnima Sadekova, and Mikhail A. Kudinov. Grad-tts: A diffusion probabilistic model for text-to-speech. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 8599–8608. PMLR, 2021. URL http://proceedings.mlr.press/v139/popov21a.html.

Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al. The kaldi speech recognition toolkit. In *IEEE 2011 workshop on automatic speech recognition and understanding*, number CONF. IEEE Signal Processing Society, 2011.

Ryan Prenger, Rafael Valle, and Bryan Catanzaro. Waveglow: A flow-based generative network for speech synthesis. In *ICASSP*, pages 3617–3621. IEEE, 2019.

Yi Ren, Yangjun Ruan, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu. Fastspeech: Fast, robust and controllable text to speech. In *NeurIPS*, pages 3165–3174, 2019.

Jonathan Shen, Ruoming Pang, Ron J. Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, RJ-Skerrv Ryan, Rif A. Saurous, Yannis Agiomyrgiannakis, and Yonghui Wu. Natural TTS synthesis by conditioning wavenet on MEL spectrogram predictions. In *ICASSP*, pages 4779–4783. IEEE, 2018.

Xu Tan, Tao Qin, Frank K. Soong, and Tie-Yan Liu. A survey on neural speech synthesis. *CoRR*, abs/2106.15561, 2021.

Andros Tjandra, Berrak Sisman, Mingyang Zhang, Sakriani Sakti, Haizhou Li, and Satoshi Nakamura. VQVAE unsupervised unit discovery and multi-scale code2spec inverter for zerospeech challenge 2019. In *Interspeech*, pages 1118–1122. ISCA, 2019.

Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew W. Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. In *The 9th ISCA Speech Synthesis Workshop*, page 125. ISCA, 2016.

Aäron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 6306–6315, 2017.

Christophe Veaux, Junichi Yamagishi, Kirsten MacDonald, et al. Superseded-cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit. 2016.

Tao Wang, Jianhua Tao, Ruibo Fu, Jiangyan Yi, Zhengqi Wen, and Rongxiu Zhong. Spoken content and voice factorization for few-shot speaker adaptation. In *Interspeech*, pages 796–800. ISCA, 2020.

Yihan Wu, Xu Tan, Bohan Li, Lei He, Sheng Zhao, Ruihua Song, Tao Qin, and Tie-Yan Liu. Adaspeech 4: Adaptive text to speech in zero-shot scenarios. In *Interspeech 2022, 23rd Annual Conference of the International Speech Communication Association, Incheon, Korea, 18-22 September 2022*, pages 2568–2572. ISCA, 2022. doi: 10.21437/Interspeech.2022-901.

Jingjing Xu, Xu Sun, Zhiyuan Zhang, Guangxiang Zhao, and Junyang Lin. Understanding and improving layer normalization. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 4383–4393, 2019.

Neil Zeghidour, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi. Soundstream: An end-to-end neural audio codec. *IEEE ACM Trans. Audio Speech Lang. Process.*, 30: 495–507, 2022.

Heiga Zen, Viet Dang, Rob Clark, Yu Zhang, Ron J. Weiss, Ye Jia, Zhifeng Chen, and Yonghui Wu. Libritts: A corpus derived from librispeech for text-to-speech. In *Interspeech*, pages 1526–1530. ISCA, 2019.