

ProsoCodec: Prosody-Oriented Speech Codec for Voice Conversion

Jeongsoo Choi¹, Ji-Hoon Kim², Shujie Hu³, Joon Son Chung^{1,**}

¹ KAIST, South Korea

² Chung-Ang University, South Korea

³ The Chinese University of Hong Kong, China

jeongsoo.choi@kaist.ac.kr, joonson@kaist.ac.kr

Abstract

Neural speech codecs efficiently compress speech and have become a foundation for speech generation, but they are typically learned as holistic representations that intertwine linguistic content, speaker identity, and prosody. While this design is effective for zero-shot voice cloning, it hinders downstream tasks that require prosody preservation or transfer, such as voice conversion. To address this, we introduce ProsoCodec, a prosody-oriented speech codec that models prosody as a conditional residual rather than as a disentangled stream. Specifically, by conditioning both the encoder and decoder on text and speaker embeddings as prefix tokens, the discrete bottleneck is encouraged to capture prosodic variation not explained by content and speaker. To further preserve prosody, we use the low-frequency mel band and train the model on paired same-speaker utterances. Experiments on voice conversion show improved style preservation and reduced interference between content and timbre.

Index Terms: neural speech codec, voice conversion, prosody modeling

1. Introduction

Recent advancements in zero-shot speech generation and voice cloning have demonstrated remarkable capabilities in producing high-fidelity speech from brief reference prompts [1, 2, 3, 4]. Much of this progress has been driven by neural speech codecs [5, 6], which discretize continuous speech into compact tokens, serving as the foundation for frameworks employing language models for in-context learning [7, 8, 9]. However, to faithfully reconstruct speech, these codecs are typically learned as holistic representations that intertwine linguistic content, speaker identity, and prosody. While effective for zero-shot voice cloning, which aims to replicate the reference entirely, this entanglement makes it difficult to control prosody without altering content or timbre [10, 11].

This issue becomes particularly evident in voice conversion, where the goal is to modify speaker timbre while strictly preserving the source utterance’s linguistic content and prosodic variation [12, 11]. Previous approaches attempt to achieve this by decomposing speech into distinct attributes such as content, timbre, and pitch [13, 14], or by learning disentangled representations through restrictive bottlenecks [15] or adversarial objectives [10] within an autoencoder. Voice conversion is then performed by replacing the speaker representation while keeping the remaining factors fixed. However, prosody is not a standalone attribute that can be cleanly swapped, as it is conditioned on both content and speaker, varying with context even within a single speaker [16]. Consequently, enforcing a fully speaker-independent prosody

representation often discards speaker-conditioned prosodic nuances and limits expressiveness [17].

To overcome this, we propose ProsoCodec, a prosody-oriented speech codec conditioned on text and speaker identity to encode residual prosodic variation. Unlike neural speech codecs that aim to faithfully reconstruct holistic acoustic signals under a limited bitrate [5, 18, 6, 19], our objective is selective representation learning tailored for fine-grained prosodic control. Leveraging the strong performance of recent automatic speech recognition (ASR) [20, 21] and speaker verification (SV) [22, 23] systems, linguistic content and global speaker identity have become highly reliable and easily accessible features. By conditioning both the encoder and decoder on these attributes and introducing a discrete bottleneck through quantization, ProsoCodec encourages the latent representation to focus on residual prosodic variation rather than redundant content or speaker information. This conceptual shift allows the model to achieve effective residual modeling without relying on fragile disentanglement mechanisms such as adversarial objectives.

ProsoCodec builds upon a diffusion autoencoder architecture [24, 25], where an encoder maps speech into discrete tokens and a diffusion decoder reconstructs high-fidelity mel-spectrograms from them. We restrict the input mel-spectrogram to its low-frequency band to bias the tokens toward prosodic variation, inspired by previous works [26, 27]. For voice conversion, the decoder generates speech from the source tokens while conditioning on a reference prompt to adopt the target speaker timbre. However, such prompt-based generation tends to imitate the reference holistically, and can therefore override the source prosody [28]. To mitigate this, we introduce a dual-utterance training strategy using paired utterances from the same speaker, making utterance-level style harder to copy from the prompt, thereby improving prosody preservation. Extensive experiments demonstrate that ProsoCodec outperforms previous zero-shot voice conversion methods in prompt speaker similarity, source content preservation, and source prosody preservation. We further conduct ablation studies to provide insights into how different design components influence voice conversion performance.

Our contributions are three-fold: 1) We propose ProsoCodec, a novel generative speech codec that models prosody as a conditional residual by explicitly conditioning on text and speaker identity, without requiring hand-crafted prosody supervision or complex adversarial disentanglement. 2) We introduce a simple yet effective dual-utterance training strategy to explicitly prevent prompt-style leakage, decoupling utterance-level prosody from global speaker timbre. 3) We present comprehensive ablation studies on input representations and bottleneck designs, offering insights into balancing timbre conversion quality and source-style preservation for high-quality zero-shot voice conversion.

**indicates the corresponding author.

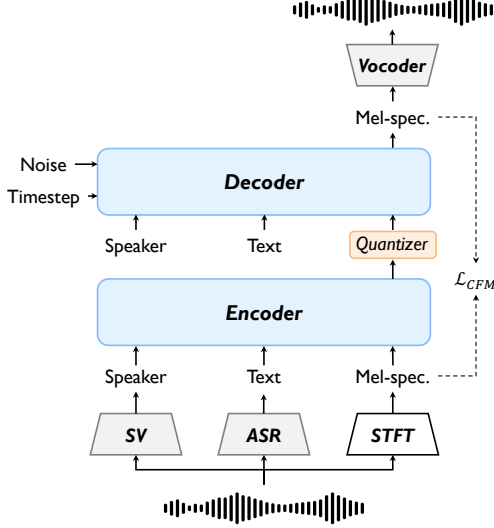


Figure 1: Model architecture of ProsoCodec.

2. Method

2.1. Overview

Given a source utterance and a reference utterance, our voice conversion framework aims to generate speech that preserves the linguistic content and prosodic style of the source while adopting the speaker timbre of the reference. Both waveforms are separately encoded by the encoder of the proposed ProsoCodec and quantized into discrete token sequences. Then the decoder generates timbre-converted speech from the source tokens, conditioned on the reference tokens and reference speech provided as prompts to guide generation toward the target speaker timbre. To encourage the discrete tokens to retain more prosodic style information rather than linguistic and timbre information, we additionally condition ProsoCodec on transcriptions and speaker embeddings obtained from pretrained models. The overall architecture of ProsoCodec is illustrated in Fig. 1.

2.2. ProsoCodec

2.2.1. Explicit Text and Speaker Priors

To encourage the codec to capture residual information beyond linguistic and speaker content, we incorporate explicit text and speaker priors during both encoding and decoding. An ASR model pretrained on large-scale, diverse data provides utterance transcriptions, which are tokenized into subwords, embedded via a lookup table, and processed by an MLP-based text encoder to match the module input space, producing $\mathbf{e}_{\text{txt}} \in \mathbb{R}^{L \times D}$, where L is the token length and D the model dimension. Speaker characteristics are extracted by a pretrained SV model as an utterance-level embedding and projected through an MLP-based speaker encoder to obtain $\mathbf{e}_{\text{spk}} \in \mathbb{R}^{1 \times D}$.

2.2.2. Prior-guided Encoder

The encoder compresses the input speech waveform, conditioned on the explicit priors. Given an input waveform, we extract a mel-spectrogram using the short-time Fourier transform (STFT) and project it to the encoder dimension through a linear layer, resulting in $\mathbf{e}_{\text{mel}} \in \mathbb{R}^{T \times D}$, where T denotes the temporal length. The conditioning signals (\mathbf{e}_{spk} and \mathbf{e}_{txt}) are utilized as prefix tokens

and concatenated with the mel-spectrogram along the temporal dimension: $\mathbf{e} = [\mathbf{e}_{\text{spk}}; \mathbf{e}_{\text{txt}}; \mathbf{e}_{\text{mel}}]$, where $\mathbf{e} \in \mathbb{R}^{(1+L+T) \times D}$. The concatenated sequence is processed by a Transformer [29]-based encoder with bidirectional self-attention, allowing the mel-spectrogram to be encoded while being interpreted based on the prefix information. The guided mel features are obtained by slicing the encoder output:

$$\mathbf{h} = \text{Encoder}(\mathbf{e}), \quad \mathbf{h}' = \mathbf{h}_{1+L:1+L+T}. \quad (1)$$

2.2.3. Quantizer

While the explicit priors provide strong guidance regarding linguistic content and speaker identity, the continuous representation \mathbf{h}' possesses sufficient capacity to retain entangled acoustic details [11]. To encourage the model to encode the residual prosodic style, we introduce a strict information bottleneck via quantization. By restricting the representation capacity to a discrete space, this bottleneck discourages the model from allocating limited resources to redundant information. Instead, it forces the model to rely on the explicit priors, directing its discrete tokens to capture the structural variations corresponding to prosody, such as intonation and rhythm. To further control this information bottleneck via the token rate, we apply linear interpolation before and after quantization. We employ binary spherical quantization (BSQ) [30], which maps the latents to a discrete token sequence by projecting features onto an ℓ_2 -normalized hypersphere and performing binary quantization:

$$\mathbf{h}_q = \text{Quantizer}(\text{Interp}(\mathbf{h}')). \quad (2)$$

The quantized output $\mathbf{h}_q \in \{-1, 1\}^{T' \times d}$ has sequence length T' after interpolation and projected dimension d , producing an implicit binary codebook of size 2^d .

2.2.4. Diffusion-based Decoder

To reconstruct high-fidelity mel-spectrograms from the prosody-oriented discrete tokens, we adopt a diffusion-based decoder. The decoder is based on Diffusion Transformer (DiT) [35], which shares a similar architecture with the encoder but additionally incorporates a diffusion timestep. The timestep embedding is injected into every layer via adaptive layer normalization, indicating the current noise level. Given a clean mel-spectrogram $\mathbf{x} \in \mathbb{R}^{T \times 128}$, we sample Gaussian noise ϵ and construct a noisy input by interpolation, $\mathbf{x}_t = t\mathbf{x} + (1-t)\epsilon$, where $t \sim \mathcal{U}(0, 1)$. To simulate the inference setting where reference audio is provided as a prompt, we apply a random span mask $\mathbf{m} \in \{0, 1\}^T$. The masked noisy mel-spectrogram is defined as

$$\tilde{\mathbf{x}}_t = \mathbf{m} \odot \mathbf{x}_t + (1 - \mathbf{m}) \odot \mathbf{x}, \quad (3)$$

where \odot denotes element-wise multiplication. The noisy mel-spectrogram is projected to the model dimension as $\mathbf{e}_{\text{noisy}} \in \mathbb{R}^{T \times D}$. The quantized representations \mathbf{h}_q are projected and temporally interpolated to match the same length, yielding $\mathbf{e}_q \in \mathbb{R}^{T \times D}$. Similar to the encoder, the decoder receives the concatenated sequence $\mathbf{e} = [\mathbf{e}_{\text{spk}}; \mathbf{e}_{\text{txt}}; \mathbf{e}_{\text{noisy}} + \mathbf{e}_q]$,

$$\mathbf{h} = \text{Decoder}(\mathbf{e}), \quad \mathbf{h}' = \mathbf{h}_{1+L:1+L+T}. \quad (4)$$

Finally, \mathbf{h}' is linearly projected to predict the diffusion velocity field. We optimize the decoder with a flow-matching objective, encouraging the predicted velocity to match the target ($\mathbf{x} - \epsilon$).

Table 1: Objective and subjective results of different methods for voice conversion. The merged evaluation set contains LibriTTS test-clean, test-other, and VCTK. Best and second-best results are in **bold** and underlined, respectively.

Model	Content		Timbre		Prosody		Naturalness	
	WER↓	SIM _r ↑	SIM _s ↓	S-MOS↑	RMSE↓	P-MOS↑	UTMOS↑	N-MOS↑
Source Speech	3.668	0.090	1.000	-	0.000	-	3.897	4.253 ± 0.136
DDDM-VC [31]	9.186	0.297	0.296	3.309 ± 0.139	0.472	3.012 ± 0.174	3.624	2.006 ± 0.152
UniAudio [32]	9.076	0.276	<u>0.178</u>	2.809 ± 0.175	0.470	2.475 ± 0.163	3.631	3.006 ± 0.174
HierSpeech++ [33]	7.568	0.381	0.312	3.494 ± 0.148	<u>0.461</u>	3.389 ± 0.170	3.869	2.679 ± 0.167
FACodec [10]	5.454	0.354	0.347	3.247 ± 0.169	0.455	3.364 ± 0.193	3.544	2.302 ± 0.172
Seed-VC [34]	5.078	<u>0.531</u>	0.239	4.210 ± 0.124	0.473	3.463 ± 0.156	3.712	3.611 ± 0.163
Vevo [11]	<u>4.826</u>	0.478	0.243	4.130 ± 0.139	0.464	<u>3.506</u> ± 0.155	3.800	4.056 ± 0.144
ProsoCodec (Ours)	4.451	0.565	0.167	<u>4.167</u> ± 0.134	0.428	3.852 ± 0.172	<u>3.853</u>	<u>4.000</u> ± 0.148

2.3. Training and Inference

The encoder, quantizer, and decoder of our model are trained in an end-to-end manner with conditional flow matching loss:

$$\mathcal{L}_{CFM} = \mathbb{E}_t \|v_t(\mathbf{x}_t | \mathbf{e}, \theta) - u_t(\mathbf{x}_t)\|^2, \quad (5)$$

where θ refers to the DiT decoder, v_t represents the estimated vector fields, and u_t is the target vector field.

During inference, both the source and reference speech are separately encoded. The decoder then generates the converted speech by conditioning on the speaker representation of the reference speech. Here, the text and token sequences of the reference speech are prepended as a prefix prompt to facilitate in-context consistency.

2.4. Training Strategies for Prosody Preservation

Directly extending random span masking and inpainting to voice conversion can cause prompt-style leakage. Since the prompt and source come from the same utterance, and prompt-based generation holistically imitates the prompt [28], its style leaks into the output rather than the source’s prosody being preserved. To mitigate this issue, we introduce a dual-utterance training strategy that samples paired utterances from the same speaker, using one as the prompt and the other as the source. This leads to better preservation of the source prosody at inference. We alternate this strategy with random span masking during training.

Furthermore, since we do not impose an explicit disentanglement objective, the bottleneck risks retaining excessive acoustic detail to minimize the flow-matching loss. As prosodic cues primarily reside in the low-frequency region of the mel-spectrogram [26, 27], we feed only that band to the encoder, biasing the tokens toward prosody. In contrast, the prompt fed to the decoder uses the full-band mel-spectrogram to provide speaker characteristics.

3. Experiments

3.1. Datasets

Our model is trained on the LibriTTS [36] dataset, which comprises 585 hours of 24 kHz read speech from 2,456 speakers. To evaluate its performance, we utilize the LibriTTS test-clean and test-other splits, alongside the VCTK dataset [37] for assessing out-of-domain generalization. From each evaluation set, we randomly sample 1,000 utterances with durations between 2 and 8 seconds. For each sample, we select a reference prompt from the same speaker for speech resynthesis, and another from a different speaker for zero-shot voice conversion.

3.2. Implementation Details

The speech waveform is converted into a mel-spectrogram using 128 mel filter banks with an 80 ms window length and 20 ms hop size, resulting in a frame rate of 50 Hz. We use pretrained Qwen3-ASR [21] and CAM++ [23] to obtain transcriptions and speaker embeddings, respectively. ProsoCodec comprises a Transformer encoder [29] and a Diffusion Transformer (DiT) decoder [35], both initialized from TaDiCodec [24] for efficient training. The encoder and decoder consist of 8 and 16 layers, respectively, each with a hidden size of 1024, an intermediate size of 4096, and 16 attention heads. We train the model with the AdamW [38] optimizer using a learning rate of 2×10^{-5} , $\beta_1 = 0.9$, $\beta_2 = 0.999$, and a weight decay of 0.01. The learning rate is linearly warmed up over the first 10k updates, and training is conducted for 150k updates with a global batch size of 160 seconds. During inference, mel-spectrograms are generated using 32 sampling steps via the Euler ODE solver and subsequently converted into 24 kHz waveforms by a Vocos vocoder [39].

3.3. Evaluation Metrics

The quality of the synthesized speech is assessed through a comprehensive suite of subjective and objective metrics. Given that subjective evaluation remains the gold standard for synthesized speech [40], we conducted Mean Opinion Score (MOS) tests. For each test subset, 15 listeners rated 10 randomly sampled utterances on a 5-point Likert scale. We specifically measured three criteria: speaker similarity (**S-MOS**) to the reference, prosody similarity (**P-MOS**) to the source, and overall naturalness (**N-MOS**). For objective evaluation, we compute Word Error Rate (**WER**) by comparing the source transcription with the automatic transcription of the generated sample. To avoid evaluation bias, we utilize Whisper-large-v3 [20], which is different from the ASR model used in ProsoCodec. To evaluate speaker similarity, we compute the cosine similarity between the speaker embeddings of the source and synthesized speech (**SIM_r**), where the embeddings are extracted using the pre-trained WavLM-Large model [22]. Furthermore, to quantify potential speaker identity leakage during voice conversion, we report the **SIM_s** between the source and converted speech; here, lower similarity scores indicate superior quality. For prosody preservation, we adopt fundamental frequency (f_0) as a proxy for prosody dynamics. Specifically, we compute the Root Mean Square Error (**RMSE**) between the source and the generated speech, where f_0 values are extracted via Harvest [41] and log-scaled to align with human perception. **UTMOS** [42] is used to estimate perceptual naturalness of the generated speech, serving as an automatic MOS predictor.

Table 2: Ablation study about zero-shot voice conversion on LibriTTS test-clean dataset.

Method	WER↓	SIM _r ↑	SIM _s ↓	RMSE↓	UTMOS↑
Source Speech	3.921	0.081	1.000	0.000	4.091
ProsoCodec	4.684	0.608	0.168	0.376	3.978
– low-freq. mel	4.810	0.585	0.192	0.386	4.004
– dual-utter.	5.630	0.634	0.166	0.421	3.884
– speaker cond.	6.793	0.600	0.186	0.397	3.891
– text cond.	86.601	0.569	0.108	0.434	3.569

Table 3: Ablation study about bottleneck configuration on LibriTTS test-clean dataset. FR: frame rate in Hz; CS: codebook size; BR: bitrate in bps.

Resynthesis			Voice Conversion						
FR	CS	BR	WER↓	SIM↑	RMSE↓	WER↓	SIM _r ↑	SIM _s ↓	RMSE↓
25	4096	300	4.28	0.72	0.21	4.49	0.56	0.20	0.36
12.5	16384	175	4.54	0.71	0.22	4.71	0.60	0.18	0.37
12.5	4096	150	4.30	0.71	0.22	4.68	0.61	0.17	0.38
12.5	1024	125	4.57	0.70	0.22	4.91	0.62	0.16	0.38
6.25	4096	75	4.85	0.69	0.23	6.17	0.64	0.14	0.41
Zero-Shot TTS			5.85	0.67	0.29	6.04	0.68	0.11	0.48

4. Results

4.1. Comparisons with Previous Methods

We compare ProsoCodec with several state-of-the-art open-source voice conversion models. In particular, we employ DDDM-VC [31], HierSpeech++ [33], and Vevo [11], which are resynthesis-based models that utilize decomposed speech attributes; UniAudio [32], an LLM-based unified audio generation framework built on a neural audio codec; FACodec [10], a factorized speech codec-based model; and Seed-VC [34], a prompt-based diffusion model trained with a timbre-shift strategy. Table 1 reports the objective and subjective results on the merged evaluation set. The proposed ProsoCodec model demonstrates superior performance in terms of WER, indicating that the linguistic content of the source speech is well preserved during conversion. The lower SIM_s and higher SIM_r further show that our method effectively suppresses timbre leakage from the source speaker while adopting the target speaker timbre.

These improvements can be attributed to several design choices in ProsoCodec, including explicit conditioning on robust transcriptions and speaker embeddings, the prompt-based conversion mechanism with dual-utterance training strategy, and the use of low-frequency band mel-spectrograms. We analyze the contribution of each component in the ablation studies.

4.2. Ablation Study

Table 2 presents a cumulative ablation on the LibriTTS test-clean dataset to analyze the contribution of each component in ProsoCodec. When the low-frequency mel-spectrogram is replaced with full-band mel-spectrogram input, the model shows degraded performance across several metrics, suggesting that restricting the acoustic bandwidth helps the codec focus on prosodic variations rather than fine-grained spectral details. Removing the dual-utterance training strategy further degrades prosody preservation, as reflected by the increased RMSE. This result suggests that the strategy encourages the codec to retain prosodic information required for generation by preventing the model from trivially copying information from the prompt. When speaker conditioning is removed, speaker similarity drops,

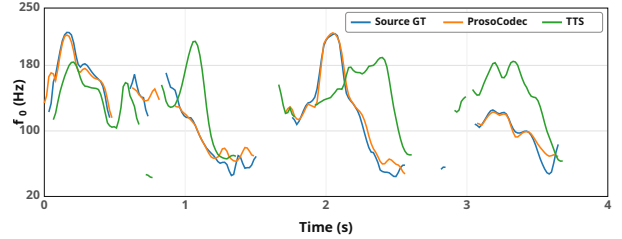


Figure 2: Pitch contour of resynthesis result of ProsoCodec along with source speech and zero-shot TTS output.

reflected by decreased SIM_r and increased SIM_s, indicating stronger leakage from the source speaker. Finally, removing text conditioning leads to a substantial increase in WER. These results highlight that explicit text and speaker conditioning play critical roles in guiding the codec to encode residual information within the limited bitrate of the discrete bottleneck.

4.3. Analysis

Table 3 presents the ablation study on the codec bottleneck configuration. Consistent with observations in previous studies, strong reconstruction does not necessarily imply strong conversion. Increasing the bitrate improves resynthesis performance as more acoustic details can be preserved, however, this also leads to source timbre leakage that harms voice conversion performance. Considering this trade-off, we select a bottleneck configuration of 12.5 Hz frame rate with a codebook size of 4096 (150 bps), which provides a good balance between content preservation and voice conversion quality.

We further analyze an extreme scenario where the decoder is trained entirely without codec tokens. Since it remains conditioned on the acoustic prompt, speaker embedding, and text, this setup effectively operates as a zero-shot TTS or voice cloning model [43, 44]. While this baseline successfully generates speech that follows the prompt’s speaker characteristics well, it derives its prosodic patterns directly from the prompt or synthesizes generalized text-driven prosody. This behavior is also reflected in the pitch contours in Fig. 2. ProsoCodec closely resynthesizes the source f_0 trajectory, whereas the model without codec tokens produces pitch that deviates from the source. Such behavior is fundamentally misaligned with the objective of voice conversion, which requires mapping the exact prosodic variations of the source.

5. Conclusion

In this paper, we presented ProsoCodec, a prosody-oriented speech codec designed for high-fidelity zero-shot voice conversion. Explicitly conditioned on linguistic content and speaker identity, our model effectively captures residual prosodic variations without compromising speaker-conditioned nuances and expressiveness. We further introduced a dual-utterance training strategy and low-frequency input biasing to significantly mitigate the common issue of prompt-style leakage. Extensive evaluations demonstrate that ProsoCodec excels in preserving both source prosody and target speaker timbre, outperforming existing state-of-the-art frameworks. We believe that our approach to residual prosody modeling provides a robust foundation for prosodic representations, which can be widely utilized across various speech tasks, including expressive text-to-speech synthesis, voice conversion, and spoken dialogue systems.

6. Acknowledgments

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korean government (MSIT, RS-2025-02263977, Development of Communication Platform supporting User Anonymization and Finger Spelling-Based Input Interface for Protecting the Privacy of Deaf Individuals).

7. Generative AI Use Disclosure

Generative AI tools were used only for editing and polishing this manuscript and were not used for producing any significant part of the manuscript.

8. References

- [1] S. Chen, C. Wang, Y. Wu, Z. Zhang, L. Zhou, S. Liu, Z. Chen, Y. Liu, H. Wang, J. Li *et al.*, “Neural codec language models are zero-shot text to speech synthesizers,” *IEEE/ACM Trans. on Audio, Speech, and Language Processing*, 2025.
- [2] P. Anastassiou, J. Chen, J. Chen, Y. Chen, Z. Chen, Z. Chen, J. Cong, L. Deng, C. Ding, L. Gao *et al.*, “Seed-tts: A family of high-quality versatile speech generation models,” *arXiv preprint arXiv:2406.02430*, 2024.
- [3] L. Meng, L. Zhou, S. Liu, S. Chen, B. Han, S. Hu, Y. Liu, J. Li, S. Zhao, X. Wu *et al.*, “Autoregressive speech synthesis without vector quantization,” in *Proc. ACL*, 2025.
- [4] Z. Du, C. Gao, Y. Wang, F. Yu, T. Zhao, H. Wang, X. Lv, H. Wang, C. Ni, X. Shi *et al.*, “Cosyvoice 3: Towards in-the-wild speech generation via scaling-up and post-training,” *arXiv preprint arXiv:2505.17589*, 2025.
- [5] A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, “High fidelity neural audio compression,” *Trans. on Machine Learning Research*, 2023.
- [6] S. Ji, Z. Jiang, W. Wang, Y. Chen, M. Fang, J. Zuo, Q. Yang, X. Cheng, Z. Wang, R. Li *et al.*, “Wavtokenizer: an efficient acoustic discrete codec tokenizer for audio language modeling,” in *Proc. ICLR*, 2025.
- [7] S. Chen, S. Liu, L. Zhou, Y. Liu, X. Tan, J. Li, S. Zhao, Y. Qian, and F. Wei, “Vall-e 2: Neural codec language models are human parity zero-shot text to speech synthesizers,” *arXiv preprint arXiv:2406.05370*, 2024.
- [8] P. Peng, P.-Y. Huang, D. Li, A. Mohamed, and D. Harwath, “Voicecraft: Zero-shot speech editing and text-to-speech in the wild,” in *Proc. ACL*, 2024.
- [9] Z. Ye, X. Zhu, C.-M. Chan, X. Wang, X. Tan, J. Lei, Y. Peng, H. Liu, Y. Jin, Z. Dai *et al.*, “Llaza: Scaling train-time and inference-time compute for llama-based speech synthesis,” *arXiv preprint arXiv:2502.04128*, 2025.
- [10] Z. Ju, Y. Wang, K. Shen, X. Tan, D. Xin, D. Yang, Y. Liu, Y. Leng, K. Song, S. Tang *et al.*, “Naturalspeech 3: Zero-shot speech synthesis with factorized codec and diffusion models,” in *Proc. ICML*, 2024.
- [11] X. Zhang, X. Zhang, K. Peng, Z. Tang, V. Manohar, Y. Liu, J. Hwang, D. Li, Y. Wang, J. Chan *et al.*, “Vevo: Controllable zero-shot voice imitation with self-supervised disentanglement,” in *Proc. ICLR*, 2025.
- [12] R. Wang, Y. Ding, L. Li, and C. Fan, “One-shot voice conversion using star-gan,” in *Proc. ICASSP*, 2020.
- [13] A. Polyak, Y. Adi, J. Copet, E. Kharitonov, K. Lakhotia, W.-N. Hsu, A. Mohamed, and E. Dupoux, “Speech resynthesis from discrete disentangled self-supervised representations,” in *Proc. Interspeech*, 2021.
- [14] H.-S. Choi, J. Lee, W. Kim, J. Lee, H. Heo, and K. Lee, “Neural analysis and synthesis: Reconstructing speech from self-supervised representations,” in *NeurIPS*, 2021.
- [15] K. Qian, Y. Zhang, S. Chang, M. Hasegawa-Johnson, and D. Cox, “Unsupervised speech decomposition via triple information bottleneck,” in *Proc. ICML*, 2020.
- [16] R. Skerry-Ryan, E. Battenberg, Y. Xiao, Y. Wang, D. Stanton, J. Shor, R. Weiss, R. Clark, and R. A. Saurous, “Towards end-to-end prosody transfer for expressive speech synthesis with tacotron,” in *Proc. ICML*, 2018.
- [17] L. Qu, C. Weber, W. Wang, J. Jin, Y. Gao, T. Li, and S. Wermter, “Disentanglement of prosody representations via diffusion models and scheduled gradient reversal,” *IEEE Trans. Neural Networks and Learning Systems*, 2025.
- [18] H. Li, L. Xue, H. Guo, X. Zhu, Y. Lv, L. Xie, Y. Chen, H. Yin, and Z. Li, “Single-codec: Single-codebook speech codec towards high-performance speech generation,” in *Proc. Interspeech*, 2024.
- [19] J. D. Parker, A. Smirnov, J. Pons, C. Carr, Z. Zukowski, Z. Evans, and X. Liu, “Scaling transformers for low-bitrate high-quality speech coding,” in *Proc. ICLR*, 2025.
- [20] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, “Robust speech recognition via large-scale weak supervision,” in *Proc. ICML*, 2023.
- [21] X. Shi, X. Wang, Z. Guo, Y. Wang, P. Zhang, X. Zhang, Z. Guo, H. Hao, Y. Xi, B. Yang *et al.*, “Qwen3-asr technical report,” *arXiv preprint arXiv:2601.21337*, 2026.
- [22] 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*, 2022.
- [23] H. Wang, S. Zheng, Y. Chen, L. Cheng, and Q. Chen, “Cam++: A fast and efficient network for speaker verification using context-aware masking,” in *Proc. Interspeech*, 2023.
- [24] Y. Wang, D. Chen, X. Zhang, J. Zhang, J. Li, and Z. Wu, “Tadicodec: Text-aware diffusion speech tokenizer for speech language modeling,” in *NeurIPS*, 2025.
- [25] Y. Wang, Z. Tang, Y. Wang, A. Hinsvark, Y. Liu, Y. Li, K. Peng, J. Ao, M. Ma, M. Seltzer *et al.*, “Scaling speech tokenizers with diffusion autoencoders,” in *Proc. ICLR*, 2025.
- [26] Y. Ren, M. Lei, Z. Huang, S. Zhang, Q. Chen, Z. Yan, and Z. Zhao, “Prosospeech: Enhancing prosody with quantized vector pre-training in text-to-speech,” in *Proc. ICASSP*, 2022.
- [27] Z. Jiang, Y. Ren, Z. Ye, J. Liu, C. Zhang, Q. Yang, S. Ji, R. Huang, C. Wang, X. Yin *et al.*, “Mega-tts: Zero-shot text-to-speech at scale with intrinsic inductive bias,” *arXiv preprint arXiv:2306.03509*, 2023.
- [28] H. Pei, S. Liu, Y. Liu, J. Yu, Y. Qian, G. Huang, S. Zhao, and Y. Lu, “A unified neural codec language model for selective editable text to speech generation,” *arXiv preprint arXiv:2601.12480*, 2026.
- [29] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” in *NeurIPS*, 2017.
- [30] Y. Zhao, Y. Xiong, and P. Krähenbühl, “Image and video tokenization with binary spherical quantization,” in *Proc. ICLR*, 2025.
- [31] H.-Y. Choi, S.-H. Lee, and S.-W. Lee, “Dddm-vc: Decoupled denoising diffusion models with disentangled representation and prior mixup for verified robust voice conversion,” in *Proc. AAAI*, 2024.
- [32] D. Yang, J. Tian, X. Tan, R. Huang, S. Liu, X. Chang, J. Shi, S. Zhao, J. Bian, X. Wu *et al.*, “Uniaudio: An audio foundation model toward universal audio generation,” in *Proc. ICML*, 2024.
- [33] S.-H. Lee, H.-Y. Choi, S.-B. Kim, and S.-W. Lee, “Hierspeech++: Bridging the gap between semantic and acoustic representation of speech by hierarchical variational inference for zero-shot speech synthesis,” *IEEE Trans. Neural Networks and Learning Systems*, 2025.
- [34] S. Liu, “Zero-shot voice conversion with diffusion transformers,” *arXiv preprint arXiv:2411.09943*, 2024.

- [35] W. Peebles and S. Xie, “Scalable diffusion models with transformers,” in *Proc. ICCV*, 2023.
- [36] 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 *Proc. Interspeech*, 2019.
- [37] J. Yamagishi, C. Veaux, and K. MacDonald, “Cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit (version 0.92),” 2019.
- [38] I. Loshchilov and F. Hutter, “Decoupled weight decay regularization,” in *Proc. ICLR*, 2019.
- [39] H. Siuzdak, “Vocos: Closing the gap between time-domain and fourier-based neural vocoders for high-quality audio synthesis,” in *Proc. ICLR*, 2024.
- [40] T. Saeki, S. Maiti, S. Takamichi, S. Watanabe, and H. Saruwatari, “Speechbertscore: Reference-aware automatic evaluation of speech generation leveraging nlp evaluation metrics,” in *Proc. Interspeech*, 2024.
- [41] M. Morise, “A high-performance fundamental frequency estimator from speech signals,” in *Proc. Interspeech*, 2017.
- [42] T. Saeki, D. Xin, W. Nakata, T. Koriyama, S. Takamichi, and H. Saruwatari, “Utmos: Utokyo-sarulab system for voicemos challenge 2022,” in *Proc. Interspeech*, 2022.
- [43] M. Le, A. Vyas, B. Shi, B. Karrer, L. Sari, R. Moritz, M. Williamson, V. Manohar, Y. Adi, J. Mahadeokar *et al.*, “Voicebox: Text-guided multilingual universal speech generation at scale,” in *NeurIPS*, 2023.
- [44] Y. Chen, Z. Niu, Z. Ma, K. Deng, C. Wang, J. Zhao, K. Yu, and X. Chen, “F5-tts: A fairytaler that fakes fluent and faithful speech with flow matching,” in *Proc. ACL*, 2025.