Interspeech 2025
17-21 August 2025, Rotterdam, The Netherlands

SpeechSEC: A Unified Multi-Task Framework for Speech Synthesis, Editing, and Continuation

Liming Liang[1], *Dongchao Yang*[2], *Xianwei Zhuang*[1], *Yuxin Xie*[1], *Luo Chen*[1], *Yuehan Jin*[3], *Yuexian Zou*[*1]

[1]Guangdong Provincial Key Laboratory of Ultra High Definition Immersive Media Technology, Shenzhen Graduate School, Peking University, China; [2]The Chinese University of Hong Kong, China; [3]South China University of Technology, China

{limingliang,xwzhuang}@stu.pku.edu.cn,dcyang@se.cuhk.edu.hk,zouyx@pku.edu.cn

Abstract

Recent advancements in non-autoregressive single-task speech synthesis have garnered significant attention. However,traditional single-task speech synthesis methods focus primarily on mapping semantic tokens to acoustic tokens, which overlooking the internal relationships within acoustic features. Addressing this gap, we propose SpeechSEC, a unified multi-task framework designed for Speech **S**ynthesis, **E**diting, and **C**ontinuation tasks by dynamically adjusting input conditions. SpeechSEC not only surpasses previous state-of-the-art method in audio quality (4.20 vs 4.00), and voice preservation (0.72 vs 0.58) for synthesis task by acquiring shared knowledge, but also efficiently executes editing and continuation tasks with good performance via non-autoregressive techniques. Additionally, SpeechSEC exhibits a strong adaptability to current speech discretization methods, like Hubert, Descript-Audio-Codec and SpeechTokenizer, which showcases robustness of our approach. Audio samples are available.[1]

Index Terms: multi-task learning, speech synthesis, speech editing, speech continuation

1. Introduction

The demand for audio generation spans various applications, from generating speech with a specific voice from text (or semantic tokens) using speech synthesis models [1, 2, 3, 4, 5, 6, 7, 8, 9], to detailed editing of speech segments (e.g., modifying 'Jack is a good student' to 'Jack is an excellent and smart student') [10, 11, 12, 13, 14, 15, 16], and generating continuations of speech [17, 18, 19, 20, 21].

Notable frameworks and methods like AudioLM [17], SoundStream [22], and SoundStorm [23] have paved the way for speech synthesis, editing, and continuation. They introduced the transition of speech from continuous to discrete domains by tokenizing speech into semantic and acoustic tokens, enabling the use of Transformer-based models [24, 25, 26] for audio generation. AudioLM achieved high-quality audio generation by treating the task as language modeling but uses an autoregressive approach. SoundStorm improved on this by employing non-autoregressive methods, using bidirectional attention and parallel decoding to increase generation speed. Pheme [27] further optimized conversational speech generation by using SpeechTokenizer [28] with smaller-scale data, improving efficiency and real-time performance.

Inspired by these breakthroughs, we propose **SpeechSEC**, which aims to offer a versatile solution capable of handling various audio processing tasks within a single, cohesive architecture. The driving motivation behind SpeechSEC is to harness the rapid, high-quality generation capabilities of non-autoregressive models like SoundStorm, coupled with leveraging the sophisticated masking techniques and multi-task joint training schemes found in MAGVIT [29] to create a multi-task framework that excels in speech synthesis, editing, and continuation.

Our approach, through multi-task training, aims for the model to acquire diverse knowledge across different tasks. For instance, in speech synthesis task, it learns to predict acoustic tokens from semantic tokens. In speech continuation tasks, it grasps the relationships internally between acoustic tokens, predicting subsequent acoustic information based solely on previous acoustic tokens. In speech editing tasks, it enhances capacity for seamless transitions and natural integration between speech segments. SpeechSEC, our multi-task training framework, not only surpasses single-task training in speech synthesis but also outperforms the state-of-the-art method (SoundStorm [23]) in both audio quality and voice preservation, as demonstrated in our results. SpeechSEC also efficiently handles the other two tasks (editing and continuation) with high speed and quality, leveraging its non-autoregressive nature. Moreover, we prove our method's adaptability across various methods for extracting semantic and acoustic tokens from raw wavform including SpeechTokenizer[28], Descript-Audio-Codec[30] and Hubert[31, 32]. Lastly, through ablation studies, we demonstrate the effectiveness of multi-task training in enhancing speech synthesis performance.

In a nutshell, our contributions are as below:

- We significantly improve speech synthesis performance through multi-task joint training, enhancing intelligibility, voice preservation and audio quality, while ensuring fast execution with non-autoregressive methods. Our framework outperforms the state-of-the-art method in both voice preservation and audio quality.
- We propose a unified multi-task framework that handles speech synthesis, editing, and continuation in a single model, achieving high efficiency and versatility in audio processing tasks.
- We demonstrate the adaptability and robustness of our approach by showing its effectiveness across different semantic and acoustic token extraction methods, highlighting its broad applicability and potential for real-world use.

2. Proposed Methods

In this section, we detail the architecture and functionality of SpeechSEC, illustrated in Figure 1.

[1] `https://speechsec-2025.github.io/`

[*] Yuexian Zou is the corresponding author.

10.21437/Interspeech.2025-1776

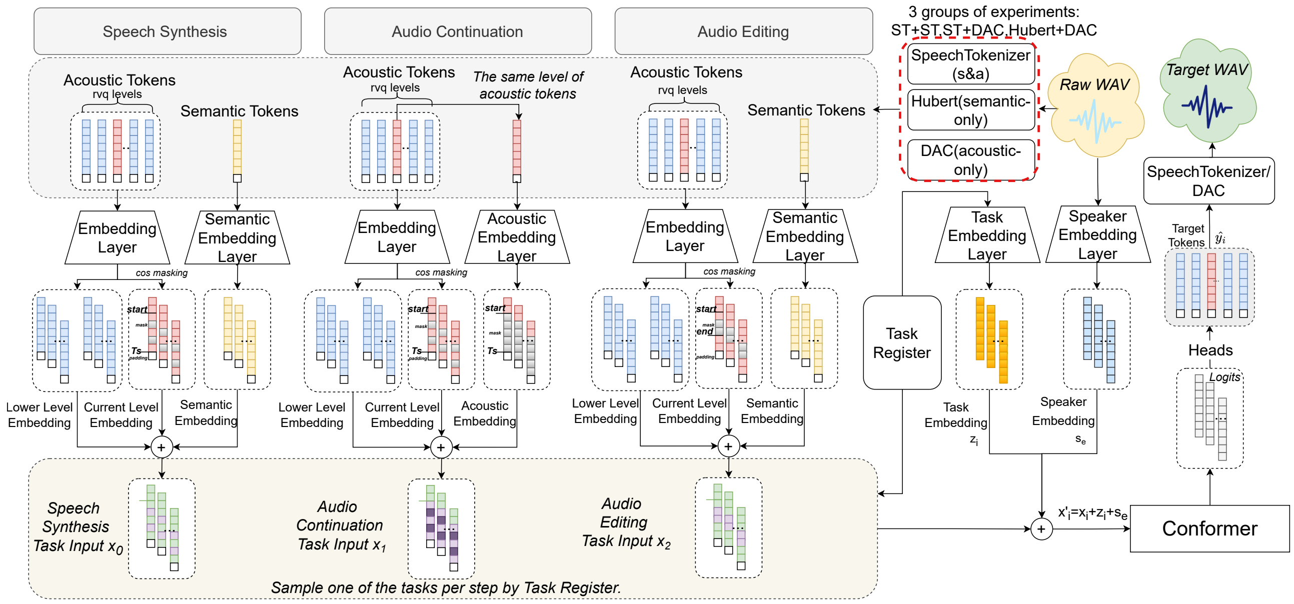

Figure 1: *Illustration of the proposed SpeechSEC training Framework. We sample one of the tasks at each training step and build its condition inputs x_i by padding the raw audio and processing differently, where **red**,blue and yellow denotes valid tokens and white is padding.*

2.1. Problem Definition and Framework Formulation

In SpeechSEC's training process, we adopt a multi-task learning framework that handles three tasks: speech synthesis, speech editing, and speech continuation. Each instance in the dataset is represented as $\mathcal{D} = \{(x_i, y_i, z_i, s_e)\}_{i=1}^{N}$, where:

- x_i: the input sequence, which includes both semantic and acoustic tokens extracted from the raw audio,
- y_i: the target sequence representing the quantized audio tokens,
- z_i: the Task Register that indicates the specific task (e.g., speech synthesis, continuation, or editing),
- s_e: the speaker embedding, capturing the characteristics of the speaker for each speech segment.

The primary objective of SpeechSEC is to predict the target quantized audio tokens y_i conditioned on the input sequence x_i, task information z_i, and speaker embedding s_e. This can be formulated as:

$$P(y_i|x_i, z_i, s_e; \theta), \tag{1}$$

where θ represents the model parameters that are optimized during training. During inference, the semantic and acoustic tokens of x_i are obtained differently depending on the task. For synthesis and editing tasks, semantic tokens can be derived from text using T5 pre-trained model [25, 27]. For continuation and editing tasks, acoustic tokens are extracted from the original audio, providing the necessary context. This flexible input processing enables SpeechSEC to efficiently adapt to various tasks.

2.2. Model Architecture

2.2.1. Semantic and Acoustic Tokens Extractor

During training, we utilize extractors to obtain semantic and acoustic tokens from the audio. Three different methods used in this study are described as below:

- **SpeechTokenizer** is formulated as:

$$\text{ST}(\text{wav}) = \{q_1, \ldots, q_8\}, \quad q_i \in \{1, \ldots, C\}^T, \tag{2}$$

with $C = 1,024$ representing the codebook size. The first layer q_1 represents the semantic tokens, while subsequent layers $q_2, \ldots, q_8$ represent the acoustic tokens.

- **Hubert** is formulated as:

$$\text{Hubert}(\text{wav}) = \{s_1, \ldots, s_T\}, \quad s_t \in \{1, \ldots, C\}, \tag{3}$$

where $C = 500$ denotes the codebook size and s_t is the semantic token at timestep t . Hubert is only used to extract semantic tokens from wavform.

- **DAC** is formulated as:

$$\text{DAC}(\text{wav}) = \{d_1, \ldots, d_{12}\}, \quad d_i \in \{1, \ldots, C\}^T, \tag{4}$$

with $C = 1,024$ representing the codebook size. All 12 layers represent the acoustic tokens. DAC is only used to extract acoustic tokens from wavform.

Each token extractor operates over the duration D of the audio sampled at rate r_s, and the resulting sequence length is defined as $T = \frac{D \cdot r_s}{r_d}$, where r_d is the downsample rate.

2.2.2. Input Condition Embedding Processing

To handle the three tasks of speech synthesis, speech editing, and speech continuation, we process the input conditions differently, obtaining the model input x_i specific to each task.

- **Speech synthesis**: We embed acoustic and semantic tokens, then randomly select one RVQ level for masking. A cosine masking strategy is applied by choosing a random start point $start \in (0, T_s)$, where T_s is the shortest acoustic token length in the batch. Masking follows a cosine schedule $\gamma(\cdot)$, with a probability defined as $p = \cos(u)$, where u is uniformly sampled from $[0, \pi/2]$. The embeddings are summed with the lower-level embeddings and semantic tokens to generate x_0, the speech synthesis task input.

- **Speech continuation**: Acoustic tokens are embedded, and the same masking strategy is applied. However, during inference, since the continuation of the speech is entirely unknown, we mask the unknown future tokens $t \in (start, T_s)$. The masked tokens are then summed with the lower-level embeddings to generate x_1, the continuation task input.
- **Speech editing**: We modify specific audio segments by embedding acoustic tokens and applying a similar masking strategy as in synthesis. Instead of masking up to T_s, we introduce an endpoint *end* after *start*, ensuring masking occurs between $start$ and end. By setting $start$ and end, the model can flexibly handle variable-length speech editing. The model learns to unmask these tokens using semantic tokens, generating x_2, the input for the editing task.

2.2.3. Task Register and Speaker Embedding

To enhance the model's ability to discern inputs for different tasks during joint multi-task learning and more accurately capture the characteristics of the speaker, we implemented a Task Register and integrated speaker embedding. At each training step, we randomly select one task from the available tasks. The index of this task was embedded as z_i. Then we extract speaker embeddings s_e for each speech segment based on the methods described in [33, 34]. The final input to the model x_i' is:

$$x_i' = x_i + z_i + s_e \tag{5}$$

This approach allows the model to learn a generalized representation across tasks while capturing unique speaker characteristics.

2.2.4. Conformer and Decoding Procedure

Our model employs a standard Conformer network with bidirectional self-attention [35] and rotary positional embeddings [36].

The decoding procedure follows SoundStorm's iterative process through RVQ layers, advancing to the next layer $q + 1$ only after selecting tokens from all previous layers $1, \ldots, q$. At each stage, we use a confidence-based sampling approach [23], which reduces forward pass operations compared to autoregressive models like MQTTS [37].

2.3. Training Objective

For each training instance, given the task-specific mask $\mathcal{M}_{rvq}$ identifying RVQ layer-selected tokens for prediction, the training objective for our model becomes:

$$\mathcal{L}(x_i', y_i; \mathcal{M}_{\text{rvq}}) = \sum_{j \in \mathcal{M}_{\text{rvq}}} \left[-\sum_{c=1}^{C} y_{ij}^{(c)} \log(\hat{y}_{ij}^{(c)}) \right] \tag{6}$$

where $\mathcal{L}$ denotes the cross-entropy loss, $y_{ij}^{(c)}$ represents the target probability of class c in the sequence y_i at the j^{th} token position selected by the mask $\mathcal{M}_{\text{rvq}}$, and $\hat{y}_{ij}^{(c)} = M(x_i')_j^{(c)}$ is the predicted probability for the same class and token position by the model M. This formulation ensures the model's focus is exclusively on improving predictions for the masked (i.e., to be predicted) token positions, thereby optimizing the model's learning towards accurate generation and helping the model learn how to unmask the masked tokens.

3. Experiments and Results

3.1. Datasets, Metrics and Implementation Details

Datasets We train SpeechSEC on a dataset designed for text-to-speech (TTS) use: LibriTTS-R dataset [38]. LibriTTS-R is an open, high-quality dataset commonly used for text-to-speech, which consists of 585 hours of speech data at 24 kHz sampling rate from 2,456 speakers and the corresponding texts. During training, we utilize the *train_clean_100*, *train_clean_360*, and *train_clean_500* subset of it and utilize the *test_clean* subset during testing.

Metrics Following [23],we assess synthesized speech in terms of speech intelligibility,voice preservation,audio quality, and synthesis time. For intelligibility, we use the Conformer XL model [35] , evaluating via ASR results by comparing synthesized speech to the original text to obtain WER and CER metrics, where lower scores indicate higher intelligibility. For audio quality, we adopt a MOS evaluation method following the approach used in [23], with higher scores indicating better quality. Voice preservation is assessed through audio cosine similarity between our model's output and the prompt speaker's voice[39, 40], with scores between -1 to 1. Scores closer to 1 on the cosine similarity indicate better preservation of speaker identity. Finally, the average time to generate acoustic tokens from semantic tokens represents synthesis time, conducted on an RTX 3090 GPU.

Implementation Details During training, we preprocess the data by filtering out audio files shorter than one second and resampling to 16kHz. Audio files are sorted by length and batched accordingly (batch size: 64, hidden size: 512). In testing, we extract semantic and acoustic tokens from all audio files in *test_clean*.

For speech synthesis, audio is generated from semantic tokens. In the speech editing task, portions of the acoustic tokens are randomly masked, and the corresponding semantic tokens are provided to generate the modified audio, simulating real-world speech editing scenarios. Our framework supports flexible, variable-length editing, allowing for the addition, removal, or replacement of parts of the audio through adaptive length adjustments, leveraging a pre-trained text-to-semantic model (T5), as demonstrated in the demo. In the speech continuation task, the model is given the first portion of acoustic tokens, and it predicts the subsequent tokens based on the provided context. Since the continuation is generated from prior audio context, WER and CER metrics are not applicable due to the variability in the generated content.

3.2. Main Experiment and Results

To validate the effectiveness of our proposed multi-task learning framework across various semantic and acoustic extractors, experiments are conducted in three groups: *ST*, utilizing SpeechTokenizer for both semantic and acoustic token extraction; *STDAC*, utilizing SpeechTokenizer for semantic and DAC for acoustic tokens; and *HuDAC*, utilizing HuBERT for semantic and DAC for acoustic tokens.

As a baseline, we train each group on the speech synthesis task independently. The results, as illustrated in Table 1, demonstrate that SpeechSEC significantly improves speech synthesis performance across all setups. Improvements are observed in speech intelligibility, audio quality, and voice preservation, while maintaining efficient inference time.

Our results not only remain highly competitive with prior works, such as SoundStorm[23] , but also significantly sur-

Table 1: *Experiments Results for Speech Synthesis Task*

	WER↓	**CER↓**	**Audio Quality↑**	**Voice Preservation↑**	**Acoustic Inference Time↓**
ST (BaseLine)	9.2	3.8	3.98	0.68	**0.19**
ST (SpeechSEC)	**8.7**	**3.6**	**4.20**	**0.72**	0.23
STDAC (BaseLine)	17.9	10.3	3.63	0.57	**0.89**
STDAC (SpeechSEC)	**14.3**	**6.6**	**3.65**	**0.61**	1.21
HuDAC (BaseLine)	14.1	9.4	3.72	**0.59**	**0.86**
HuDAC (SpeechSEC)	**11.3**	**5.7**	**3.83**	**0.58**	1.36

Table 2: *Experiments Results for Speech Editing and Speech Continuation Tasks*

Task Type	**Model**	**WER↓**	**CER↓**	**Audio Quality↑**	**Voice Preservation↑**	**Acoustic Inference Time↓**
Speech Editing	ST	**5.1**	**1.8**	**3.93**	**0.82**	**0.35**
	STDAC	8.3	4.6	3.70	0.64	1.30
	HuDAC	6.6	3.2	3.83	0.58	1.41
Speech Continuation	ST	/	/	**3.63**	**0.66**	**0.43**
	STDAC	/	/	3.50	0.59	0.87
	HuDAC	/	/	3.56	0.58	1.03

Table 3: *Ablation Study Results with Performance Delta Compared with SpeechSEC*

	WER↓	**CER↓**	**Audio Quality↑**	**Voice Preservation↑**
ST(SpeechSEC)	**8.7**	**3.6**	**4.20**	**0.72**
ST (w/o Edit)	$9.7_{+1.0}$	$4.0_{+0.4}$	$3.86_{-0.16}$	$0.70_{-0.02}$
ST (w/o Con)	$9.1_{+0.4}$	$3.8_{+0.2}$	$3.84_{-0.18}$	$0.70_{-0.02}$
STDAC(SpeechSEC)	**14.3**	**6.6**	**3.65**	**0.61**
STDAC (w/o Edit)	$18.7_{+4.4}$	$11.1_{+4.5}$	3.65_{equal}	$0.62_{+0.01}$
STDAC (w/o Con)	$14.8_{+0.5}$	$7.4_{+0.8}$	$3.63_{-0.02}$	$0.58_{-0.03}$
HuDAC(SpeechSEC)	**11.3**	**5.7**	**3.83**	**0.58**
HuDAC(w/o Edit)	$15.3_{+4.0}$	$10.8_{+5.1}$	$3.73_{-0.1}$	0.58_{equal}
HuDAC(w/o Con)	$12.8_{+1.5}$	$8.4_{+2.7}$	$3.67_{-0.16}$	$0.52_{-0.06}$

pass it in key performance metrics, establishing a new benchmark in the field. Specifically, we outperform SoundStorm in both Audio Quality (4.20 vs 4.00, averaged across short, mid, and long durations in the 'with a speaker prompt' setting) and Voice Preservation (0.72 vs 0.58, similarly averaged), demonstrating superior synthesis fidelity and speaker identity preservation. This remarkable achievement is especially noteworthy considering that our model was trained on a smaller dataset, highlighting the efficiency and effectiveness of our approach. These results position SpeechSEC as a state-of-the-art solution for high-quality, robust speech synthesis, offering a significant advancement over previous models.

Additionally, the model shows strong performance in speech editing and continuation tasks, as detailed in Table 2, demonstrating SpeechSEC's versatility and robustness across different speech processing tasks.

3.3. Ablation Study

The ablation study highlights the impact of multi-task learning on speech synthesis by utilizing shared knowledge from editing and continuation tasks. Removing either task during training reduces model performance, as shown in Table 3, underscoring the importance of multi-task learning.

Performance deltas, marked by subscripts, reflect individual task contributions and the synergistic benefits within the SpeechSEC framework. Continuation tasks improve the model's ability to capture internal acoustic relationships, mainly enhancing audio quality and voice preservation, while editing tasks ensure seamless transitions and natural integration between speech segments, primarily boosting intelligibility. Collectively, these tasks enable SpeechSEC to capture a more comprehensive representation of speech, leading to superior synthesis results.

4. Conclusions

In this work, we introduce SpeechSEC, a cutting-edge multi-task framework designed for speech synthesis, editing, and continuation. Through extensive experiments, we show that SpeechSEC surpasses state-of-the-art methods in key areas such as audio quality and voice preservation for speech synthesis tasks. By leveraging the shared knowledge across tasks, SpeechSEC not only achieves superior performance in speech synthesis but also enables high-quality, efficient speech editing and continuation using non-autoregressive techniques. Our approach demonstrates strong adaptability across different token extraction methods, proving its robustness and broad applicability. Future work will focus on extending SpeechSEC's capabilities to handle additional audio tasks, further enhancing its performance and versatility through advanced non-autoregressive methods and multi-task learning strategies.

5. Acknowledgements

This work is supported by Guangdong Provincial Key Laboratory of Ultra High Definition Immersive Media Technology. (Grant No. 2024B1212010006)

6. References

[1] J. Kim, J. Kong, and J. Son, "Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech," *Cornell University - arXiv,Cornell University - arXiv*, Jun 2021.

[2] J. Kong, J. Park, B. Kim, J. Kim, D. Kong, and S. Kim, "Vits2: Improving quality and efficiency of single-stage text-to-speech with adversarial learning and architecture design," Jul 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," Jan 2023.

[4] G. Liu, Y. Zhang, Y. Lei, Y. Chen, R. Wang, Z. Li, and L. Xie, "Promptstyle: Controllable style transfer for text-to-speech with natural language descriptions."

[5] S. Mehta, R. Tu, J. Beskow, Székely, and G. Henter, "Matcha-tts: A fast tts architecture with conditional flow matching."

[6] Y. Guo, C. Du, Z. Ma, X. Chen, and K. Yu, "Voiceflow: Efficient text-to-speech with rectified flow matching," Sep 2023.

[7] Y. Lee, I. Yeon, J. Nam, and J. Chung, "Voiceldm: Text-to-speech with environmental context," Sep 2023.

[8] A. Dekel, S. Shechtman, R. Fernandez, D. Haws, Z. Kons, and R. Hoory, "Speak while you think: Streaming speech synthesis during text generation," Sep 2023.

[9] Y. Xie, Z. Zhu, X. Zhuang, L. Liang, Z. Wang, and Y. Zou, "Gpa: global and prototype alignment for audio-text retrieval," in *Proc. Interspeech 2024*, 2024, pp. 5078–5082.

[10] D. Yang, J. Tian, X. Tan, R. Huang, S. Liu, X. Chang, J. Shi, S. Zhao, J. Bian, X. Wu, Z. Zhao, and H. Meng, "Uniaudio: An audio foundation model toward universal audio generation," Oct 2023.

[11] D. Tan, L. Deng, Y. Yeung, X. Jiang, X. Chen, and T. Lee, "Editspeech: A text based speech editing system using partial inference and bidirectional fusion," *Cornell University - arXiv,Cornell University - arXiv*, Jul 2021.

[12] M. Morrison, L. Rencker, Z. Jin, N. Bryan, J.-P. Caceres, and B. Pardo, "Context-aware prosody correction for text-based speech editing," *Cornell University - arXiv,Cornell University - arXiv*, Feb 2021.

[13] P. Peng, P.-Y. Huang, D. Li, A. Mohamed, and D. Harwath, "Voicecraft: Zero-shot speech editing and text-to-speech in the wild."

[14] T. Wang, J. Yi, R. Fu, J. Tao, and Z. Wen, "Campnet: Context-aware mask prediction for end-to-end text-based speech editing."

[15] M. Morrison, C. Churchwell, N. Pruyne, and B. Pardo, "Fine-grained and interpretable neural speech editing," 2024. [Online]. Available: https://arxiv.org/abs/2407.05471

[16] H. Bai, R. Zheng, J. Chen, X. Li, M. Ma, and L. Huang, "A^{3}t: Alignment-aware acoustic and text pretraining for speech synthesis and editing."

[17] Z. Borsos, R. Marinier, D. Vincent, E. Kharitonov, O. Pietquin, M. Sharifi, O. Teboul, D. Grangier, M. Tagliasacchi, N. Zeghidour, and G. Research, "Audiolm: a language modeling approach to audio generation."

[18] H. Wu, K.-W. Chang, Y.-K. Wu, and H.-y. Lee, "Speechgen: Unlocking the generative power of speech language models with prompts," Jun 2023.

[19] 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," Sep 2023.

[20] E. Kharitonov, A. Lee, A. Polyak, Y. Adi, J. Copet, K. Lakhotia, T.-A. Nguyen, M. Rivière, A. Mohamed, E. Dupoux, and W.-N. Hsu, "Text-free prosody-aware generative spoken language modeling," 2022. [Online]. Available: https://arxiv.org/abs/2109.03264

[21] E. Nachmani, A. Levkovitch, R. Hirsch, J. Salazar, C. Asawaroengchai, S. Mariooryad, E. Rivlin, R. Skerry-Ryan, and M. Ramanovich, "Spoken question answering and speech continuation using spectrogram-powered llm," Oct 2023.

[22] N. Zeghidour, A. Luebs, A. Omran, J. Skoglund, and M. Tagliasacchi, "Soundstream: An end-to-end neural audio codec," *arXiv: Sound,arXiv: Sound*, Jul 2021.

[23] Z. Borsos, M. Sharifi, D. Vincent, E. Kharitonov, N. Zeghidour, and M. Tagliasacchi, "Soundstorm: Efficient parallel audio generation," May 2023.

[24] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *Neural Information Processing Systems,Neural Information Processing Systems*, Jun 2017.

[25] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *arXiv: Learning,arXiv: Learning*, Oct 2019.

[26] X. Zhuang, Y. Xie, Y. Deng, D. Yang, L. Liang, J. Ru, Y. Yin, and Y. Zou, "Vargpt-v1. 1: Improve visual autoregressive large unified model via iterative instruction tuning and reinforcement learning," *arXiv preprint arXiv:2504.02949*, 2025.

[27] P. Budzianowski, T. Sereda, T. Cichy, and I. Vuli'c, "Pheme: Efficient and conversational speech generation," Jan 2024.

[28] X. Zhang, D. Zhang, S. Li, Y. Zhou, X. Qiu, and H. W2vbert, "Speechtokenizer: Unified speech tokenizer for speech large language models."

[29] L. Yu, Y. Cheng, K. Sohn, J. Lezama, H. Zhang, H. Chang, A. G. Hauptmann, M.-H. Yang, Y. Hao, I. Essa, and L. Jiang, "Magvit: Masked generative video transformer," Dec 2022.

[30] R. Kumar, P. Seetharaman, A. Luebs, I. Descript, and I. Kumar, "High-fidelity audio compression with improved rvqgan."

[31] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, 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*, p. 3451–3460, Jan 2021. [Online]. Available: http://dx.doi.org/10.1109/taslp.2021.3122291

[32] M. Ott, S. Edunov, A. Baevski, A. Fan, S. Gross, N. Ng, D. Grangier, and M. Auli, "fairseq: A fast, extensible toolkit for sequence modeling," in *Proceedings of NAACL-HLT 2019: Demonstrations*, 2019.

[33] A. Plaquet and H. Bredin, "Powerset multi-class cross entropy loss for neural speaker diarization," in *Proc. INTERSPEECH 2023*, 2023.

[34] H. Bredin, "pyannote.audio 2.1 speaker diarization pipeline: principle, benchmark, and recipe," in *Proc. INTERSPEECH 2023*, 2023.

[35] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang, "Conformer: Convolution-augmented transformer for speech recognition," in *Interspeech 2020*, Oct 2020. [Online]. Available: http://dx.doi.org/10.21437/interspeech.2020-3015

[36] J. Su, Y. Lu, S. Pan, B. Wen, and Y. Liu, "Roformer: Enhanced transformer with rotary position embedding," *Cornell University - arXiv,Cornell University - arXiv*, Apr 2021.

[37] L.-W. Chen, S. Watanabe, and A. Rudnicky, "A vector quantized approach for text to speech synthesis on real-world spontaneous speech," Feb 2023.

[38] Y. Koizumi, H. Zen, S. Karita, Y. Ding, K. Yatabe, N. Morioka, M. Bacchiani, Y. Zhang, W. Han, A. Bapna, G. Google, and J. Japan, "Libritts-r: A restored multi-speaker text-to-speech corpus."

[39] S. Chen, Y. Wu, C. Wang, Z. Chen, Z. Chen, S. Liu, J. Wu, Y. Qian, F. Wei, J. Li, and X. Yu, "Unispeech-sat: Universal speech representation learning with speaker aware pre-training," 2021.

[40] C. Wang, Y. Wu, Y. Qian, K. Kumatani, S. Liu, F. Wei, M. Zeng, and X. Huang, "Unispeech: Unified speech representation learning with labeled and unlabeled data," in *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, ser. Proceedings of Machine Learning Research, M. Meila and T. Zhang, Eds., vol. 139. PMLR, 2021, pp. 10 937–10 947. [Online]. Available: http://proceedings.mlr.press/v139/wang21y.html