



On the Effectiveness of Acoustic BPE in Decoder-Only TTS

Bohan Li¹, Feiyu Shen¹, Yiwei Guo¹, Shuai Wang², Xie Chen¹, Kai Yu^{1,*}

¹MoE Key Lab of Artificial Intelligence, AI Institute, X-LANCE Lab, Department of Computer Science and Engineering, Shanghai Jiao Tong University, China

²Shenzhen Research Institute of Big Data, CUHK-Shenzhen, China

{everlastingnight, francis_sfy, yiwei.guo, chenxie95, kai.yu}@sjtu.edu.cn¹
{wangshuai}@cuhk.edu.cn²

Abstract

Discretizing speech into tokens and generating them by a decoder-only model have been a promising direction for text-to-speech (TTS) and spoken language modeling (SLM). To shorten the sequence length of speech tokens, acoustic byte-pair encoding (BPE) has emerged in SLM that treats speech tokens from self-supervised semantic representations as characters to further compress the token sequence. But the gain in TTS has not been fully investigated, and the proper choice of acoustic BPE remains unclear. In this work, we conduct a comprehensive study on various settings of acoustic BPE to explore its effectiveness in decoder-only TTS models with semantic speech tokens. Experiments on LibriTTS verify that acoustic BPE uniformly increases the intelligibility and diversity of synthesized speech, while showing different features across BPE settings. Hence, acoustic BPE is a favorable tool for decoder-only TTS.

Index Terms: discrete speech token, acoustic byte-pair encoding, decoder-only text-to-speech

1. Introduction

The language modeling paradigm has been revolutionizing text-to-speech (TTS) by its strong generation ability since the birth of large decoder-only language models (LMs). There have been LM-based TTS models that exhibit versatile, expressive and even emergent abilities [1, 2, 3, 4, 5, 6].

Unlike text, speech is intrinsically a continuous signal with a much lower information density. To adapt speech into LMs and generate it in an autoregressive manner, researchers have adopted different methods to map speech into discrete tokens [2, 4]. By the purpose of discretization, discrete speech tokens can be divided into acoustic tokens [7, 8, 9] that aim to reconstruct the signal perfectly, and semantic tokens [10, 11, 12, 13], from self-supervised models that provide a compact abstraction of speech semantics. After discretization, decoder-only LMs like VALL-E [3] can be seamlessly applied to TTS, where the discrete speech tokens are treated as the targets given a text input.

However, the low rate of information behind speech still leads to excessively lengthy discrete speech sequences compared to text transcriptions [14]. For example, more than 500 HuBERT [11] tokens may be required for a single sentence of around 30 words just to convey an idea or two. It is even worse for the acoustic tokens, since most of neural speech codec models require a short frame shift and the residual vector quantization [8] technique to reconstruct the signal decently. Regardless of the type of tokens, this feature of speech poses a great challenge for long-context modeling of LM-based TTS systems.

To address this issue, one possible way is to further compress the discrete speech sequence. A promising approach is the acoustic byte-pair encoding (BPE) technique, which is proposed in [15]. It is a similar method to the traditional BPE algorithm [16] in natural language processing. It treats the discrete indexes of speech as literal characters and iteratively compresses consecutive tokens based on the frequency in the training corpus. Such compression will coherently reduce sequence length with the increase of vocabulary size. For speech discrete tokens, usually a group of multiple tokens occur together to represent a specific phoneme or syllable, and organizing them to be a unique modeling unit would provide a higher abstraction of semantics and morphological information. Therefore, it is intuitively reasonable to apply acoustic BPE on discrete speech tokens in both reducing sequence length and improving representability. In previous researches, such acoustic BPE has been adopted to encode pseudo-target labels in HuBERT pretraining [15] and automatic speech recognition [17].

Nevertheless, the effectiveness of applying acoustic BPE in the TTS task still remains unclear to the literature. Although BASE-TTS [6] and VoxLM [18] mentions the use of acoustic BPE in generation, the design space of acoustic BPE is still not fully investigated, and the gain of such technique in TTS needs to be further explored. Despite of the possible improvements, acoustic BPE could bring more difficulties in choosing the correct unit for generation, since the vocabulary of the LM in TTS is greatly enlarged. Too much abstraction of acoustic units could also make language modeling harder. Moreover, acoustic BPE might have different behaviors and performances on different types of discrete speech tokens.

Therefore, in this work, we conduct a comprehensive study on the effectiveness of introducing acoustic BPE to TTS in the decoder-only LM paradigm. We implement various settings of acoustic BPE on the semantic tokens extracted by speech self-supervised models, and then train a VALL-E autoregressive decoder-only transformer as the acoustic model to generate acoustic BPEs from text inputs. Afterwards, the original semantic tokens are unfolded and fed to a unit-based vocoder [19, 20] for waveform synthesis. We consider HuBERT [11] and WavLM [13] as the source of semantic tokens, adjust the number of clusters in extracting the semantic tokens from 2048 to 8192, and increase the vocabulary size of acoustic BPE up to 20k in order to observe the effects made by different acoustic BPE settings. We perform most of the experiments on LibriTTS [21] and evaluate the effectiveness of such settings via speech intelligibility, sample diversity and speech quality in objective and subjective measurements.

Our findings suggest that models employing acoustic BPE method are capable of generating audio with high intelligibility and quality, while also achieving much faster training and in-

^{0*}: The corresponding author.

ference processes. This indicates that acoustic BPE can be of great value for the TTS task when properly configured. This could lay a solid foundation for future decoder-only LM-based TTS models, as the pros of acoustic BPE weighs much more than its con.

2. Decoder-Only TTS with Acoustic BPE

2.1. Acoustic BPE

BPE is a widely-used algorithm for data compression and text encoding. In natural language processing, it is initialized with a vocabulary that contains all the characters with their emergence frequency in the text corpus, and iteratively merges the most frequent character pairs until a specified number of merges or a target vocabulary size is reached [22]. For most languages like English, there exists blanks as the obvious boundaries between words in the text. However, the textual sequences of audio discrete tokens lack obvious boundaries, akin to linguistic features found in Chinese. Following [14], we first obtain speech discrete tokens from k-means clusters of features derived from speech self-supervised learning (SSL) models (HuBERT, WavLM). Then we establish a bijective mapping for conversion between the speech discrete tokens and the Chinese-character Unicodes, simply using an integer offset. Chinese characters mapping to speech discrete tokens are used to train the BPE model. With these operations, the acoustic BPE encoding is composed of conversion to Chinese characters and the encoding of pretrained BPE model. The acoustic BPE decoding is exactly the inverse process. This research utilizes the acoustic BPE method as a part of the tokenizer, which plays a role in preprocessing inputs for the decoder-only language model mentioned in the next subsection(2.2). Similar to natural language processing, we can consider the upcoming modeling process as a spoken language modeling process.

Based on the method description above, it is evident that the method involves multiple dimensions of configuration, such as the SSL model, k-means cluster number, BPE vocabulary size, and so on. Configuring combinations across different dimensions may lead to entirely new characteristics, which can have significant implications in spoken language modeling. A comprehensive and systematic exploration of these configurations' impact on modeling is crucial for better constructing spoken language models.

2.2. Decoder-Only TTS Language Modeling

The language model can estimate the probability of the input sequence, which is usually expressed as:

$$p(\mathbf{x}|\theta) = \prod_{i=1}^t p(x_i|x_{<i}, \theta) \quad (1)$$

where $\mathbf{x} = \{x_1, \dots, x_t\}$ is the input sequence with length t and the language model is parameterized by θ .

However, the spoken language model in TTS pays the other attention on text conditions. Based on the decoder-only Transformer architecture, the training and inference process of the TTS model follows the autoregressive (AR) stage in VALL-E.

2.2.1. Decoder-only TTS model

During training process, we put the phoneme sequence \mathbf{x} and tokenized audio sequence \mathbf{s} into the decoder-only language model (with parameters denoted as θ), estimating the probability au-

to regressively, formulated as:

$$p(\mathbf{s}|\mathbf{x}; \theta) = \prod_{i=1}^t p(s_i|\mathbf{s}_{:i-1}, \mathbf{x}; \theta) \quad (2)$$

The model optimizes its parameters by performing the task of predicting the next acoustic feature based on the phonemized text and historical speech features. This training process can be viewed as maximizing the sequence probability of the tokenized audio sequence under the condition of the text, which can be described as:

$$\hat{\theta} = \arg \max_{\theta} p(\mathbf{s}|\mathbf{x}; \theta) = \arg \max_{\theta} \prod_{i=1}^t p(s_i|\mathbf{s}_{:i-1}, \mathbf{x}; \theta) \quad (3)$$

where $\hat{\theta}$ is our optimization target.

2.2.2. Inference with prompts

During inference, the model generates speech with similar speakers and prosody as the given audio prompt. Concatenating tokenized audio prompt $\mathbf{s}_{\text{prompt}}$, phonemized source text \mathbf{x} , and the phonemized text prompt $\mathbf{x}_{\text{prompt}}$ corresponding to the audio prompt, we build the input of the spoken language model in format of $\{\mathbf{x}_{\text{prompt}}, \mathbf{x}, \mathbf{s}_{\text{prompt}}\}$. The model then autoregressively generates the remaining sequence after this sequence. The decoding process consists of acoustic BPE decoding (if used) and the vocoding process of a discrete-unit-based vocoder [20] for speech waveform synthesis. Additionally, the Mel-spectrogram extracted from the speech prompt is also input into the vocoder to achieve better speaker control.

3. Experiments and Results

In this section, we will introduce the exploration involving the configuration of various dimensions in the acoustic BPE method and analyze the results regarding Decoder-only TTS performance with these configurations.

3.1. Experimental Setup

3.1.1. Datasets

The experiments were conducted on the LibriSpeech [23] and LibriTTS [21] datasets. LibriTTS, designed for text-to-speech tasks, consists of approximately 585 hours of English speech from multiple speakers. Its train-960 subset was used for model training. The LibriSpeech dataset served primarily as the test set for speech synthesis. In this experiment, sentences shorter than 4 seconds or longer than 10 seconds were filtered out from the test-clean subset of LibriSpeech. The remaining 1145 sentences, totaling approximately 2.02 hours of speech, were used as test cases. All speech audio data was downsampled to 16kHz before use.

3.1.2. Settings of acoustic BPE

The encoding process of acoustic BPE consists of two stages: speech discretization and acoustic BPE model training. In the speech discretization stage, we utilized the HuBERT-large¹ model pretrained with masked prediction on the 60k hours of LibriLight [24] dataset and the WavLM-large² model pretrained on extra dataset consisting of 10k hours of Gigaspeech [25] and 24k hours English data subset of VoxPopuli [26], as the self-supervised speech representation model. Continuous audio features were extracted from the final layer's output activations of

¹<https://github.com/facebookresearch/fairseq/tree/main/examples/hubert>

²<https://huggingface.co/microsoft/wavlm-large>

their Transformer encoder, to train k-means models with 256, 2048, 4096 and 8192 centroids, which encoded the semantic features extracted from the LibriTTS datasets into speech discrete token sequences. Due to memory constraints, we randomly sampled 100 hours of speech from the LibriTTS train-960 set during k-means model training.

In the next stage, we trained the acoustic BPE model on the speech discrete token sequences extracted from the LibriTTS train-960 set. All discrete token sequences were first converted into Unicode strings, and then a BPE model was training on these strings with the SentencePiece³ toolkit. The pretrained BPE model encoded the speech discrete token sequences from LibriTTS into acoustic BPE sequences. Four different acoustic BPE encodings were experimented with in this study: no acoustic BPE encoding, and acoustic BPE encoding with vocabulary sizes of 5,000, 10,000, and 20,000 subwords, respectively.

3.1.3. Decoder-only TTS architecture and training

The TTS model is based on the AR model of VALL-E [3], which is a decoder-only Transformer architecture consisting of 12 Transformer layers, each with 16 attention heads and a hidden feature dimension of 1024. The absolute positional encoding based on trigonometric functions is employed separately to the text and speech feature sequences. As there are no official implementations, we resorted to an unofficial repository⁴.

During training, the model was optimized using the ScaledAdam [27] optimizer and the Eden [27] learning rate scheduler with an initial learning rate of 0.05. Each batch of training data contained approximately 100 seconds of audio data, including only speech samples with duration between 1 second and 15 seconds. To achieve a larger effective batch size, gradients were accumulated four times before each update. The model was trained for 20 epochs on an NVIDIA A10 GPU.

3.1.4. Vocoder setup

In each setting of the SSL extractor and k-means clustering, we trained a discrete-unit-based vocoder to convert semantic tokens into waveform⁵. This vocoder contains two conformer blocks each with 2 layers and 184 attention dimensions. After the conformer blocks, a HiFiGAN [28] is cascaded and the same learning criterions apply. We trained all the replicas for 1M steps on LibriTTS, and shared them within different acoustic BPE settings.

3.1.5. Evaluation metrics

To conduct a thorough investigation on the effectiveness of the acoustic BPE, we established the evaluation process based on a range of metrics, encompassing both subjective and objective dimensions. Contrasting with scenarios where this method is absent, these metrics serve to provide an intuitive way of gauging the impact of acoustic BPE. Below is an explanation of the metrics utilized.

Speech Intelligibility: We used a publicly available automatic speech recognition (ASR) model based on the Conformer-Transducer architecture⁶ to transcribe synthesized speech into text, and then computed the word error rate (WER) between the transcribed text and the ground truth text.

³<https://github.com/google/sentencepiece>

⁴<https://github.com/lifeiteng/vall-e>

⁵<https://github.com/X-LANCE/UniCATS-CTX-vec2wav>

⁶https://huggingface.co/nvidia/stt_en_conformer_transducer_xlarge

Table 1: Results of the decoder-only TTS model performance (WER, MOS) and inference speed (RTF) metrics under different BPE vocabulary sizes of HuBERT+kmeans (kms) 2048 centroids. Here “resyn.” means resynthesis from vocoder, while “aBPE” refers to the BPE vocabulary size.

| Tokenizer | Task | aBPE | WER↓ | MOS↑ | RTF↓ |
|--------------------|--------|-------|------------|--------------------|--------------|
| | resyn. | - | 1.9 | 4.39 ± 0.13 | - |
| HuBERT kms 2048 | TTS | - | 9.1 | 4.13 ± 0.19 | 0.129 |
| | | 5000 | 5.7 | 4.20 ± 0.17 | 0.069 |
| | | 10000 | 5.5 | 4.19 ± 0.13 | 0.052 |
| | | 20000 | 5.2 | 4.11 ± 0.19 | 0.045 |

Speech Quality and Naturalness: We employed naturalness Mean Opinion Score (MOS) as the subjective metric of speech quality and naturalness. In the listening test, each participant was assigned with multiple test cases, each containing high-intelligibility speech synthesized by different settings from the same sentence. Participants were asked to give a score of 1-5 to each speech based on its naturalness. Mel cepstral distortion (MCD) [29] with dynamic time warping (DTW), measuring the MFCC distance between the synthesized and reference mel-spectrum features with DTW helping to find the optimal time-sequence alignment, was utilized as the objective metrics.

Inference Speed: This research focuses on the inference speed of the decoder-only language model, which is the main architecture of the TTS system. Real-time factor (RTF) was measured to evaluate the inference speed.

Sample Diversity: Number of statistically-different bins (NDB) [30] and Jensen-Shannon (JS) divergence were employed to objectively evaluate the sample diversity. Detailed calculating process will be introduced in Section 3.3.1.

3.2. Acoustic BPE Enhances TTS Model Performance

After our preliminary experiments, it was found that in general situation, incorporating the acoustic BPE method indeed leads to significant improvements in the TTS model’s ability to synthesize speech. We primarily focus on intelligibility and quality of synthesized speech to reflect this ability. The results of selected outperforming settings are presented in Table 1.

3.2.1. Improvement of speech intelligibility and quality

We conducted intelligibility experiments on the entire test set, which consists of 1145 sentences. The WER of the synthesized audios ASR results are calculated. As shown in Table 1, the intelligibility performance of audio generated by TTS models using acoustic BPE method has a significant advantage over those without using it. Under the pre-configuration (tokenizer composed of HuBERT-large and a 2048-centroid kmeans model), the reduction is up to 3.9% WER. The results of the MOS metric indicate that models using the acoustic BPE method can generate competitive audio quality. In the worst-case scenario, it is slightly lower by 0.02 compared to not using it, and it is accompanied by lower confidence. However, in the best-case scenario, it can be higher by 0.07, accompanied by better confidence.

3.2.2. Acceleration of inference and training process

The calculation of RTF in this research is equivalently defined as the ratio of the decoder-only LM inference time to the length of its input. As the presented result, it significantly decreases as the expansion of its vocabulary increase. This is due to the merging operations of BPE, which shorten the sequence length of the input language model. The resulting acceleration effect of the model far exceeds the time spent on the increased com-

Table 2: The objective metrics (WER and MCD) of synthetic speech under different SSL, k-means and acoustic BPE settings. Results are presented in the format: [w/o. BPE] / [w. BPE 10000].

| SSL | HuBERT | | | | WavLM | | | |
|---------|-------------|-------------|------------------|--------------------|-------------|-------------|-------------|-------------|
| K-means | 256 | 2048 | 4096 | 8192 | 256 | 2048 | 4096 | 8192 |
| WER ↓ | 6.0 / 6.3 | 9.1 / 5.5 | 7.7 / 4.5 | 7.8 / 8.0 | 9.9 / 6.0 | 8.3 / 7.8 | 11.4 / 9.0 | 9.0 / 9.5 |
| MCD ↓ | 11.9 / 12.6 | 12.1 / 12.0 | 11.9 / 11.8 | 11.9 / 11.7 | 11.9 / 12.5 | 12.2 / 12.4 | 12.3 / 12.3 | 11.9 / 12.0 |

putational cost of token embeddings due to the expansion of the vocabulary. From an intuitive perspective, it results in up to 2.9 times inference speedup. Additionally, we record the time cost of the whole training process, the settings with acoustic BPE method also perform approximately 1.5 ~ 2 times acceleration decided to configurations.

3.3. Enrichment of sample diversity

We are interested in investigating whether the acoustic BPE method can influence the generation of more diverse intonations or speech rates when reading the same sentence. We opted to utilize NDB and JS divergence as metrics for assessing the diversity of samples in this TTS model. Experimental results from various configurations are presented in Table 3.

3.3.1. The NDB and JS divergence metrics

The NDB metric is proposed in [30]. In our experiment, we extracted prosodic features⁷ from selected low-WER synthesized utterances in the test set. Additionally, with speaker information already present in prompts, the influence of semantic and speaker information on sample diversity can be almost negligible. Samples, considered as the prosodic features of frames, were divided into a training and evaluation set. We first set a fixed number k of cluster bins by training a k-means model with samples in the training set, maintaining k centroids. Let p and q denote the statistical distributions of samples in the training and evaluation set on these bins, and let m and n denote the sample numbers of the training and evaluation set, respectively. The calculation of NDB metric was conducted with a *two-proportion z-test* on p, q and m, n , counting the number of the bins whose final p -values are less than a manually set value of *significant level*, and dividing it by k .

The definition of Jensen-Shannon (JS) divergence is

$$JS = \frac{1}{2} \left[KLD \left(p \parallel \frac{p+q}{2} \right) + KLD \left(q \parallel \frac{p+q}{2} \right) \right] \quad (4)$$

where $KLD(\cdot|\cdot)$ is the Kullback-Leibler (KL) divergence function. To reduce the impact of uncertain selection of two sample sets, we calculated two metrics using a public tool⁸ for 10 times and used their average values as the final results. Greater values of NDB and JS indicate a larger difference between the statistical distributions of the samples, suggesting better performance in terms of diversity.

3.3.2. Increment of TTS sample diversity with acoustic BPE

According to the presented results, TTS models employing the acoustic BPE method showed significant advantages in synthesized audio diversity compared to those without it. Referring to Figure 1, this phenomenon is essentially prevalent. We will discuss extreme boundary cases in detail in next section 3.4.

⁷Namely Kaldi-style [31] pitch, probability of voice and energy.

⁸<https://github.com/eitanrich/gans-ngmms/blob/master/utis/ndb.py>

Table 3: Sample diversity results. Here “kms” means kmeans centroids, and “aBPE” refers to the BPE vocabulary size.

| Tokenizer | aBPE | NDB↑ | JS↑ |
|-----------------|-------|--------------|----------------|
| | - | 0.649 | 0.00313 |
| HuBERT kms 2048 | 5000 | 0.676 | 0.00439 |
| | 10000 | 0.668 | 0.00396 |
| | 20000 | 0.678 | 0.00388 |
| | - | 0.665 | 0.00364 |
| WavLM kms 2048 | 5000 | 0.680 | 0.00415 |
| | 10000 | 0.687 | 0.00408 |
| | 20000 | 0.684 | 0.00453 |

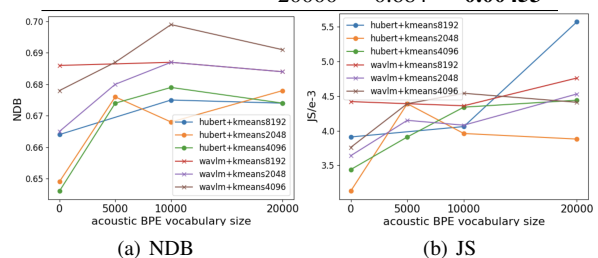


Figure 1: NDB and JS divergence results under varied semantic token and acoustic BPE settings.

3.4. Discussion on boundary cases and limitations

Although the benefits of the acoustic BPE in improving model performance are substantial, every method has its applicable range and limitations. This is particularly observed in our study, where there are configurable parameters that can be flexibly adjusted. Exploring the boundary conditions of this method under our research conditions can provide intuitive insights for constructing acoustic BPE methods in other decoder-only LM-based TTS systems with complex configurations. According to the results in Table 2, both too small and too large number of k-means centroids can lead to a plateau or even a decline in model performance. In practical experiments, when there is a significant gap between the number of k-means centroids and vocabulary size, instability may arise, e.g. leading to the model repeatedly outputting the same token. Moreover, WavLM also exhibits instability in the TTS architecture used in this experiment. At this time, the use of acoustic BPE method may exacerbate this instability, resulting in worse TTS performance.

4. Conclusion

In conclusion, the application of the acoustic BPE method in TTS tasks brings significant performance benefits, including improvements in the intelligibility, quality, and diversity of generated audio, as well as notable enhancements in training and inference speed. This undoubtedly demonstrates the potential of the method in discrete speech language modeling and speech-text language modeling. However, the presence of the acoustic BPE vocabulary increases the number of discrete tokens in speech. This imposes certain limitations on its configuration selection. In future work, we will conduct experiments with this method under scaled up datasets and models. Additionally, we will explore other effective methods for tokenizing audio.

5. Acknowledgements

This work was supported by the China NSFC Project (No. 92370206), Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102) and Development Program of Jiangsu Province, China (No.BE2022059).

6. References

- [1] K. Lakhota, E. Kharitonov, W.-N. Hsu, Y. Adi, A. Polyak, B. Bolte, T.-A. Nguyen, J. Copet, A. Baevski, A. Mohamed, and E. Dupoux, "On Generative Spoken Language Modeling from Raw Audio," *TACL*, vol. 9, pp. 1336–1354, 12 2021.
- [2] Z. Borsos, R. Marinier, D. Vincent, E. Kharitonov, O. Pietquin, M. Sharifi, D. Roblek, O. Teboul, D. Grangier, M. Tagliasacchi, and N. Zeghidour, "AudiomLM: A language modeling approach to audio generation," *IEEE/ACM TASLP*, vol. 31, pp. 2523–2533, 2023.
- [3] C. Wang, S. Chen, Y. Wu, Z. Zhang, L. Zhou, S. Liu, Z. Chen, Y. Liu, H. Wang, J. Li, L. He, S. Zhao, and F. Wei, "Neural codec language models are zero-shot text to speech synthesizers," 2023.
- [4] E. Kharitonov, D. Vincent, Z. Borsos, R. Marinier, S. Girgin, O. Pietquin, M. Sharifi, M. Tagliasacchi, and N. Zeghidour, "Speak, Read and Prompt: High-Fidelity Text-to-Speech with Minimal Supervision," *TACL*, vol. 11, pp. 1703–1718, 12 2023.
- [5] X. Zhu, Y. Lv, Y. Lei, T. Li, L. Xie, W. HE, H. Zhou, and H. Lu, "Vec-tok speech: Speech vectorization and tokenization for neural speech generation," 2024. [Online]. Available: <https://openreview.net/forum?id=C53xlgEqVh>
- [6] M. Łajszczak, G. C ambara, Y. Li, F. Beyhan, A. van Korlaar, F. Yang, A. Joly,  Alvaro Mart ın-Cortinas, A. Abbas, A. Michalski, A. Moinet, S. Karlapati, E. Muszy nska, H. Guo, B. Putrycz, S. L. Gambino, K. Yoo, E. Sokolova, and T. Drugman, "BASE TTS: Lessons from building a billion-parameter text-to-speech model on 100k hours of data," 2024.
- [7] N. Zeghidour, A. Luebs, A. Omran, J. Skoglund, and M. Tagliasacchi, "Soundstream: An end-to-end neural audio codec," *IEEE/ACM TASLP*, vol. 30, pp. 495–507, 2022.
- [8] A. D efossez, J. Copet, G. Synnaeve, and Y. Adi, "High fidelity neural audio compression," *Transactions on Machine Learning Research*, 2023, featured Certification, Reproducibility Certification.
- [9] R. Kumar, P. Seetharaman, A. Luebs, I. Kumar, and K. Kumar, "High-fidelity audio compression with improved rvqgan," in *Advances in Neural Information Processing Systems*, A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds., vol. 36. Curran Associates, Inc., 2023, pp. 27 980–27 993.
- [10] A. Baevski, S. Schneider, and M. Auli, "vq-wav2vec: Self-supervised learning of discrete speech representations," in *ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, 2020.
- [11] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhota, R. Salakhutdinov, and A. Mohamed, "HuBERT: Self-supervised speech representation learning by masked prediction of hidden units," *IEEE/ACM TASLP*, vol. 29, pp. 3451–3460, 2021.
- [12] Y.-A. Chung, Y. Zhang, W. Han, C.-C. Chiu, J. Qin, R. Pang, and Y. Wu, "w2v-bert: Combining contrastive learning and masked language modeling for self-supervised speech pre-training," in *2021 IEEE ASRU*, 2021, pp. 244–250.
- [13] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao, J. Wu, L. Zhou, S. Ren, Y. Qian, Y. Qian, J. Wu, M. Zeng, X. Yu, and F. Wei, "WavLM: Large-scale self-supervised pre-training for full stack speech processing," *IEEE JSTSP*, vol. 16, no. 6, pp. 1505–1518, 2022.
- [14] F. Shen, Y. Guo, C. Du, X. Chen, and K. Yu, "Acoustic BPE for speech generation with discrete tokens," 2024.
- [15] S. Ren, S. Liu, Y. Wu, L. Zhou, and F. Wei, "Speech pre-training with acoustic piece," in *Interspeech 2022, 23rd Annual Conference of the International Speech Communication Association, Incheon, Korea, 18-22 September 2022*, H. Ko and J. H. L. Hansen, Eds. ISCA, 2022, pp. 2648–2652.
- [16] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019*, J. Burstein, C. Doran, and T. Solorio, Eds. Association for Computational Linguistics, 2019, pp. 4171–4186.
- [17] X. Chang, B. Yan, Y. Fujita, T. Maekaku, and S. Watanabe, "Exploration of Efficient End-to-End ASR using Discretized Input from Self-Supervised Learning," in *Proc. INTERSPEECH 2023*, 2023, pp. 1399–1403.
- [18] S. Maiti, Y. Peng, S. Choi, J.-W. Jung, X. Chang, and S. Watanabe, "VoxTLM: Unified decoder-only models for consolidating speech recognition, synthesis and speech, text continuation tasks," in *ICASSP 2024*, 2024, pp. 13 326–13 330.
- [19] C. Du, Y. Guo, X. Chen, and K. Yu, "VqTts: High-fidelity text-to-speech synthesis with self-supervised vq acoustic feature," *arXiv preprint arXiv:2204.00768*, 2022.
- [20] C. Du, Y. Guo, F. Shen, Z. Liu, Z. Liang, X. Chen, S. Wang, H. Zhang, and K. Yu, "UniCATS: A unified context-aware text-to-speech framework with contextual VQ-diffusion and vocoding," *CoRR*, vol. abs/2306.07547, 2023.
- [21] H. Zen, V. Dang, R. Clark, Y. Zhang, R. J. Weiss, Y. Jia, Z. Chen, and Y. Wu, "LibriTTS: A corpus derived from librispeech for text-to-speech," in *Interspeech 2019*, G. Kubin and Z. Kacic, Eds. ISCA, 2019, pp. 1526–1530.
- [22] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *ACL 2016, August 7-12, 2016*.
- [23] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "LibriSpeech: An ASR corpus based on public domain audio books," in *Proc. ICASSP 2015*. IEEE, 2015, pp. 5206–5210.
- [24] J. Kahn, M. Riviere, W. Zheng, E. Kharitonov, Q. Xu, P. Mazar e, J. Karadayi, V. Liptchinsky, R. Collobert, C. Fuegen, T. Likhomanenko, G. Synnaeve, A. Joulin, A. Mohamed, and E. Dupoux, "Libri-light: A benchmark for asr with limited or no supervision," in *Proc. ICASSP 2020*, 2020, pp. 7669–7673.
- [25] G. Chen, S. Chai, G. Wang, J. Du, W. Zhang, C. Weng, D. Su, D. Povey, J. Trmal, J. Zhang, M. Jin, S. Khudanpur, S. Watanabe, S. Zhao, W. Zou, X. Li, X. Yao, Y. Wang, Z. You, and Z. Yan, "GigaSpeech: An evolving, multi-domain ASR corpus with 10, 000 hours of transcribed audio," in *Interspeech 2021, Brno, Czechia 2021*. ISCA, 2021, pp. 3670–3674.
- [26] C. Wang, M. Riviere, A. Lee, A. Wu, C. Talnikar, D. Haziza, M. Williamson, J. M. Pino, and E. Dupoux, "Voxpopuli: A large-scale multilingual speech corpus for representation learning, semi-supervised learning and interpretation," in *Proc. ACL/IJCNLP 2021, August 1-6, 2021*, pp. 993–1003.
- [27] Z. Yao, L. Guo, X. Yang, W. Kang, F. Kuang, Y. Yang, Z. Jin, L. Lin, and D. Povey, "Zipformer: A faster and better encoder for automatic speech recognition," in *12th ICLR*, 2024.
- [28] J. Kong, J. Kim, and J. Bae, "HiFi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis," in *NeurIPS 2020, December 6-12, 2020*.
- [29] R. Kubichek, "Mel-cepstral distance measure for objective speech quality assessment," in *Proceedings of IEEE Pacific Rim Conference on Communications Computers and Signal Processing*, vol. 1, 1993, pp. 125–128 vol.1.
- [30] E. Richardson and Y. Weiss, "On GANs and GMMs," in *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., vol. 31. Curran Associates, Inc., 2018.
- [31] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The Kaldi speech recognition toolkit." IEEE Signal Processing Society, 2011, IEEE Catalog No.: CFP11SRW-USB.