



# PAST: Phonetic-Acoustic Speech Tokenizer

Nadav Har-Tuv, Or Tal, Yossi Adi

The School of Computer Science and Engineering  
The Hebrew University of Jerusalem, Israel

nadav.har-tuv1@mail.huji.ac.il

## Abstract

We present PAST, a novel end-to-end framework that jointly models phonetic information alongside signal reconstruction, eliminating the need for external pretrained models. Unlike previous approaches that rely on pretrained self-supervised models, PAST employs supervised phonetic data, directly integrating domain knowledge into the tokenization process via auxiliary tasks. Additionally, we introduce a streamable, causal variant of PAST, enabling real-time speech applications. Results demonstrate that PAST surpasses existing evaluated baseline tokenizers across common evaluation metrics, including phonetic representation and speech reconstruction. Notably, PAST also achieves superior performance when serving as a speech representation for speech language models, further highlighting its effectiveness as a foundation for spoken language generation. To foster further research, we release the full implementation. Code, model checkpoints, and samples see [pages.cs.huji.ac.il/adiyoss-lab/PAST](https://pages.cs.huji.ac.il/adiyoss-lab/PAST).

**Index Terms:** Speech Tokenization, Phonetic and Acoustic Tokens, Speech Language Models

## 1. Introduction

Speech and audio language models have recently attracted significant attention in the research community, showcasing remarkable performance across a range of tasks [1–6]. These models usually operate over acoustic tokens or phonetic speech tokens (also known as semantic tokens). Acoustic tokenizers, such as EnCodec [7] and SoundStream [8], are designed for high-fidelity waveform reconstruction but are less ideal for language modeling without an external text supervision. In contrast, phonetic tokenizers, such as those derived from quantizing wav2vec 2.0 [9] and HuBERT [10] latent representations, primarily capture linguistic information [11]. This makes them more suitable for sequential modeling. However, they require an additional vocoder module [12] for speech synthesis, which increases complexity and can often degrade reconstruction quality.

Hybrid tokenizers, such as [13–15], integrate phonetic and acoustic information into a unified representation. This is achieved by extracting and distilling phonetic features from pretrained self-supervised models (SSL models), e.g. WavLM [16]. Self-supervised representations exhibit correlation to phonetic content, e.g. measured by phoneme mutual information [10, 11], but they are not explicitly optimized to capture it. Despite showing promise, due to their dependence on pretrained SSL models, such hybrid approaches are potentially limited in their ability to fully capture the phonetic richness of the input. That being said, explicit phonetic supervision could not only benefit the tokenizer’s ability to capture phonetic con-

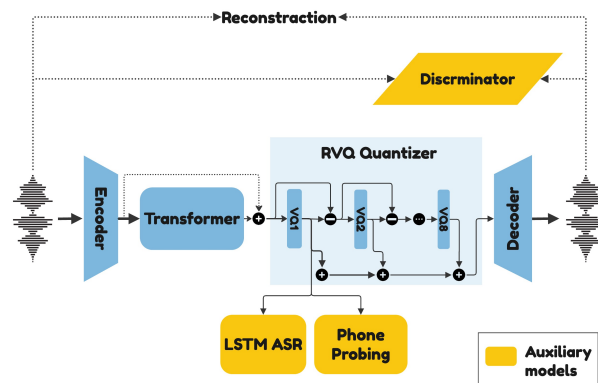


Figure 1: Schematic of the PAST pipeline. The auxiliary heads use the output of the first vector quantization module as input.

tent, it could also replace the dependence on pretrained SSL models and lower the computational cost.

In this work, we introduce Phonetic-Acoustic Speech Tokenizer (PAST), a novel end-to-end tokenizer that jointly captures both phonetic and acoustic representations without requiring external pretrained models or vocoders (see Figure 1). In addition to the standard model, we present a variant of PAST designed for streaming applications, which operates causally by relying only on past information. PAST achieves this by leveraging supervised auxiliary tasks, such as phoneme classification and automatic speech recognition, to incorporate phonetic information directly into the quantization process. This approach allows PAST to outperform existing methods in phonetic and acoustic benchmarks while maintaining a simpler pipeline.

**Our contributions** are as follows:

1. We propose a novel approach for jointly learning phonetic and acoustic representations using supervised data, eliminating the need for pretrained models or external vocoders.
2. We achieve superior performance compared to hybrid tokenizers across both phonetic and acoustic benchmarks, demonstrating the effectiveness of our approach.
3. We introduce a streaming-compatible variant of PAST that operates causally, ensuring that it relies only on previous context, making it suitable for real-time speech applications.
4. We open-source our implementation, including training and inference pipelines in addition to model checkpoints.

## 2. Related Work

Approaches to tokenization can be roughly categorized into phonetic tokenizers, acoustic tokenizers, and hybrid-tokenizers. **Acoustic tokenizers** aim to compress speech into discrete rep-

representations optimized for high-fidelity reconstruction. EnCodec [7] and SoundStream [8] have set the foundation for streamable high-fidelity audio compression by introducing Residual Vector Quantization (RVQ). Subsequent works focused on enhancing fidelity through (i) improved quantization [17, 18] (ii) improving latent representation structure [19] (iii) increasing training stability [20] or (iv) utilizing spectral representations [21, 22].

**Phonetic tokenizers**, on the other hand, quantize latent representations obtained by speech encoder model such as HuBERT [10], wav2vec 2.0 [9], or Whisper encoder [23, 24] are trained to maximize mutual information across latent sequences of vectors, yet without explicit reconstruction constraints. These types of models usually operate over continuous latent spaces, and hence require some discretization method like k-means, or learned clustering [25, 26], to tokenize continuous embeddings into discrete tokens, enabling downstream modeling. Many studies have demonstrated that these tokens effectively capture phonetic information [1, 2, 4, 6]. They exhibit strong correlations with phonemes [11], and their representations have been successfully used to improve performance in speech applications [27] and to train Speech Language Models (SLM), with a vocoder component employed to synthesize speech from generated tokens.

**Hybrid tokenizers** aimed at merging phonetic and acoustic tokens. The common approach nowadays is to leverage a pre-trained phonetic teacher model as auxiliary objective for distillation guidance. These approaches aim to jointly capture phonetic information while maintaining signal reconstruction capabilities. SpeechTokenizer [13] applies phonetic distillation to the first RVQ codebook, treating its indices as phonetic tokens while later codebooks encode acoustic details. This hierarchical approach enhances phonetic representation while maintaining signal quality. X-Codec [14] utilizes a pretrained SSL model during training and inference. It concatenate a low rank projection of the latent representation to the acoustic encoder output, in addition to applying an auxiliary distillation objective over the post-quantization latent. Mimi, the tokenizer used by Moshi [15], is designed to condense phonetic information into a single quantized stream, which is then combined with the encoded acoustic quantized latent representation.

Despite their strong performance, approaches relying on pseudo-labels from SSL models have several limitations. First, by relying on pseudo-supervision instead of supervised data, these approaches fail to fully utilize domain knowledge and may not align with explicit phonetic structures. Second, these models require vast amounts of unlabeled speech data, making their training computationally expensive and less feasible for low-resource languages. Finally, the inherent distillation process may encode cross-vector properties that are not directly relevant to phonetic representation, leading to redundancy and inefficiencies in the learned token space.

## 3. Method

### 3.1. Problem Setup

Our model is composed of three main components: Encoder, Quantizer, and Decoder. Given a waveform signal  $\mathbf{x} \in \mathbb{R}^{f_s \cdot t}$  of duration  $t$ [sec], sampled at  $f_s$ [Hz], the encoder transforms  $\mathbf{x}$  into a dense latent representation  $\mathbf{z} \in \mathbb{R}^{D \times T}$ . Here,  $T = f_r \cdot t$  denotes the temporal resolution of the latent space, determined by the frame rate  $f_r$ , and  $D$  represents the latent dimension. The Quantizer module then processes  $\mathbf{z}$ , producing a quantized

latent representation  $\hat{\mathbf{z}} \in \mathbb{R}^{D \times T}$ . Finally, the Decoder reconstructs the original signal, yielding  $\hat{\mathbf{x}} \in \mathbb{R}^{f_s \cdot t}$ . To encourage capturing of phonetic content in the encoded latent representation we use paired phoneme and character-level transcription supervision with a set of auxiliary losses as described in subsection 3.3. Our goal is then to (i) minimize the reconstruction error between  $\mathbf{x}$  and  $\hat{\mathbf{x}}$ , and (ii) to ensure that the encoded latent representation  $\mathbf{z}$  captures meaningful phonetic information.

### 3.2. Model Architecture

Our model is built on top of EnCodec [7] with an addition of a transformer encoder prior to the quantization module as depicted in Figure 1. The Encoder block is comprised from a convolutional encoder module, same as EnCodec’s encoder, followed by a transformer encoder module. Experiments with and without the transformer encoder module are presented in Table 4 and discussed in Section 5.3.

To enhance training stability, during training, the input to the quantization module is chosen from one of three modes of operations: (i) the output of the transformer block, with probability  $p_{\text{rms-only}}$ ; (ii) the output of the encoder (skip connection), with probability  $p_{\text{skip-only}}$ ; or (iii) the average of (i) and (ii). During inference, the model is using the averaged representation only. For the quantization module we employ Residual Vector Quantization. The RVQ component contains  $N_q$  sequential vector quantization (VQ) layers, which iteratively quantize  $\mathbf{z}$  along with its residuals. Specifically, given  $\mathbf{z} \in \mathbb{R}^{D \times T}$ , for every  $\tau \in T$  the first VQ module replaces  $\mathbf{z}_\tau$  with the closest entry in a learned embedding table yielding  $\hat{\mathbf{z}}_1$ . Then the process is repeated in the next VQ layers over the residue, i.e. for  $i \in \{2, \dots, N_q\} : \text{VQ}_i(\mathbf{z} - \sum_{j \in [i-1]} \hat{\mathbf{z}}_j) = \hat{\mathbf{z}}_i$ . The RVQ module outputs  $N_q$  quantized streams, which can be represented either as the quantized vectors  $\hat{\mathbf{z}}_i$  or as their corresponding indices  $\mathbf{q}_i \in \mathbb{N}^T$  in the embedding table of each  $\text{VQ}_i$ . The Decoder component mirrors the convolutional encoder module, replacing the strided convolutions with transposed convolution layers. The input to the decoder is  $\hat{\mathbf{z}} = \sum_{i \in [N_q]} \hat{\mathbf{z}}_i$ .

**Streamable configuration.** We introduce a streamable variant of our model with three key modifications: (i) causal convolutions utilizing left-only padding following [7]; (ii) unidirectional LSTM and (iii) causal attention. This setup requires a 20ms look-ahead window over the audio signal.

### 3.3. Auxiliary Heads

To encourage the encoding of phonetic information, we incorporate auxiliary heads and training objectives that operate over the first quantized output stream  $\hat{\mathbf{z}}_1$ . This approach aims to replace the distillation of pseudo-phonetic tokens with direct supervision using target character transcriptions and phonemes.

**CTC character match.** The CTC auxiliary head takes  $\hat{\mathbf{z}}_1 \in \mathbb{R}^{D \times T}$  as input and outputs a distribution over the set of all characters  $M$  at each entry  $\mathbf{y} \in \mathbb{R}^{|M| \times T}$ . The module is constructed from a linear projection from  $D$  to a hidden dimension  $h$  followed by a single-layer BiLSTM, and a linear projection from  $h$  to  $|M|$ . A connectionist temporal classification (CTC) [28] loss, denoted as  $\mathcal{L}_{\text{ctc}} = \text{CTC}(\mathbf{y}|\text{chars})$ , is then applied to align the predicted sequence with the transcription target.

**Phoneme classification.** The second auxiliary head is a simple linear projection that takes  $\hat{\mathbf{z}}_1$  as input and outputs a distribution over the set of all phonemes  $P$  at each entry  $\hat{\mathbf{p}} \in \mathbb{R}^{|P| \times T}$ . The auxiliary head is trained using a cross-entropy objective, denoted as  $\mathcal{L}_{\text{phn}} = \text{CE}(\hat{\mathbf{p}}, \mathbf{p})$ .

Table 1: Comparison based on phonetic information.

Tokenizer	PNMI $\uparrow$	ABX $\downarrow$		WER $\downarrow$	
		Within	Across	Clean	Other
D. HuBERT 500	0.67	3.91	4.73	11.3	24.7
SpeechTokenizer	0.72	3.43	4.50	18.5	41.3
X-Codec	0.40	9.42	12.6	17.1	37.1
<b>PAST</b>	<b>0.75</b>	<b>2.82</b>	<b>3.54</b>	15.7	36.8
<b>PAST - Streamable</b>	0.74	3.05	3.89	<b>14.3</b>	<b>32.3</b>

### 3.4. Training Objective

We use the reconstruction training objective as defined in EnCodec’s [7] paper, with the addition of two auxiliary terms. EnCodec contains multiple training objectives, working in a weighted combination to optimize signal reconstruction. As we do not change the recommended weighted configuration of the involved objectives, we will refer to the overall EnCodec objective term as  $\mathcal{L}_{\text{EnCodec}}$  and refrain from defining all of the components and notations here. For further details, please refer to [7]. The overall training objective minimizes the following:

$$\mathcal{L} = \lambda_{\text{ctc}} \mathcal{L}_{\text{ctc}} + \lambda_{\text{phn}} \mathcal{L}_{\text{phn}} + \mathcal{L}_{\text{EnCodec}} \quad (1)$$

where  $\lambda_{\text{ctc}}$  and  $\lambda_{\text{phn}}$  control the weights of the CTC loss  $\mathcal{L}_{\text{ctc}}$  and phoneme loss  $\mathcal{L}_{\text{phn}}$ , respectively.

## 4. Experimental Setup

### 4.1. Data

We use all training subsets of LibriSpeech [29] and TIMIT [30] for our training set, yielding a total of 965 hours of raw audio. To improve training efficiency and avoid redundant padding, we sample 3-second audio segments from each data sample. This segmentation is done purely for efficiency and is not an inherent part of our approach. To achieve this, we obtain character-level alignment for paired text transcriptions using pretrained Wav2Vec2 [9] model. For each sampled audio segment, Wav2Vec2 outputs a character distribution per temporal entry, from which we select the most probable alignment based on log-likelihood. For the phoneme classification, we utilize the phonetic transcriptions provided by the TIMIT dataset. We sample instances in a 9:1 ratio (LS:T), resulting in 10% of batch samples having phoneme supervision.

### 4.2. Model Configuration

The encoder and decoder architectures follow the configuration from EnCodec [7] considering temporal downscaling with ratios [8, 5, 4, 2], resulting in a frame rate of 50 Hz. The latent dimensionality  $D$  is set to 128. The transformer module includes 8 layers with a hidden size of 768, 16 attention heads, and a feed-forward size of 2048. During training, the sampling probabilities for the transformer skip-connection are  $p_{\text{trns.-only}} = 0.3$  and  $p_{\text{skip-only}} = 0.1$ . The transformer processes up to 150 frames (3 seconds), and longer sequences are divided into chunks with 1-second overlap, averaging the results over the overlapping regions. The auxiliary loss weights are  $\lambda_{\text{ctc}} = 12$  and  $\lambda_{\text{phn}} = 5$ , while the reconstruction loss weights follow the configuration defined in [7]. The RVQ contains 8 codebooks, each with 1024 entries. The dimension of the CTC auxiliary head is  $h = 512$ . The model has 185M parameters, with 68M allocated to the transformer and 4M to the auxiliary heads.

Table 2: Signal reconstruction evaluation.

Tokenizer	SISNR $\uparrow$	VISQOL $\uparrow$	PESQ $\uparrow$
EnCodec	7.49	4.48	3.88
SpeechTokenizer	0.44	4.38	3.15
X-Codec	-7.12	<b>4.46</b>	3.33
<b>PAST</b>	<b>4.84</b>	4.40	<b>3.55</b>
<b>PAST - Streamable</b>	3.90	4.37	3.40

### 4.3. Training Configuration

Training is performed using two *NVIDIA A100* GPUs with a batch size of 80 for a total of 400,000 steps, using the ADAM optimizer with [0.5, 0.9] betas and no weight decay. The learning rate was managed using a cosine decay scheduler, starting at  $3 \cdot 10^{-4}$  and decreasing gradually to zero with a warm-up phase of 4,000 steps.

### 4.4. Evaluation Metrics

#### 4.4.1. Signal Reconstruction Quality Metrics

The following are computed on the LibriSpeech clean-test set. **Virtual Speech Quality Objective Listener (ViSQOL)** [31] evaluates the reconstruction quality by comparing the spectral and temporal features of the reconstructed signal to the source signal. It produces an approximation of Mean Opinion Score. **Scale-Invariant Signal-to-Noise Ratio (SISNR)** measures the similarity between the original and reconstructed signals by quantifying the ratio of target signal energy to residual noise. **PESQ (Perceptual Evaluation of Speech Quality)** [32] assesses the perceptual degradation of reconstructed signals, following the ITU-T P.862.2 wideband recommendation.

#### 4.4.2. Phonetic Information Evaluation

**Phone-Normalized Mutual Information (PNMI)** [10] quantifies the percentage of uncertainty about a given phone label  $Y$  eliminated after observing a token  $X$ . It measures the mutual information  $I(X; Y)$ , normalized by the entropy of  $Y$ . Higher values correspond to better phonetic encoding. In our evaluation, PNMI is computed on the tokenizer’s first RVQ codebook token  $q_1$  using the TIMIT test-set.

**ABX** metric [33] measures the model’s ability to discriminate phonetic contrasts by comparing triplets of sounds  $A$ ,  $B$ , and  $X$ . Lower ABX error rates reflect better phonetic preservation. We compute ABX on the reconstructed representation after the RVQ, evaluating both within-speaker and across-speaker contexts. This evaluates how well the tokenizer retains phonetic distinctions through the quantization process.

**Word Error Rate (WER)** measures the accuracy of generated transcriptions with respect to a reference transcription, where lower values indicate better performance. WER is computed using the DASB benchmark [34] on the discrete tokens from all codebooks  $q$ . Training and validation were performed on LibriSpeech *train-clean-100* and *dev-clean* subsets, while testing was conducted on the *test-clean* and *test-other* subsets.

#### 4.4.3. Speech Language Modeling Evaluation

In order to compare the observed methods, we train an identical backbone language model (LM) for each of the observed tokenizers. We leverage the base architecture of the AudioGen model [35] using the delay pattern, as suggested in MusicGen [36]. We use a 300M parameter model configuration and train on 10-second audio segments for 300k update steps

Table 3: Evaluation of SLM performance across tokenizers.

Tokenizer	sWUGGY $\uparrow$	
	Inter	Oov
EnCodec	56.3	53.7
D. HuBERT 500	67.9	55.4
SpeechTokenizer	63.7	55.6
X-Codec	55.1	52.9
<b>PAST</b>	<b>71.8</b>	<b>57.5</b>
<b>PAST - Streamable</b>	70.2	56.3

with batch size of 256, using all train subsets of LibriSpeech without any additional conditioning, text or prompts. Equipped with this LM, we measure the **SWUGGY** metric [37]. This metric evaluates the model’s ability to assign higher likelihood to valid words over pseudo-words by testing both Inter (within-vocabulary) and OOV (out-of-vocabulary) categories. The probabilities are derived solely from the distribution of the first codebook, as it encapsulates the majority of the phonetic information in the output.

## 5. Results

### 5.1. Baseline Comparison

We compare PAST with two baseline hybrid models, SpeechTokenizer and X-Codec, on both reconstruction and phonetic information metrics. For context, we use EnCodec as an acoustic topline, since its objective is purely signal fidelity without phonetic supervision. We have discarded Mimi [15] from this comparison as our reproduced results, using the published model, were far from being on par with the observed baselines. We also compare PAST with two top-line models: k-means over HuBERT for phonetic representation, and EnCodec for acoustic reconstruction. First, we evaluate the observed methods over a set of phonetic metrics. Results depicted in Table 1 suggests that PAST outperforms the observed baselines across all observed metrics, even surpassing the topline on several key aspects. These results demonstrate that direct supervision effectively captures phonetic information, eliminating the need for distillation from pretrained SSL models.

Next, we evaluate PAST’s signal generation capabilities. Table 2 presents the results for reconstruction quality evaluation. PAST significantly surpasses the observed baselines, further showcasing its ability to balance phonetic richness with acoustic quality. We speculate that X-Codec’s notably low SISNR is due to the absence of a point-wise metric loss objective over the reconstructed waveform, leading to misalignment in signal reconstruction. Moreover, Tables 1, 2 also highlight PAST’s superiority while utilizing a causal modeling configuration which allows for streaming capabilities.

### 5.2. Speech Language Modeling

The following experiment revolves around the evaluation of the observed methods as speech tokenizers, where we train an identical LM as specified in subsection 4.4.3 using each method to tokenize the audio signals to discrete token sequences. Results summarized in Table 3 shows that PAST notably surpasses all other observed models, further emphasizing the advantages of our data-driven approach.

### 5.3. Component Analysis

Table 4 highlights the impact of the different components that compose PAST and reveals two main trends. First, including

Table 4: Ablation study over model components.

Trns	Phn clf	CTC	PNMI $\uparrow$		ABX $\downarrow$		WER $\downarrow$		SISNR $\uparrow$
			W.in	Ac.	Cl.	Oth.			
X	X	X	0.32	21.2	26.7	87.3	93.7	7.49	
V	X	X	0.35	19.4	25.5	93.8	96.6	7.02	
V	V	X	0.71	16.0	22.3	43.0	71.8	5.74	
V	X	V	0.50	4.42	5.51	15.2	35.8	4.94	
X	V	V	0.73	3.48	4.28	15.8	36.6	5.24	
V	V	V	0.75	2.82	3.54	15.7	36.8	4.84	

Table 5: Impact of transformer skip-connection dropout.

Mode	PNMI $\uparrow$		ABX $\downarrow$		WER $\downarrow$		SISNR $\uparrow$
	W.in	Ac.	Cl.	Oth.			
Skip - No Drop	0.74	3.93	4.96	15.5	36.5	5.24	
<b>PAST</b>	0.75	2.82	3.54	15.7	36.8	4.84	

the suggested auxiliary heads and their corresponding training objectives is crucial for encapsulating phonetic information in the learned latent space, with CTC being the most crucial of the two, as reflected in the ABX scores. Since the CTC objective optimizes all possible alignments between the latent representation and the character transcription, we hypothesize that it encourages the latent space to encode phonetic information in a structured manner across time steps. Second, considering both auxiliary objectives without the transformer module leads to a significant difference across all metrics. However, including it improves sequence modeling and signal reconstruction, as reflected in the ABX and SISNR trends.

Practically, adding the transformer module and naively cascading it after the encoder leads to vanishing gradients, causing divergence during training. Table 5 highlights the importance of including the suggested skip connection dropout during training. By design, we utilize small auxiliary heads with limited sequence modeling capacity to ensure that certain modeling capabilities are captured at the latent representation level. Our motivation for adding the transformer component is to enhance the model’s sequence modeling capacity. Without dropout on the skip connections, the model effectively bypassed the transformer encoder, leading to similar performance as the configuration where the transformer is entirely omitted, as shown in Table 4. Hence, imposing the dropout constraint empirically shows notable improvement and prevents this phenomenon.

## 6. Discussion

In this work, we introduce PAST, a novel and unified phonetic-acoustic speech tokenizer that integrates supervised phonetic information into the tokenization process while maintaining high-fidelity signal reconstruction. Unlike existing approaches that rely on SSL models and external vocoders, PAST directly incorporates phonetic supervision, producing a representation that is both semantically meaningful and acoustically precise. Our results demonstrate that PAST surpasses state-of-the-art tokenizers in phonetic representation, speech reconstruction, and speech language modeling. Notably, its supervised design eliminates the need for pretrained SSL models. Furthermore, the proposed streamable variant extends PAST’s applicability to real-time speech applications. Although PAST offers several benefits, its reliance on labeled phonetic data constrains its scalability to a multilingual setting. Future work will focus on adapting PAST to a multilingual setting.

## 7. References

- [1] R. Algayres *et al.*, “Generative spoken language model based on continuous word-sized audio tokens,” in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2023, pp. 3008–3028.
- [2] M.-J. Hwang *et al.*, “Textless acoustic model with self-supervised distillation for noise-robust expressive speech-to-speech translation,” *Findings of the Association for Computational Linguistics: ACL 2024*, 2024.
- [3] Z. Borsos *et al.*, “Audiolm: a language modeling approach to audio generation,” *IEEE/ACM transactions on audio, speech, and language processing*, vol. 31, pp. 2523–2533, 2023.
- [4] E. Kharitonov *et al.*, “Text-free prosody-aware generative spoken language modeling,” *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8666–8681, 2022.
- [5] K. Lakhota *et al.*, “On generative spoken language modeling from raw audio,” *Transactions of the Association for Computational Linguistics*, vol. 9, pp. 1336–1354, 2021.
- [6] M. Hassid *et al.*, “Textually pretrained speech language models,” *Advances in Neural Information Processing Systems*, vol. 36, 2024.
- [7] A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, “High fidelity neural audio compression,” *arXiv preprint: 2210.13438*, 2022.
- [8] N. Zeghidour, A. Luebs, A. Omran, J. Skoglund, and M. Tagliasacchi, “Soundstream: An end-to-end neural audio codec,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 495–507, 2021.
- [9] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, “wav2vec 2.0: A framework for self-supervised learning of speech representations,” *Advances in neural information processing systems*, vol. 33, pp. 12 449–12 460, 2020.
- [10] 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 transactions on audio, speech, and language processing*, vol. 29, pp. 3451–3460, 2021.
- [11] A. Sichertman and Y. Adi, “Analysing discrete self supervised speech representation for spoken language modeling,” in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, p. 1–5.
- [12] J. Kong, J. Kim, and J. Bae, “Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis,” *Advances in neural information processing systems*, vol. 33, pp. 17 022–17 033, 2020.
- [13] X. Zhang, D. Zhang, S. Li, Y. Zhou, and X. Qiu, “Spechtok-enizer: Unified speech tokenizer for speech language models,” in *The Twelfth International Conference on Learning Representations*, 2024.
- [14] Z. Ye *et al.*, “Codec does matter: Exploring the semantic shortcoming of codec for audio language model,” *arXiv preprint: 2408.17175*, 2024.
- [15] A. Défossez, L. Mazaré, M. Orsini, A. Royer, P. Pérez, H. Jégou, E. Grave, and N. Zeghidour, “Moshi: a speech-text foundation model for real-time dialogue,” *Technical report, Kyutai*, 2024.
- [16] S. Chen *et al.*, “Wavlm: Large-scale self-supervised pre-training for full stack speech processing,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, p. 1505–1518, Oct. 2022.
- [17] D. Yang, S. Liu, R. Huang, J. Tian, C. Weng, and Y. Zou, “Hifi-codec: Group-residual vector quantization for high fidelity audio codec,” *arXiv preprint: 2305.02765*, 2023.
- [18] A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, “High fidelity neural audio compression,” *Transactions on Machine Learning Research*, 2023, featured Certification, Reproducibility Certification.
- [19] J. Chen *et al.*, “Pyramidcodec: Hierarchical codec for long-form music generation in audio domain,” in *Findings of the Association for Computational Linguistics: EMNLP 2024*, 2024, pp. 4253–4263.
- [20] Y.-C. Wu *et al.*, “Audiodec: An open-source streaming high-fidelity neural audio codec,” in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.
- [21] Z. Du, S. Zhang, K. Hu, and S. Zheng, “Funcodec: A fundamental, reproducible and integrable open-source toolkit for neural speech codec,” in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 591–595.
- [22] S. Ji *et al.*, “Wavtokenizer: an efficient acoustic discrete codec tokenizer for audio language modeling,” 2024.
- [23] A. Zeng, Z. Du, M. Liu, L. Zhang, S. Jiang, Y. Dong, and J. Tang, “Scaling speech-text pre-training with synthetic interleaved data,” *arXiv preprint: 2411.17607*, 2024.
- [24] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, “Robust speech recognition via large-scale weak supervision,” in *International conference on machine learning*. PMLR, 2023, pp. 28 492–28 518.
- [25] A. Turetzy and Y. Adi, “Last: Language model aware speech tokenization,” *arXiv preprint: 2409.03701*, 2024.
- [26] S. Messica and Y. Adi, “Nast: Noise aware speech tokenization for speech language models,” *arXiv preprint: 2406.11037*, 2024.
- [27] O. Tal, M. Mandel, F. Kreuk, and Y. Adi, “A systematic comparison of phonetic aware techniques for speech enhancement,” in *Interspeech 2022*, 2022, pp. 1193–1197.
- [28] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, “Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks,” in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 369–376.
- [29] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, “Librispeech: An asr corpus based on public domain audio books,” in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 5206–5210.
- [30] J. Garofolo, L. Lamel, W. Fisher, J. Fiscus, D. Pallett, N. Dahlgren, and V. Zue, “Timit acoustic-phonetic continuous speech corpus,” *Linguistic Data Consortium*, 11 1992.
- [31] M. Chinen, F. S. Lim, J. Skoglund, N. Gureev, F. O’Gorman, and A. Hines, “Visqol v3: An open source production ready objective speech and audio metric,” in *2020 twelfth international conference on quality of multimedia experience (QoMEX)*. IEEE, 2020, pp. 1–6.
- [32] A. Rix, J. Beerends, M. Hollier, and A. Hekstra, “Perceptual evaluation of speech quality (pesq)-a new method for speech quality assessment of telephone networks and codecs,” in *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.01CH37221)*, vol. 2, 2001, pp. 749–752 vol.2.
- [33] T. Schatz *et al.*, “Evaluating speech features with the minimal-pair abx task: analysis of the classical mfc/plp pipeline,” in *Interspeech 2013*, 2013, pp. 1781–1785.
- [34] P. Mousavi, L. D. Libera, J. Duret, A. Ploujnikov, C. Subakan, and M. Ravanelli, “Dasb - discrete audio and speech benchmark,” *arXiv preprint: 2406.14294*, 2024.
- [35] F. Kreuk, G. Synnaeve, A. Polyak, U. Singer, A. Défossez, J. Copet, D. Parikh, Y. Taigman, and Y. Adi, “Audiogen: Textually guided audio generation,” *arXiv preprint: 2209.15352*, 2023.
- [36] J. Copet, F. Kreuk, I. Gat, T. Remez, D. Kant, G. Synnaeve, Y. Adi, and A. Défossez, “Simple and controllable music generation,” *Advances in Neural Information Processing Systems*, vol. 36, 2024.
- [37] T. A. Nguyen *et al.*, “The zero resource speech benchmark 2021: Metrics and baselines for unsupervised spoken language modeling,” *Self-Supervised Learning for Speech and Audio Processing Workshop @ NeurIPS*, 2020.