StarVC: A Unified Auto-Regressive Framework for Joint Text and Speech Generation in Voice Conversion

Fengjin Li[1], *Jie Wang*[2], *Yadong Niu*[2], *Yongqing Wang*[2], *Meng Meng*[2], *Jian Luan*[2], *Zhiyong Wu*[1,†]

[1]Shenzhen International Graduate School, Tsinghua University, China
[2]MiLM Plus, Xiaomi Inc., China

lifj23@mails.tsinghua.edu.cn, zywu@sz.tsinghua.edu.cn

Abstract

Voice Conversion (VC) modifies speech to match a target speaker while preserving linguistic content. Traditional methods usually extract speaker information directly from speech while neglecting the explicit utilization of linguistic content. Since VC fundamentally involves disentangling speaker identity from linguistic content, leveraging structured semantic features could enhance conversion performance. However, previous attempts to incorporate semantic features into VC have shown limited effectiveness, motivating the integration of explicit text modeling. We propose **StarVC**, a unified autoregressive VC framework that first predicts text tokens before synthesizing acoustic features. The experiments demonstrate that StarVC outperforms conventional VC methods in preserving both linguistic content (i.e., WER and CER) and speaker characteristics (i.e., SECS and MOS). Audio demo can be found at: `https://thuhcsi.github.io/StarVC/`.

Index Terms: voice conversion, speech synthesis

1. Introduction

Speaker identity is one of the fundamental characteristics of human speech, influencing communication, personalization and speech perception. Voice Conversion (VC) is a technique that modifies the speaker identity of an utterance while preserving its linguistic content. Traditional VC methods [1, 2] often attempt to extract and manipulate speaker identity directly from speech signals, yet this approach is inherently challenging due to the strong coupling between speaker characteristics and linguistic content [3]. Disentangling these factors effectively remains an open problem, as naive transformations often lead to timbre leakage, unnatural prosody or a loss of intelligibility.

Compared to speaker-related features, extracting semantic information from speech is a more mature and well-established field, with robust methodologies [4, 5, 6, 7] available. This raises an intriguing question: if a VC system can effectively incorporate semantic representation into its framework, could it help to better disentangle speaker identity from linguistic content? Rather than directly extracting and modifying speaker characteristics, leveraging structured semantic representations might enable more effective speaker adaptation while maintaining intelligibility and naturalness. Some recent VC methods [8, 9] have attempted to integrate semantic information into their frameworks, but their effectiveness remains limited. Many of these approaches rely on latent text-based features rather than explicitly generating textual content, which may result in weaker constraints on linguistic consistency and speaker preservation.

Recent advancements in VC can be categorized into two main directions: diffusion-based methods and large language model (LLM)-based methods. Diffusion-based VC models, such as Diff-VC [10] and StableVC [11], generate high-quality speech by directly modeling audio signals, which allows for high-fidelity conversion while benefiting from recent optimizations that enable faster inference. These models now generate speech in a single step, significantly improving efficiency. On the other hand, LLM-driven approaches, including OpenVoice V2 [12], LM-VC [13], and DualVC3 [14], operate on acoustic tokens rather than raw audio, which may lead to some information loss but provides advantages in handling long-form speech. Despite these advancements, there remains a gap in integrating semantic information effectively into VC frameworks, highlighting the need for a more structured approach.

Furthermore, conversation-oriented models such as Moshi [15], Mini-Omni [16], and Llama-Omni [17] have demonstrated the potential of integrating text generation with speech transformation. These models highlight how explicitly modeling linguistic content alongside speech synthesis can improve coherence and intelligibility. Inspired by this, we propose **StarVC** (**S**peech-**T**ext **A**uto-**R**egressive **V**oice **C**onversion), a novel VC framework that unifies speech conversion and text generation within one auto-regressive structure. Unlike conventional approaches that either rely on ASR-based transcriptions or operate in a purely textless manner, StarVC introduces an intermediate step where text tokens are predicted before generating acoustic features. By grounding speech synthesis in a structured semantic representation, this approach may provide stronger linguistic consistency while preserving the speaker's unique characteristics.

StarVC is designed with a multi-stage training strategy. Initially, the model undergoes ASR pretraining, which may refine its ability to extract and encode semantic information. The VC training phase then optimizes the transformation of speaker identity while preserving intelligibility, leveraging both semantic and speaker representations. Finally, a multi-task learning framework is introduced to jointly optimize ASR and VC objectives, which may encourage better disentanglement of speaker and linguistic content. Additionally, data augmentation techniques, incorporating both real and synthesized utterances, may help improve generalization and robustness in diverse linguistic and speaker conditions.

By bridging the gap between diffusion-based, LLM-based, and traditional VC methodologies, StarVC has the potential to enable high-quality voice conversion with synchronized text outputs. If effective, this approach may enhance applications such as personalized speech synthesis, interactive voice assistants, and content creation, offering a more unified and efficient solution to the challenges of VC research.

† Corresponding author.

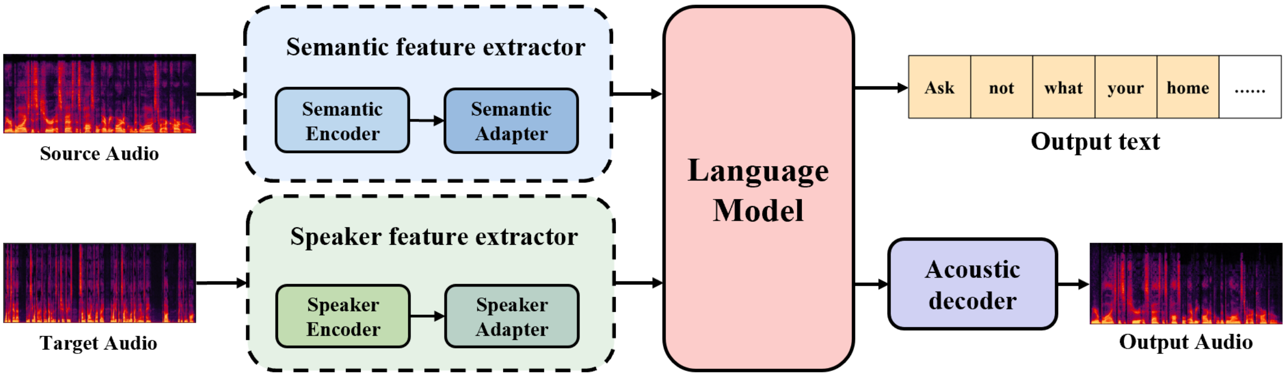

Figure 1: *Architecture of StarVC*

2. Proposed Approach

2.1. System Architecture

Our proposed StarVC framework is designed to jointly model speech conversion and text generation in an auto-regressive manner. As shown in Figure 1, the system consists of four main components: Semantic Feature Extractor, Speaker Feature Extractor, Language Model and Acoustic Decoder.

2.1.1. Feature Extractors

Semantic Feature Extractor We employ the encoder of Whisper-small[1] from OpenAI as the Semantic Encoder to extract high-level semantic features from the source speech. Whisper [18] is a widely used ASR model trained on diverse speech data, making it well-suited for capturing linguistic content in voice conversion. As shown in Figure 2, the source audio is first converted into a mel spectrogram $M_s \in \mathbb{R}^{T_m \times F_m}$, where T_m is the number of frames and F_m is the number of mel bins. It is then passed through the Semantic Encoder and Semantic adapter to obtain semantic features $S = \{s_1, s_2, \ldots, s_T\}$, $s_i \in \mathbb{R}^{dim}$, T is the length of the semantic features and dim is the model's embedding dimension.

Speaker Feature Extractor We employ ERes2Net-large[2] as the Speaker Encoder to extract speaker-specific features from the target speech. ERes2Net [19] is designed for speaker embedding extraction, incorporating multi-scale feature fusion for improved robustness. As shown in Figure 2, we extract FBank features $B_t \in \mathbb{R}^{T_f \times F_f}$, where T_f is the number of frames and F_f is the number of FBank bins. It is then processed through the Speaker Encoder and Speaker adapter to obtain the speaker feature $S' = s'_1$, where $s'_1 \in \mathbb{R}^{dim}$.

2.1.2. Auto-regressive Language Model

We apply delay method adopted in MusicGen [20], which effectively conditions acoustic tokens on prior text tokens. All text tokens are generated one step ahead of the first corresponding acoustic token, helping acoustic tokens to be conditioned more explicitly on the generated text. In addition, we also employed the teacher forcing strategy [21] and the RoPE [22] encoding strategy to optimize the training effect of the model.

The extracted semantic features S and speaker feature S' are concatenated and input into the LM (Language Model) to generate the output sequence:

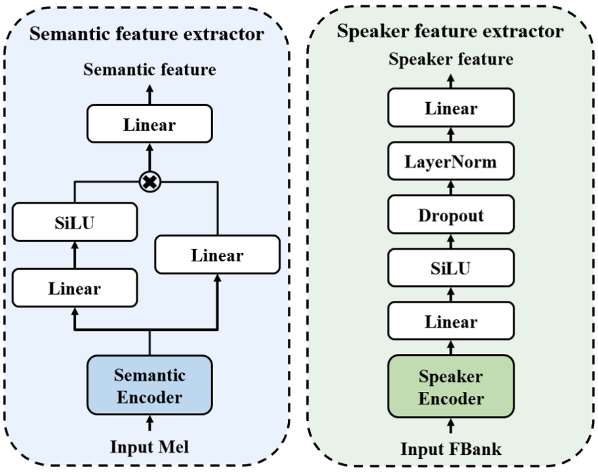

Figure 2: *Architecture of Feature Extractors*

$$Y = LM(S, S') = \{\hat{y}_t, \hat{y}_{a_1}, \ldots, \hat{y}_{a_n}\}, \hat{y}_t \in \mathbb{R}^{V_t}, \hat{y}_{a_i} \in \mathbb{R}^{V_a}$$

where n is the number of the codebook layers, V_t and V_a represent the vocabulary sizes for text and acoustic tokens respectively.

2.1.3. Acoustic Decoder

For waveform reconstruction, we utilize Mimi[3], an open-source audio codec optimized for both compression efficiency and reconstruction fidelity. While Mimi allows flexible control over the number of output code layers, we follow the same setting as Moshi [15] and select eight layers. The first layer is distilled from WavLM [4] to retain critical semantic information, while the remaining layers encode fine-grained acoustic details. This multi-layer structure, combined with the auto-regressive text-acoustic generation strategy, significantly enhances the naturalness of synthesized speech. Compared to other codecs such as Encodec [23], SNAC [24], DAC [25] and SemantiCodec [26], Mimi demonstrates superior performance in both speech quality and speaker similarity, making it more effective for voice conversion.

[1] `https://github.com/openai/whisper`

[2] `https://github.com/modelscope/3D-Speaker/tree/main`

[3] `https://github.com/kyutai-labs/moshi`

2.2. Multi-Stage Training Strategy

To effectively optimize our model for both text generation and voice conversion, we adopt a three-stage training strategy. Throughout all training stages, the Semantic Encoder, Speaker Encoder and the Acoustic Decoder remain frozen, ensuring that their pre-trained representations are preserved while minimizing unnecessary parameter updates.

2.2.1. *ASR Pretraining*

In the first stage, the model is trained as an ASR system, updating the Semantic adapter and the LM while keeping all other components frozen. This phase enhances the model's ability to extract robust semantic representations, which serve as a strong foundation for improving linguistic accuracy in the subsequent voice conversion task. The ASR loss is computed as:

$$\mathcal{L}_{\text{ASR}} = \text{CE}(y_t, \hat{y}_t)$$

where y_t is the ground-truth transcription, CE is cross entropy.

2.2.2. *VC Training*

During this stage, we update both the Semantic adapter and the Speaker adapter, along with the LM, to incorporate speaker characteristics while preserving linguistic accuracy. The VC loss is a weighted combination of text and acoustic losses:

$$\mathcal{L}_{\text{VC}} = w\text{CE}(y_t, \hat{y}_t) + (1-w)\sum_{i=1}^{n} \lambda_i \text{CE}(y_{a_i}, \hat{y}_{a_i})$$

Among them, w is the weight for adjusting the balance between the text loss and the acoustic loss, and λ_i is the weight for adjusting the loss of acoustic tokens at different layers.

2.2.3. *Joint ASR-VC Training*

In the final stage, we jointly train the model for ASR and VC tasks, allocating 20% of training instances to ASR tasks and 80% to VC tasks. Since ASR tasks do not require speaker adaptation, updates are limited to the Semantic adapter and LM. The final loss function integrates ASR and VC losses:

$$\mathcal{L} = w'\mathcal{L}_{\text{ASR}} + (1-w')\mathcal{L}_{\text{VC}}$$

By jointly optimizing ASR and VC objectives, we enable the model to balance linguistic accuracy and speaker identity retention, ensuring high-quality voice conversion with minimal transcription errors.

2.3. Parallel Data Augmentation

To enhance speaker diversity, we use OpenVoice V2[4] to synthesize multiple variations of the same utterance with different speakers. During training, we randomly select real or synthesized speech as the target audio, reducing the model's reliance on strictly parallel real-world recordings and enhancing its robustness.

3. Experiments

3.1. Training

We train our model on 8 NVIDIA H100 80G GPUs using three sequential phases: ASR Pretraining (30 hours), VC Training (50 hours) and Joint ASR-VC Training (100 hours). Our LM is a 24-layer transformer [27] (intermediate size=4,864, embedding dim=896 and 14 heads), following the same architecture as Qwen2.5-0.5B[5]. During VC Training, we adopt a 50%/50% probability of selecting real or OpenVoice V2-synthesized parallel data. In order to allow the model to synthesize more real speech, the ratio was adjusted to 80%/20% in the Joint ASR-VC Training. Acoustic token weights are set to [1.0, 1.0, 0.9, 0.9, 0.8, 0.8, 0.7, 0.7] across eight codebook layers. The batch size is set to 6, and we utilize Qwen's [28, 29] tokenizer for text tokenization, with a text vocabulary size of 151,936 and an audio vocabulary size of 2,048.

3.1.1. *Datasets*

We use two English speech datasets for training: GigaSpeech [30] (10,000 hours) and LibriTTS [31, 32] (960 hours) . GigaSpeech covers diverse domains and acoustic conditions, while LibriTTS is cleaner and more consistent. By combining them, we enable StarVC to handle both real-world variability and well-segmented, high-quality speech.

3.2. Evaluation

We select 400 utterances from the LibriTTS test-clean set, partitioning them into 200 source and 200 target utterances. None of these samples appear in the training process, ensuring a fair evaluation.

Baseline Methods. We compare our proposed StarVC against three existing methods:

- *CosyVoice* [33, 34]: A diffusion-based speech generation approach from Alibaba, featuring a VC variant.
- *OpenVoice V2* [12]: A large-scale, zero-shot VC model built upon YourTTS [35].
- *TriAAN-VC* [36]: A deep-learning-based framework for any-to-any VC, focusing on disentangling linguistic content and target speaker characteristics.

Evaluation Metrics. We adopt the following metrics to gauge performance:

- *SECS (Speaker Embedding Cosine Similarity)*: Quantifies timbre similarity by comparing embeddings (extracted via Resemblyzer[6] and a fine-tuned WavLM[7]) from the converted speech to those of the target.
- *WER (Word Error Rate)*: Assesses how accurately the linguistic content is preserved by transcribing the converted audio (using Whisper-large [18]) and comparing it to the ground-truth text. We also report WER-Text, reflecting the accuracy of StarVC's generated text tokens.
- *CER (Character Error Rate)*: Evaluates character-level transcription accuracy. We also track CER-Text to measure how closely the model's generated text tokens match the reference at the character level. A lower CER signifies fewer insertions, deletions, or substitutions, indicating better fine-grained textual fidelity.
- *MOS (Mean Opinion Score)*: Subjective listening tests on 20 randomly selected source-target pairs, participated by 20 listeners. We report naturalness MOS (NMOS) and speaker similarity MOS (SMOS).

[4] `https://github.com/myshell-ai/OpenVoice`

[5] `https://github.com/QwenLM/Qwen2.5`

[6] `https://github.com/resemble-ai/Resemblyzer`

[7] `https://github.com/BytedanceSpeech/seed-tts-eval`

Table 1: *Objective Evaluation of StarVC and Baselines (including Ablations).* ***Note:*** *SECS-Res and SECS-Wavlm are SECS metrics from Resemblyzer and a fine-tuned WavLM [4], respectively. WER-Text and CER-Text evaluate the text generated by StarVC at word-level and character-level accuracy. Smaller model's LM is a 12-layer Transformer (intermediate size=2,240, embedding dim=448).* ***Bold*** *values indicate the best results, and underlined values indicate the second-best results.*

Model	SECS-Res ↑	SECS-Wavlm ↑	WER ↓	WER-Text ↓	CER ↓	CER-Text ↓
CosyVoice	**0.839**	**0.478**	8.24%	/	4.27%	/
OpenVoice V2	0.771	0.284	8.17%	/	4.15%	/
TriAAN-VC	0.756	0.241	19.67%	/	12.18%	/
StarVC	0.835	0.472	**6.27%**	**4.95%**	**4.09%**	**1.51%**
– w/o multi-stage	0.812	0.429	7.24%	5.09%	4.60%	1.61%
– w/o text token	0.771	0.382	7.30%	/	4.31%	/
– smaller model	0.750	0.383	8.04%	5.33%	5.67%	1.56%

Table 2: *MOS Evaluation of StarVC and Baselines with 95% confidence interval.* ***Note: Bold*** *values indicate the best results, and underlined values indicate the second-best results.*

Model	SMOS ↑	NMOS ↑
CosyVoice	3.94 ± 0.09	4.15 ± 0.08
OpenVoice V2	3.97 ± 0.08	4.09 ± 0.08
TriAAN-VC	3.25 ± 0.09	3.06 ± 0.10
StarVC	$\mathbf{3.98 \pm 0.08}$	$\mathbf{4.17 \pm 0.08}$

3.3. Results and Analysis

3.3.1. Overall Objective Metrics

Table 1 presents comprehensive objective evaluation results for StarVC and baselines, including ablation studies. SECS-Res and SECS-WavLM quantify speaker similarity, while WER and CER measure linguistic content preservation. WER-Text and CER-Text assess StarVC's generated text accuracy.

Analysis reveals distinct performance patterns. Regarding speaker similarity, CosyVoice[8] attained marginally higher SECS-Res and SECS-WavLM scores than StarVC, indicating a slight edge via embedding metrics. Crucially, StarVC demonstrated robust timbre adaptation, closely approaching CosyVoice's performance. TriAAN-VC[9] exhibited significantly lower scores, suggesting weaker speaker retention.

StarVC achieved clear superiority in linguistic fidelity. It attained the lowest WER at 6.27%, substantially outperforming CosyVoice's 8.24% and OpenVoice V2's 8.17%. Simultaneously, StarVC achieved the lowest CER of 4.09%, surpassing CosyVoice's 4.27% and OpenVoice V2's 4.15%. This dual lead across word and character levels underscores StarVC's exceptional content preservation. TriAAN-VC's high WER and CER highlight its content preservation challenges.

A key StarVC differentiator is its exceptionally accurate text token generation, with WER-Text at 4.95% and CER-Text at 1.51%. These results validate the efficacy of its integrated text generation mechanism, a unique capability among baselines.

3.3.2. Subjective Ratings

Subjective MOS evaluations in Table 2 corroborate these findings. StarVC achieved the highest scores in both speaker similarity and naturalness. CosyVoice and OpenVoice V2 delivered comparable performance, while TriAAN-VC lagged significantly.

Notably, StarVC's NMOS lower confidence bound (4.09) exceeds CosyVoice's (4.07), and its SMOS lower bound (3.90) exceeds OpenVoice V2's (3.89), suggesting a perceptible advantage. The high NMOS approaches highly natural speech levels, while the leading SMOS reflects superior perceived speaker matching.

3.3.3. Ablation Study

We examine the impact of multi-stage training, text-token generation, and model size. Without multi-stage training, performance degrades, as shown by WER rising from 6.27% to 7.24% and SECS-Res falling from 0.835 to 0.812. Eliminating text tokens results in a significant decline in speaker similarity and higher WER, though it maintains relatively good CER compared to baselines like OpenVoice V2. However, this ablation forfeits text-level outputs entirely, underscoring that text conditioning is critical for generating explicit textual representations. A smaller model further diminishes performance, revealing the need for sufficient capacity to capture both acoustic details and textual correctness.

Overall, these findings confirm that multi-stage training, text token generation and a sufficiently large model are pivotal for StarVC's success, allowing it to excel in both objective and subjective evaluations while providing accurate text output for each converted utterance.

4. Conclusion

In this work, we introduced StarVC, a multi-stage voice conversion framework that integrates speech generation with text prediction, ensuring stronger linguistic consistency. By leveraging large-scale labeled data (GigaSpeech, LibriTTS) and an autoregressive modeling approach, StarVC enhances the disentanglement of speaker identity and linguistic content. Our evaluation suggests that StarVC achieves competitive speaker similarity, intelligibility and subjective quality while uniquely incorporating text generation. Multi-stage training, text token conditioning and sufficient model capacity contribute to effective timbre transfer and linguistic preservation. Overall, StarVC unifies voice conversion and transcription, making it suitable for applications such as interactive dialogue systems, content creation, and personalized speech services.

[8] https://github.com/FunAudioLLM/CosyVoice

[9] https://github.com/winddori2002/TriAAN-VC

5. Acknowledgements

This work is supported by National Natural Science Foundation of China (62076144) and Shenzhen Science and Technology Program (JCYJ20220818101014030).

6. References

[1] Y. Stylianou, O. Cappé, and E. Moulines, “Continuous probabilistic transform for voice conversion,” *IEEE Transactions on speech and audio processing*, vol. 6, no. 2, pp. 131–142, 1998.

[2] A. R. Bargum, S. Serafin, and C. Erkut, “Reimagining speech: A scoping review of deep learning-powered voice conversion,” *arXiv preprint arXiv:2311.08104*, 2023.

[3] B. Sisman, J. Yamagishi, S. King, and H. Li, “An overview of voice conversion and its challenges: From statistical modeling to deep learning,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 132–157, 2020.

[4] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao *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, pp. 1505–1518, 2022.

[5] 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*, vol. 29, pp. 3451–3460, 2021.

[6] S. Schneider, A. Baevski, R. Collobert, and M. Auli, “wav2vec: Unsupervised pre-training for speech recognition,” *arXiv preprint arXiv:1904.05862*, 2019.

[7] 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.

[8] A. Baade, P. Peng, and D. Harwath, “Neural codec language models for disentangled and textless voice conversion,” in *Proc. Interspeech 2024*, 2024, pp. 182–186.

[9] J. Li, Y. Guo, X. Chen, and K. Yu, “Sef-vc: Speaker embedding free zero-shot voice conversion with cross attention,” in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 12 296–12 300.

[10] V. Popov, I. Vovk, V. Gogoryan, T. Sadekova, M. Kudinov, and J. Wei, “Diffusion-based voice conversion with fast maximum likelihood sampling scheme,” *arXiv preprint arXiv:2109.13821*, 2021.

[11] J. Yao, Y. Yan, Y. Pan, Z. Ning, J. Ye, H. Zhou, and L. Xie, “Stablevc: Style controllable zero-shot voice conversion with conditional flow matching,” *arXiv preprint arXiv:2412.04724*, 2024.

[12] Z. Qin, W. Zhao, X. Yu, and X. Sun, “Openvoice: Versatile instant voice cloning,” *arXiv preprint arXiv:2312.01479*, 2023.

[13] Z. Wang, Y. Chen, L. Xie, Q. Tian, and Y. Wang, “Lm-vc: Zero-shot voice conversion via speech generation based on language models,” *IEEE Signal Processing Letters*, 2023.

[14] Z. Ning, S. Wang, P. Zhu, Z. Wang, J. Yao, L. Xie, and M. Bi, “Dualvc 3: Leveraging language model generated pseudo context for end-to-end low latency streaming voice conversion,” *arXiv preprint arXiv:2406.07846*, 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,” *arXiv preprint arXiv:2410.00037*, 2024.

[16] Z. Xie and C. Wu, “Mini-omni: Language models can hear, talk while thinking in streaming,” *arXiv preprint arXiv:2408.16725*, 2024.

[17] Q. Fang, S. Guo, Y. Zhou, Z. Ma, S. Zhang, and Y. Feng, “Llama-omni: Seamless speech interaction with large language models,” *arXiv preprint arXiv:2409.06666*, 2024.

[18] 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.

[19] Y. Chen, S. Zheng, H. Wang, L. Cheng, Q. Chen, and J. Qi, “An enhanced res2net with local and global feature fusion for speaker verification,” *arXiv preprint arXiv:2305.12838*, 2023.

[20] 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.

[21] R. J. Williams and D. Zipser, “A learning algorithm for continually running fully recurrent neural networks,” *Neural computation*, vol. 1, no. 2, pp. 270–280, 1989.

[22] J. Su, M. Ahmed, Y. Lu, S. Pan, W. Bo, and Y. Liu, “Roformer: Enhanced transformer with rotary position embedding,” *Neurocomputing*, vol. 568, p. 127063, 2024.

[23] A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, “High fidelity neural audio compression,” *arXiv preprint arXiv:2210.13438*, 2022.

[24] H. Siuzdak, F. Grötschla, and L. A. Lanzendörfer, “Snac: Multi-scale neural audio codec,” *arXiv preprint arXiv:2410.14411*, 2024.

[25] R. Kumar, P. Seetharaman, A. Luebs, I. Kumar, and K. Kumar, “High-fidelity audio compression with improved rvqgan,” *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[26] H. Liu, X. Xu, Y. Yuan, M. Wu, W. Wang, and M. D. Plumbley, “Semanticodec: An ultra low bitrate semantic audio codec for general sound,” *arXiv preprint arXiv:2405.00233*, 2024.

[27] A. Vaswani, “Attention is all you need,” *Advances in Neural Information Processing Systems*, 2017.

[28] J. Bai, S. Bai, Y. Chu, Z. Cui, K. Dang, X. Deng, Y. Fan, W. Ge, Y. Han, F. Huang *et al.*, “Qwen technical report,” *arXiv preprint arXiv:2309.16609*, 2023.

[29] A. Yang, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Li, D. Liu, F. Huang, H. Wei *et al.*, “Qwen2. 5 technical report,” *arXiv preprint arXiv:2412.15115*, 2024.

[30] G. Chen, S. Chai, G. Wang, J. Du, W.-Q. Zhang, C. Weng, D. Su, D. Povey, J. Trmal, J. Zhang *et al.*, “Gigaspeech: An evolving, multi-domain asr corpus with 10,000 hours of transcribed audio,” *arXiv preprint arXiv:2106.06909*, 2021.

[31] 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,” *arXiv preprint arXiv:1904.02882*, 2019.

[32] 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)*. IEEE, 2015, pp. 5206–5210.

[33] Z. Du, Q. Chen, S. Zhang, K. Hu, H. Lu, Y. Yang, H. Hu, S. Zheng, Y. Gu, Z. Ma *et al.*, “Cosyvoice: A scalable multilingual zero-shot text-to-speech synthesizer based on supervised semantic tokens,” *arXiv preprint arXiv:2407.05407*, 2024.

[34] Z. Du, Y. Wang, Q. Chen, X. Shi, X. Lv, T. Zhao, Z. Gao, Y. Yang, C. Gao, H. Wang *et al.*, “Cosyvoice 2: Scalable streaming speech synthesis with large language models,” *arXiv preprint arXiv:2412.10117*, 2024.

[35] E. Casanova, J. Weber, C. D. Shulby, A. C. Junior, E. Gölge, and M. A. Ponti, “Yourtts: Towards zero-shot multi-speaker tts and zero-shot voice conversion for everyone,” in *International Conference on Machine Learning*. PMLR, 2022, pp. 2709–2720.

[36] H. J. Park, S. W. Yang, J. S. Kim, W. Shin, and S. W. Han, “Triaan-vc: Triple adaptive attention normalization for any-to-any voice conversion,” in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.