Interspeech 2024
1-5 September 2024, Kos, Greece

High Fidelity Text-to-Speech Via Discrete Tokens Using Token Transducer and Group Masked Language Model

Joun Yeop Lee[1*], *Myeonghun Jeong*[2*], *Minchan Kim*[2*], *Ji-Hyun Lee*[1], *Hoon-Young Cho*[1], *Nam Soo Kim*[2]

[1]Samsung Research, Seoul, Republic of Korea
[2]Department of ECE and INMC, Seoul National University, Seoul, Republic of Korea

jounyeop.lee@samsung.com, {mhjeong, mckim}@hi.snu.ac.kr

Abstract

We propose a novel two-stage text-to-speech (TTS) framework with two types of discrete tokens, i.e., semantic and acoustic tokens, for high-fidelity speech synthesis. It features two core components: the *Interpreting* module, which processes text and a speech prompt into semantic tokens focusing on linguistic contents and alignment, and the *Speaking* module, which captures the timbre of the target voice to generate acoustic tokens from semantic tokens, enriching speech reconstruction. The *Interpreting* stage employs a transducer for its robustness in aligning text to speech. In contrast, the *Speaking* stage utilizes a Conformer-based architecture integrated with a Grouped Masked Language Model (G-MLM) to boost computational efficiency. Our experiments verify that this innovative structure surpasses the conventional models in the zero-shot scenario in terms of speech quality and speaker similarity.

Index Terms: speech synthesis, neural transducer, masked language modeling, semantic token, acoustic token

1. Introduction

Recently, there has been a notable shift in Text-to-Speech (TTS) research towards the adoption of discrete speech tokens as intermediate features [1–4], which presents diverse options for model architecture and inference strategies. These tokens are broadly categorized into two types: semantic tokens and acoustic tokens, depending on their embedded information. Semantic tokens are typically derived through quantization applied to speech features containing contextualized linguistic details. These quantized speech features are sourced from various speech encoders such as self-supervised speech models [5, 6] or speech recognition models [7]. By encoding disentangled linguistic information into discrete codes, semantic tokens alleviate the complexities arising from acoustic diversity, thereby enabling a sharper focus on semantic content critical for enhancing intelligibility. In contrast, acoustic tokens represent codewords generated by neural codecs [8–11], which have witnessed significant advancements in recent years. These tokens encapsulate acoustic details of raw waveforms, serving as alternatives to traditional frame-level acoustic features such as mel-spectrograms.

Incorporating discrete tokens into TTS offers numerous advantages over conventional speech modeling in the continuous domain. Above all, targeting discrete tokens simplifies the representation of one-to-many mappings by facilitating a categorical distribution for the output space, addressing the complex challenges posed by continuous domain generative modeling. Moreover, the discrete output space enables the integration of various specialized schemes. Particularly noteworthy is its ability to leverage recent advancements in large language models (LLM). For example, SPEAR-TTS [1] and VALL-E [2] employ in-context learning for prompt-oriented zero-shot TTS, complemented by speech continuation tasks. Additionally, recent progress in iterative sampling with masked language models (MLM) [3, 12, 13] has demonstrated high-fidelity speech synthesis with parallel computation, a capability confined to discrete sequences. Furthermore, the discrete output space can facilitate the development of robust alignment modeling, simplifying the adoption of transducers [14, 15] within the TTS framework.

In this work, we propose a high-fidelity TTS framework designed to optimize the use of semantic and acoustic tokens, leveraging the benefits of discrete tokenization. This framework follows a two-stage procedure: converting text into semantic tokens (*Interpreting*) and then into acoustic tokens (*Speaking*). There have been attempts to use such a two-stage architecture with discrete tokens. For example, SPEAR-TTS uses “reading” and “speaking” modules similar to ours. However, the “reading” utilizes an encoder-decoder transformer structure that cannot guarantee a monotonic alignment path, and has some complicated training methods that are hard to reproduce. Also, “speaking” uses a huge decoder-only transformer structure which results in computational inefficiency induced by sequential generation. On the contrary, in our architecture, the *Interpreting* stage employs a transducer [16] for robust alignment modeling, capitalizing on the inherent monotonic alignment constraint in seq2seq models for efficient computation and alignment modeling. Also, in the *Speaking* stage, we employ an MLM with parallel iterative sampling for acoustic token generation, leveraging the pre-established alignment between semantic and acoustic tokens to enhance inference speed and efficiency. Specifically, we utilize a group masked language model [12] with HiFi-Codec [10] to reduce sampling iterations and improve speech quality. Both phases efficiently exploit conditional dependencies between output sequences during inference, a crucial factor for our high-fidelity speech generation. Experimental results on zero-shot TTS demonstrate the superiority of our model over baseline models in terms of speech quality and speaker similarity. Audio samples are available on our demo page[*].

2. Method

The proposed TTS framework consists of a text-to-semantic token stage, named *Interpreting*, and semantic to the acoustic token stage, named *Speaking*. Then, the acoustic tokens are converted into raw wave through a codec decoder. The entire framework is illustrated in Figure 1.

[*]These authors contributed equally to this work.

[*]https://srtts.github.io/interpreting-speaking/

10.21437/Interspeech.2024-465

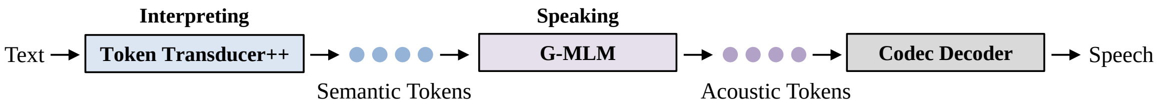
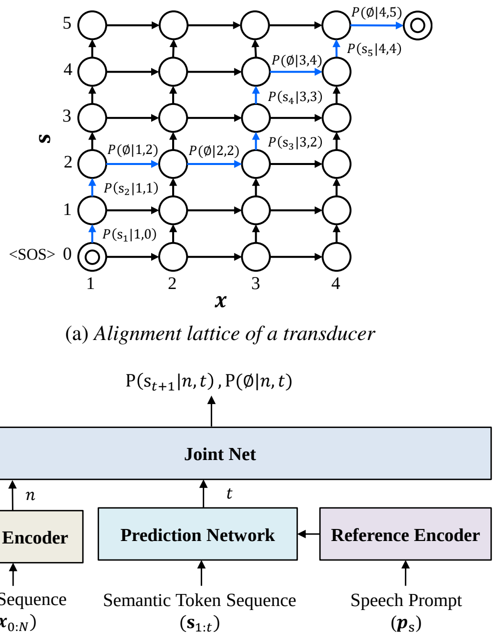


Figure 1: *Overall architecture of the proposed model.*



(a) *Alignment lattice of a transducer*

$P(s_{t+1}|n,t)$, $P(\varnothing|n,t)$
Joint Net
n
t
Text Encoder
Prediction Network
Reference Encoder
Text Sequence ($\boldsymbol{x}_{0:N}$)
Semantic Token Sequence ($\mathbf{s}_{1:t}$)
Speech Prompt ($\boldsymbol{p}_s$)

(b) *Architecture of Token Transducer++*

Figure 2: *Interpreting procedure*

As semantic tokens, we exploit the index sequence of k-means clustering on wav2vec 2.0 [6] embeddings, akin to [14]. According to [14], semantic tokens mainly focus on phonetic information, but it also dealing with some prosodic information such as speech rate and overall pitch contour. For acoustic token extraction, we leverage the HiFi-Codec [10], a sophisticated neural codec tailored for TTS applications. The HiFi-Codec employs the Group-Residual Vector Quantization (G-RVQ) to derive multiple streams of discrete token sequences. Specifically, we use the bi-group and bi-depth G-RVQ to maintain the balance between performance and computational efficiency. We denote the semantic token at time step t, as s_t, while the acoustic token is represented as y_t, each comprising $y_t^{i,j}$, with $i, j \in \{0, 1\}$ denoting indices corresponding to the group and depth of G-RVQ, respectively.

2.1. Interpreting

In the *Interpreting* stage, we generate a semantic token sequence $\mathbf{s}_{0:T}$ from the input text $\mathbf{x}_{1:N}$, addressing alignment between text and semantic tokens, as well as controlling prosody embedded in semantic tokens. While attention-based language models are often considered for this seq2seq translation task, they cannot exploit the inherent monotonic constraint of alignment, making them susceptible to misalignment issues and requiring computationally intensive key-query matching processes. Instead, we adopt a transducer, referred to as Token Transducer++, a modified version of the Token Transducer of [14]. Transducers are specifically designed architecture for discrete seq2seq models with monotonic alignment constraint, achieved through alignment lattice (Figure 2. (a)) and a special blank token $\varnothing$, which indicates the transition to the next input frame (horizontal arrows in Figure 2. (a)). The Token Transducer++ follows the same transducer formulation of [14], where the training objective is expressed as:

$$\begin{aligned}\mathcal{L}_{inter} &= -\log P(\mathbf{s}|\mathbf{x}) \\ &= -\log \sum_{\mathcal{A}\in\mathcal{F}^{-1}(\mathbf{s})} P(\mathcal{A}|\mathbf{x},\mathbf{p}_s).\end{aligned} \tag{1}$$

Here, $\mathcal{A}$ represents a possible monotonic path and, $\mathcal{F}^{-1}$ denotes the inverse of the blank removal function $\mathcal{F}$, indicating the marginalization over all $\mathcal{A}$. Also, $\mathbf{p}_s$ denotes the speech prompt for the semantic token generation, i.e., semantic prompt, which controls paralinguistic information embedded in semantic token sequence such as speech rate and pitch contour.

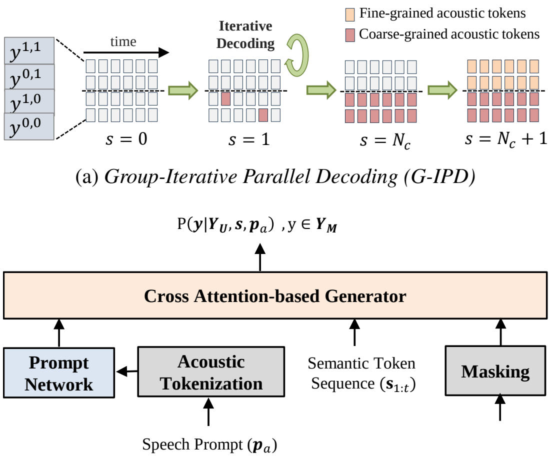
As depicted in Figure 2 (a), the Token Transducer++ has a modularized structure comprising four components: a text encoder, a reference encoder, a prediction network, and a joint network. The text encoder processes text input, producing a sequence of text embeddings. The reference encoder encodes speech prompt into fixed-dimensional reference embedding. The prediction network operates in an autoregressive manner, leveraging preceding semantic tokens and the reference embedding to generate the semantic token embedding for the current frame. Lastly, the joint network integrates text and semantic token embeddings, generating output probabilities, crucial for lattice construction. Notably, the Token Transducer++ mirrors the architecture of each module detailed in [14]. However, there are two drawbacks of transducers: (1) the autoregressive nature of the joint network presents a significant computational bottleneck during inference, and (2) the joint network's frame-wise computation neglects temporal context. To overcome these problems, when compared to the original Token Transducer, we largely reduce the size of the joint network and inject reference embedding to the prediction network instead of the joint network for temporal consideration of reference embedding. We add the reference embedding to the input embedding of the prediction network. These simple modifications not only boost inference speed but also enhance overall performance.

2.2. Speaking

The *Speaking* stage aims to translate semantic tokens produced in the *Interpreting* stage into acoustic tokens, utilizing a prompt for acoustic guidance. We tackle this pre-aligned seq2seq task via Masked Language Modeling (MLM), which effectively incorporates prompt speaker information through in-context learning. To achieve this, the *Speaking* module is meticulously trained to optimize the prediction of masked acoustic tokens as follows:

$$\mathcal{L}_{speak} = -\sum_{\forall y\in\mathbf{Y}_M} P(y|\mathbf{Y}_U, \mathbf{s}, \mathbf{p}_a), \tag{2}$$

where $\mathbf{Y}_M$, $\mathbf{Y}_U$ represents masked and unmasked target acoustic tokens. Also, $\mathbf{p}_a$ denotes an acoustic prompt, and it provides


$y^{1,1}$
$y^{0,1}$
$y^{1,0}$
$y^{0,0}$
time
Iterative Decoding
Fine-grained acoustic tokens
Coarse-grained acoustic tokens
$s = 0$
$s = 1$
$s = N_c$
$s = N_c + 1$
$\mathrm{P}(\boldsymbol{y}|\boldsymbol{Y_U}, \boldsymbol{s}, \boldsymbol{p_a})$,y ∈ $\boldsymbol{Y_M}$
Cross Attention-based Generator
Prompt Network
Acoustic Tokenization
Semantic Token Sequence ($\boldsymbol{s}_{1:t}$)
Masking
Speech Prompt ($\boldsymbol{p}_a$)

(a) *Group-Iterative Parallel Decoding (G-IPD)*

(b) *Architecture of Speaking*

Figure 3: *Speaking procedure*

detailed acoustic attributes (i.e., timbre, acoustic condition) of the prompt speaker.

Given the characteristics of RVQ-based acoustic tokens, there have been MLM approaches that capture both of temporal and RVQ-level-wise conditional dependency [3,17]. Among these methods, we employ the Group-MLM (G-MLM) approach in our *Speaking* module, which is specifically designed for G-RVQ acoustic tokens. By simultaneously masking tokens at the same level across different groups, it additionally captures group-wise conditional dependency. Such a masking strategy enables the model to more easily predict masked tokens, ultimately facilitating efficient decoding. The inference method that mimics this approach is illustrated in Figure 3. (a). First, coarse-grained acoustic tokens from different groups (e.g., $\mathbf{y}^{0,0}$, $\mathbf{y}^{1,0}$) are obtained through N_c iterations of iterative parallel decoding. Then, fine-grained acoustic tokens are predicted all at once. Note that this sampling scheme, named Group-Iterative Parallel Decoding (G-IPD), which originates from the masking strategy of G-MLM, improves the audio quality even with a small number of iterations.

As illustrated in Figure 3. (b), the *Speaking* part comprises a prompt network and a generator based on cross-attention mechanisms. Each module follows the architecture described in [12]. The prompt network and generator are based on a bidirectional conformer structure [18] to learn the underlying contextual information. The prompt embedding processed from the prompt network operates as the key and value, while the aggregated embeddings of the semantic tokens and partially masked acoustic tokens serve as the query. This cross-attention-based architecture has been proven to be efficient in terms of computational cost and inference speed, as it allows for caching of key and value during iterative decoding. This structure streamlines the whole inference process while offering rich prompt information [12, 19].

2.3. Discussion on Framework Design

The division of the two stages offers several distinct advantages. Firstly, each stage can concentrate on different aspects of TTS objective. The *Interpreting* stage can solely prioritize alignment modeling and linguistic accuracy, whereas the *Speaking* stage can focus more on high-fidelity and handling one-to-many mappings due to acoustic diversity. We design sophisticated architectures tailored to each role, considering both speech quality and inference efficiency. Additionally, from a data perspective, the *Interpreting* stage relies on text and speech pair data, whereas the *Speaking* stage doesn't require text annotations. This flexibility enables us to leverage unlabeled data, which is more abundant, resulting in higher speech quality and the ability to represent diverse acoustic conditions. Moreover, regarding zero-shot adaptation, we can separately control each stage as they govern different aspects of paralinguistic information. The *Interpreting* stage manages speech rate and global prosodic dynamics, while the *Speaking* stage focuses on timbre and acoustic attributes. This separation provides us with the flexibility to independently control each component using different speech prompts, thereby enhancing overall controllability across a broader spectrum.

3. Experiments

3.1. Experimental Setting

Dataset: We conducted zero-shot TTS experiments using the LibriTTS [20] corpus. We used all of the training subsets (train-clean-100, train-clean-360, train-other-500) for training and the test-clean subset for evaluation. All the sentences were phonemized to International Phonetic Alphabet (IPA) using the phonemizer library[*].

Implementation Details: Following [14], we employed semantic tokens obtained through k-means clustering on the wav2vec 2.0-XLSR model [6][*], with k set to 512. The acoustic tokenization was conducted using the pre-trained HiFi-Codec[*], optimized for 24kHz speech samples and a 320 times downsampling rate. We constructed the Token Transducer++ with same the configuration as the Token Transducer in [14], but we reduced the number of feed-forward blocks from 3 to 1. The Token Transducer++ were trained for 30 epochs with dynamic batch size containing up to 240 seconds on two Quadro RTX 8000 GPUs. The implementation of *Speaking* stage followed [12], unless specified otherwise. The G-MLM underwent training for $350k$ iterations with a batch size of 128 utilizing four NVIDIA A100 GPUs. We set N_c to 16, resulting in a total of 17 iterations for sampling.

Baselines: We built three baseline models for comparison: VITS, VALLE-X [21], and Kim et al. [14]. To adapt the baseline VITS to the zero-shot adaptive scenario, we incorporated the ECAPA-TDNN [22] structure as a reference encoder. We used the open-source implementation for VALLE-X [*]

Evaluation Metrics: For objective evaluations, we assessed the character error rate (CER) using a pretrained Whisper large model [7], leveraging the official implementation[*]. Additionally, we conducted averaged speaker embedding cosine similarity (SECS) analysis to evaluate speaker similarity between speech prompts and synthesized samples, utilizing the pre-trained WavLM large speaker verification model[*]. We randomly selected 500 utterances from the test dataset for these objective assessments.

For subjective evaluation, we measured Mean Opinion

[*] https://github.com/bootphon/phonemizer

[*] https://huggingface.co/facebook/wav2vec2-xlsr-53-espeak-cv-ft

[*] https://github.com/yangdongchao/AcademiCodec

[*] https://github.com/Plachtaa/VALL-E-X

[*] https://github.com/openai/whisper

[*] https://github.com/microsoft/UniSpeech/tree/main/downstreams/speaker_verification

Table 1: *Results of zero-shot TTS. MOS and SMOS are represented with 95% confidence intervals.*

Method	MOS	SMOS	CER	SECS
Ground Truth	4.21±0.09	4.13±0.10	0.97	0.653
VITS	3.52±0.16	3.23±0.20	4.81	0.385
VALLE-X	3.75±0.14	3.08±0.21	8.72	0.425
Kim et al. [14]	3.69±0.16	3.55±0.21	2.47	0.467
Proposed	**3.94±0.14**	**3.64±0.19**	**2.34**	**0.512**

Score (MOS) and Similarity MOS (SMOS). The MOS test evaluated perceptual speech quality as rated by testers. In the SMOS test, testers rated the similarity of speaker and prosody between speech prompts and synthesized samples. Both tests utilized a scoring range from 1 to 5, with higher scores indicating better performance. For both subjective tests, 90 testers participated via Amazon MTurk* platform.

3.2. Results: Zero-Shot Multi-Speaker TTS

According to Table 1, our proposed model exhibited superior performance compared to the baselines across all assessments. Notably, VALLE-X demonstrated the worst CER due to misalignment issues stemming from the lack of a monotonic alignment constraint. In contrast, alignment models based on the transducer approach, including [14] and the proposed model, exhibited higher intelligibility owing to their robust alignment capabilities. Furthermore, when compared to the transducer-based model [14], our proposed model presented enhanced speech quality and speaker similarity, attributed to the G-MLM utilized in the *Speaking* stage. Despite some inconsistencies in the results, it is noteworthy that our proposed architecture outperformed all assessed perspectives.

3.3. Ablation: TokenTransducer++

To assess the improvement brought by Token Transducer++ over the original Token Transducer [14], we conducted a comprehensive comparative analysis employing the same proposed $Speaking$ stage. We compared objective measures including real-time factor (RTF) of the Token Transducer inference, CER and SECS. Additionally, we computed the negative log-likelihood (NLL), employing the same methodology as the training objective, for both transducers. As detailed in Table 2, Token Transducer++ demonstrates significantly improved inference speed compared to the Token Transducer, even with better CER and NLL scores. While Token Transducer++ exhibits marginally lower SECS, we note that SECS is notably influenced by the *Speaking* stage, as evidenced by comparisons with [14] presented in Table 1. These results underscore the efficacy of prosody conditioning within the prediction network, even with a more compact joint network architecture. This integration not only accelerated inference speed but also yields overall performance enhancements.

3.4. Ablation: Prosody Controllability

We investigated the controllability using different speech prompts for both the *Interpreting* and *Speaking* stages. Following the approach outlined in [14], we generated samples under two conditions: (1) when $\mathbf{p}_s$ equals $\mathbf{p}_a$, and (2) when $\mathbf{p}_s$ differs from $\mathbf{p}_a$ (selected from different speakers). The former represents the general zero-shot TTS scenario, while the latter assesses the ability to independently control semantic paralinguistic elements (such as speech rate and prosody) and acoustic conditions (including speaker identity, timbre, and environmental factors). This separation allows for disentangled prosody controllability. In Table 3, we computed the speaker embedding cosine similarity (SECS) between the generated speech and the speech prompts $\mathbf{p}_s$ and $\mathbf{p}_a$, comparing the proposed method with different *Speaking* stage implementations, using the speech generator from [14]. Across all cases, our proposed model exhibited higher scores, indicating that the proposed G-MLM-based *Speaking* approach offers superior speaker similarity in both scenarios. The results shows speaker similarity is mostly controlled by *Speaking* stage ($\mathbf{p}_a$), while linguistic prosody is controlled by semantic token generation ($\mathbf{p}_s$) [14]. For subjective verification, we uploaded samples illustrating the separated control scenario to our demo page.

Table 2: *Comparison of performance between the Token Transducer and the Token Transducer++. The RTF is calculated on RTF8000 GPU.*

Method	RTF	CER	SECS	NLL
Token Transducer	14.35	2.50	**0.518**	1.326
Token Transducer++	**23.38**	**2.34**	0.512	**1.300**

Table 3: *Comparison of SECS between generated samples and the speech prompts. The* $\boldsymbol{p}_{rand}$ *denotes arbitrary samples from the test set, which are used as the standard value.*

Prompt		SECS	
		Kim et al. [14]	Proposed
$\mathbf{p}_{rand}$		0.116	**0.122**
$\mathbf{p}_s = \mathbf{p}_a$		0.462	**0.514**
$\mathbf{p}_s \neq \mathbf{p}_a$	$\mathbf{p}_s$	0.119	**0.128**
	$\mathbf{p}_a$	0.432	**0.481**

4. Conclusions

In this paper, we introduced a two-stage text-to-speech (TTS) system designed to achieve high-fidelity speech synthesis through the utilization of semantic and acoustic tokens. The first stage, termed the *Interpreting* module, effectively processes text and a speech prompt into semantic tokens, ensuring precise pronunciation and alignment. Following this, the *Speaking* module takes over, employing these semantic tokens to generate acoustic tokens that capture the target voice's acoustic attribute (timbre, acoustic condition), significantly enhancing the speech reconstruction process. Experimental results verify that our proposed method outperforms state-of-the-art baselines regarding speech intelligibility, audio quality, and speaker similarity. For future work, we plan to extend our proposed framework to multiple languages and make our model encompass a broader range of speech-generation tasks, including singing voice synthesis. Also, we will enlarge the training dataset to the unlabeled speech data to increase the generalization of *Speaking*.

*https://www.mturk.com

5. References

[1] E. Kharitonov, D. Vincent, Z. Borsos, R. Marinier, S. Girgin, O. Pietquin, M. Sharifi, M. Tagliasacchi, and N. Zeghidour, "Speak, read and prompt: High-fidelity text-to-speech with minimal supervision," *Transactions of the Association for Computational Linguistics*, vol. 11, pp. 1703–1718, 2023.

[2] C. Wang, S. Chen, 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," *arXiv preprint arXiv:2301.02111*, 2023.

[3] Z. Borsos, M. Sharifi, D. Vincent, E. Kharitonov, N. Zeghidour, and M. Tagliasacchi, "Soundstorm: Efficient parallel audio generation," *arXiv preprint arXiv:2305.09636*, 2023.

[4] C. Du, Y. Guo, F. Shen, Z. Liu, Z. Liang, X. Chen, S. Wang, H. Zhang, and K. Yu, "Unicats: A unified context-aware text-to-speech framework with contextual vq-diffusion and vocoding," *arXiv preprint arXiv:2306.07547*, 2023.

[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] 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.

[7] 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.

[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. Défossez, J. Copet, G. Synnaeve, and Y. Adi, "High fidelity neural audio compression," *Transactions on Machine Learning Research*, 2023, featured Certification, Reproducibility Certification.

[10] 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 arXiv:2305.02765*, 2023.

[11] 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.

[12] M. Jeong, M. Kim, J. Y. Lee, and N. S. Kim, "Efficient parallel audio generation using group masked language modeling," *arXiv preprint arXiv:2401.01099*, 2024.

[13] H. Chang, H. Zhang, L. Jiang, C. Liu, and W. T. Freeman, "Maskgit: Masked generative image transformer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 11 315–11 325.

[14] M. Kim, M. Jeong, B. J. Choi, S. Kim, J. Y. Lee, and N. S. Kim, "Utilizing neural transducers for two-stage text-to-speech via semantic token prediction," *arXiv preprint arXiv:2401.01498*, 2024.

[15] C. Du, Y. Guo, H. Wang, Y. Yang, Z. Niu, S. Wang, H. Zhang, X. Chen, and K. Yu, "Vall-t: Decoder-only generative transducer for robust and decoding-controllable text-to-speech," *arXiv preprint arXiv:2401.14321*, 2024.

[16] A. Graves, "Sequence transduction with recurrent neural networks," *in Representation Learning Worksop, ICML*, 2012.

[17] H. F. F. Garcia, P. Seetharaman, R. Kumar, and B. Pardo, "Vampnet: Music generation via masked acoustic token modeling," in *Ismir 2023 Hybrid Conference*, 2023.

[18] 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 *Proc. Interspeech 2020*, 2020, pp. 5036–5040.

[19] K. Shen, Z. Ju, X. Tan, E. Liu, Y. Leng, L. He, T. Qin, sheng zhao, and J. Bian, "Naturalspeech 2: Latent diffusion models are natural and zero-shot speech and singing synthesizers," in *The Twelfth International Conference on Learning Representations*, 2024.

[20] 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," *Interspeech 2019*, 2019.

[21] Z. Zhang, L. Zhou, C. Wang, S. Chen, Y. Wu, S. Liu, Z. Chen, Y. Liu, H. Wang, J. Li *et al.*, "Speak foreign languages with your own voice: Cross-lingual neural codec language modeling," *arXiv preprint arXiv:2303.03926*, 2023.

[22] B. Desplanques, J. Thienpondt, and K. Demuynck, "ECAPA-TDNN: Emphasized Channel Attention, Propagation and Aggregation in TDNN Based Speaker Verification," in *Proc. Interspeech 2020*, 2020, pp. 3830–3834.