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

Robust neural codec language modeling with phoneme position prediction for zero-shot TTS

Chunhui Lu[1], *Xue Wen*[1], *Liming Song*[1], *Junkwang Oh*[2]

[1]Samsung R&D Institute China-Beijing, China
[2]Mobile eXperience Business, Samsung Electronics, Republic of Korea
{chunhui.lu, xue.wen, lm0518.song, jun1.oh}@samsung.com

Abstract

Though large language models (LLMs) based methods have achieved remarkable progress in zero-shot text-to-speech (TTS) synthesis, they suffer from robustness issues including mispronunciation, word skipping, and word repeating. To address these robustness challenges, we propose incorporating phoneme position prediction into the LLM-based TTS model. More concretely, given an input phoneme sequence as condition, our model autoregressively predicts acoustic codes and their corresponding phoneme positions within the input sequence synchronously. This mechanism ensures accurate and complete alignment between acoustic codes and phonemes. Experimental results demonstrate that our system significantly reduces phoneme skipping/repetition errors compared to strong baselines, achieving a 52.7% relative reduction in character error rate while maintaining comparable performance in zero-shot TTS evaluations.

Index Terms: robustness, phoneme position prediction, zero-shot TTS, codec language modeling

1. Introduction

Large language models (LLMs) have exhibited impressive in-context learning abilities [1, 2], as evidenced by their capacity to handle unseen tasks via prompting in a zero-shot manner. Inspired by these successes, recent studies [3–8] have extended LLM applications to text-to-speech (TTS) systems by representing speech through discrete acoustic codes. These LLM-based models show remarkable progress in zero-shot TTS, successfully generating high-quality personalized speech for unseen speakers using a brief audio prompt. However, these models suffer from robustness issues. In transformer-based models, the alignment between generated speech and input phonemes is implicitly learned through self-attention mechanism. This indirect learning approach can produce incorrect alignments, manifesting as word skipping, repetition, and mispronunciations in generated speech.

To alleviate this challenge, RALL-E [7] employs explicit duration modeling, in which phoneme durations are first predicted as intermediate conditions for acoustic code prediction; VALL-E R [6] strengthens alignment by combining phoneme identity prediction and monotonic decoding strategy. Unlike these works, we propose a simple yet effective method to enhance the alignment between acoustic codes and phonemes while keeping the inference step unchanged and without using any constraints in decoding. As illustrated in Figure 1, our approach synchronously predicts both acoustic codes and their corresponding phoneme positions within the input sequence. We then utilize this combined information as conditions for subsequent predictions. This approach enables each acoustic code to accurately attend to its target phoneme, and the enforced monotonic progression of positional indices ensures complete phoneme coverage within the attention framework.

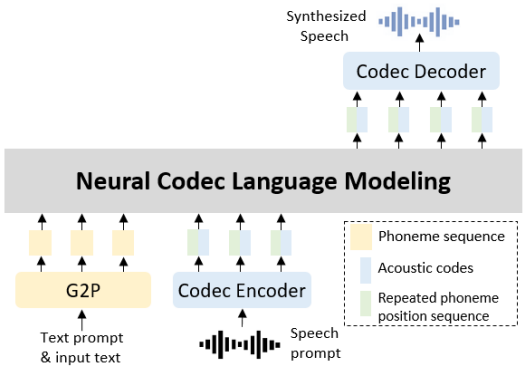


Figure 1: *The overview of our proposed method.*

2. Related work

2.1. LLM based zero-shot TTS

Text-based LLMs have exhibited impressive in-context learning capabilities through prompting. Inspired by this, VALL-E [3] pioneered the application of language modeling techniques to TTS systems by encoding speech into discrete acoustic codes using an off-the-shelf neural audio codec model. Through large-scale training, VALL-E achieved significant zero-shot TTS capability requiring only a brief speech sample as prompt. Building upon VALL-E's success, subsequent research has introduced various improvements. These enhancements include extending the application scenarios [4], improving data efficiency [9, 10], and optimizing the computational efficiency [11–13]. In contrast to these endeavors, our work primarily concentrates on the robustness of the model.

2.2. Robustness of autoregressive TTS models

In autoregressive (AR) TTS models, robustness issues predominantly stem from inaccurate alignments learned through attention mechanisms. To address this challenge, traditional encoder-decoder based TTS systems primarily adopt two methodological paradigms. The first paradigm enhances attention mechanisms by enforcing alignment constraints between text and speech sequences [14], including ensuring the locality, monotonicity, and completeness of the alignments [15–17]. The second paradigm replaces implicit attention-derived durations with explicit duration modeling [18–20].

Building on similar strategies, recent decoder-only TTS

10.21437/Interspeech.2025-1641

systems have seen robustness improvements at the inference stage: [21] employs attention-constrained decoding while [8] adopts repetition-aware sampling, both maintaining the original model architectures. Beyond these approaches, other works [5–7, 22] enhance robustness through implicit or explicit phoneme duration modeling. Among these, VALL-E R shares the most similarities with our approach, as both predict speech codes and their corresponding phoneme information synchronously. The crucial distinction resides in prediction targets: VALL-E R predicts phoneme identities whereas our work predicts phoneme positions.

3. Method

We conduct our experiments based on VALL-E. In subsequent sections, we first provide a brief review of VALL-E, then present detailed exposition of our proposed method. Notably, while implementing our method using VALL-E as foundation, the design maintains compatibility with any decoder-only AR-TTS models.

3.1. Preliminary: VALL-E

Let the target text-speech pair be denoted as $\{\mathbf{x}, \mathbf{y}\}$, and the prompt text-speech pair from same speaker as $\{\widetilde{\mathbf{x}}, \widetilde{\mathbf{y}}\}$. A pre-trained neural codec model encodes the speech $\mathbf{y}$ and $\widetilde{\mathbf{y}}$ into discrete acoustic codes, denoted as $\mathbf{C}_T^N = \mathrm{Codec}(\mathbf{y})$ and $\widetilde{\mathbf{C}}_{T'}^N$, respectively. Here, T and T' are the total number of time steps for each matrix, and N is the number of quantizers in the codec model. Concurrently, the text $\mathbf{x}$ and $\widetilde{\mathbf{x}}$ are converted into phoneme sequences $\mathbf{p} = \{p_1, p_2, \ldots, p_L\}$ and $\widetilde{\mathbf{p}}$, respectively. It should be noted that the text can also be represented as sequences of syllables or text tokens.

In general, VALL-E is a decoder-only TTS system that regards zero-shot TTS as a codec language modeling task. It is trained to generate acoustic codes $\mathbf{C}$ conditioned on the phonemes $\mathbf{p}$ and acoustic prompt $\widetilde{\mathbf{C}}$ with the optimization objective of maximizing the probability $\mathbb{P}(\mathbf{C} \mid \mathbf{p}, \widetilde{\mathbf{C}})$. During inference, VALL-E takes input phonemes $\mathbf{p}$, the phonemes $\widetilde{\mathbf{p}}$ and acoustic prompt $\widetilde{\mathbf{C}}$ of enrollment speech as input, and generates target acoustic codes that match both the content of $\mathbf{p}$ and the speaker's voice of $\widetilde{\mathbf{C}}$. These generated codes are subsequently reconstructed into waveform by the codec's decoder.

More specifically, VALL-E incorporates two Transformers to hierarchically predict the acoustic codes. One AR transformer takes $\mathbf{p}$ as input to sequentially predict the target acoustic codes of first quantizer $\mathbf{c}_{1:T}^1$ and aims to maximize the following distribution:

$$\mathbb{P}(\mathbf{c}_{1:T}^1 \mid \mathbf{p}; \theta_{AR}) = \prod_{t=1}^{T} \mathbb{P}(c_t^1 \mid \mathbf{p}, \mathbf{c}_{<t}^1; \theta_{AR}) \tag{1}$$

where θ_{AR} is the trainable parameters of the AR model. In AR model training, there is no explicit acoustic prompt, instead, any prefix sequence $\mathbf{c}_{<t}^1$ is treated as a prompt for the subsequent part $\mathbf{c}_{\geq t}^1$. To align with this training approach, during inference, $\widetilde{\mathbf{p}}$ is prepended to $\mathbf{p}$ to form the phoneme prompt and the enrollment acoustic codes from first quantizer $\widetilde{\mathbf{c}}^1$ are used as acoustic prefix.

One non-autoregressive (NAR) transformer predicts the codes of the remaining quantizers. It simultaneously predicts all target acoustic codes $\mathbf{c}_{1:T}^j$ from the j-th quantizer, using $\mathbf{p}$, $\widetilde{\mathbf{C}}$, and all codes from previous quantizers $\mathbf{c}_{1:T}^{<j}$ as conditions

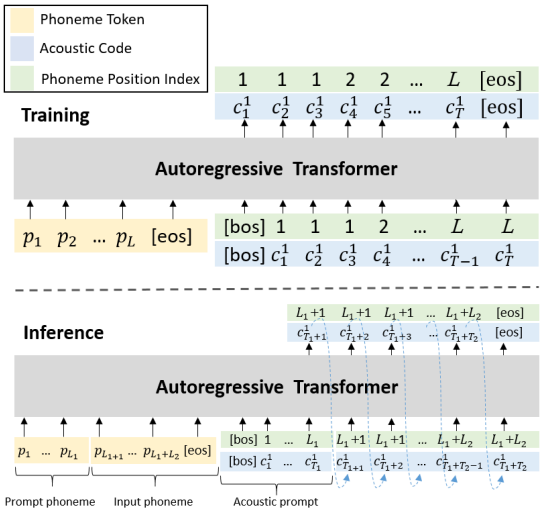


Figure 2: *Training and inference process of our proposed AR model, acoustic codes and its corresponding phoneme positions are predicted synchronously.*

and maximizes the following distribution:

$$\mathbb{P}(\mathbf{c}_{1:T}^{2:N} \mid \mathbf{p}, \widetilde{\mathbf{C}}; \theta_{NAR}) = \prod_{j=2}^{N} \mathbb{P}(\mathbf{c}_{1:T}^{j} \mid \mathbf{p}, \mathbf{c}_{1:T}^{<j}, \widetilde{\mathbf{C}}; \theta_{NAR}) \tag{2}$$

where θ_{NAR} is the trainable parameters of the NAR model.

3.2. Enhancing alignment with phoneme position prediction

In our proposed method, we primarily focus on enhancing the robustness of the AR transformer while leaving the NAR transformer unchanged. We propose incorporating phoneme position prediction into the AR model to strengthen the alignment between acoustic codes and phonemes. To achieve this, we first derive the alignment $\mathbf{d} = \{d_1, \ldots, d_L\}$ between phonemes and acoustic codes, where d_i represents the duration of i-th phoneme, and $\sum_{i=1}^{L} d_i = T$. Subsequently, we construct the repeated phoneme position sequence **pos** by repeating each position $i \in [1, L]$ d_i times. As shown in the upper section of Figure 2, our model is then trained to synchronously predict both the acoustic codes $\mathbf{c}_{1:T}^1$ and their corresponding phoneme positions $\mathbf{pos}_{1:T}$ using teacher forcing, conditioned on the phoneme sequence $\mathbf{p}$. The model is optimized by maximizing the following probability:

$$\begin{aligned} &p(\mathbf{c}_{1:T}^1, \mathbf{pos}_{1:T} \mid \mathbf{p}; \theta_{AR}) \\ &= \prod_{t=1}^{T} p(c_t^1, pos_t \mid \mathbf{p}, \mathbf{c}_{<t}^1, \mathbf{pos}_{<t}; \theta_{AR}) \end{aligned} \tag{3}$$

In practice, the model predicts c_t^1 and pos_t with two separate heads, and their embeddings are summed up and fed to the model for the prediction of the next step. This mechanism establishes explicit correspondence between acoustic codes and their target phonemes via assigned positional indices, and guarantees precise alignment through monotonically increasing indices.

3.3. Zero-shot TTS inference

We implement zero-shot TTS through prompting during inference, following VALL-E but differing in two key aspects to accommodate our AR model modifications, as depicted in the

lower portion of Figure 2. First, given an enrollment speech sample from an unseen speaker as prompt, in addition to the phoneme sequence and acoustic code matrix, phoneme durations should also be included. These durations are essential for constructing the phoneme position sequence that functions as our model's prompt input. Second, in the decoding process, we employ two distinct strategies corresponding to each prediction target. For phoneme positions, we use greedy decoding to choose the position with the highest probability, while for acoustic codes, we apply nucleus sampling [23] with a predefined top-p value of 0.98.

4. Experiment

4.1. Experimental setup

We use the Standard subset of WenetSpeech4TTS [24] for training, comprising 4,056 hours of Mandarin speech data across multiple domains, each sample accompanied by text transcriptions and character-level duration annotations. To ensure speaker consistency, we applied an open-source speaker diarization model[1] [25, 26] to detect and filter speech samples containing multiple speakers. This preprocessing yielded a refined 3,453-hour dataset for model training.

The acoustic codes are extracted using UniAudio's [27] codec model[2], which employs three quantizers under a 16 kHz sampling rate with 20 ms hop size. To construct the target repeated position sequence, we directly use character durations provided in the dataset. Additionally, we use a proprietary G2P toolkit to convert text into syllable-level pinyin sequences that align with the character-level duration annotations.

Both the AR model and the NAR model have the same transformer architecture with 12 layers, 16 attention heads, an embedding dimension of 1024, a feed-forward layer dimension of 4096, and a dropout of 0.1. The output dimension for acoustic code is 1024 and that for phoneme position is set to the maximum phoneme sequence length in the training data.

We use an open-source implementation of VALL-E[3] as the base code and train each model using 2 NVIDIA A100 40GB GPUs for 10 epochs. We use the ScaledAdam optimizer and the Eden scheduler with a 0.05 base learning rate to train all models.

4.2. Evaluations

In our experiments, we evaluate the following systems on the zero-shot TTS task:

- GT: ground-truth audio reconstructed by UniAudio's codec model.
- VALL-E: Implemented exactly following [3], without robustness enhancement techniques.
- VALL-E R: Following [6], the system synchronously predicts acoustic codes and their corresponding phoneme identities while adopting their monotonic alignment decoding strategy.
- VALL-E R (Greedy): Identical to system VALL-E R, but employing greedy decoding for phoneme prediction.
- VALL-E BOTH: Extending VALL-E by incorporating both phoneme identity and position prediction with greedy decoding in inference.
- OURS: Our proposed method.

All systems, except GT, share the same NAR model but different AR models.

4.2.1. Objective evaluations

We adopt Seed-TTS [28] *test-zh* for objective evaluation, comprising 2,020 meta cases and 400 hard cases. We use Character Error Rate (CER) and Speaker Embedding Cosine Similarity (SECS) to evaluate the robustness and speaker similarity of synthesized samples respectively. To calculate CER, we perform Automatic Speech Recognition (ASR) on the generated audio using the open-source Paraformer-large [29] model[4] and compare the result to the original transcription. For SECS evaluation, we extract speaker embeddings from both synthesized and prompt speech using the SOTA speaker verification model, WavLM-TDNN[5] [30]. We then calculate the cosine similarity between these embeddings.

Table 1: *Objective evaluation results for cross sentence zero-shot TTS on Seed-TTS test-zh testset.*

Method	SECS↑	CER (%)↓		
		All	Meta	Hard
GT	0.931	-	2.33	-
VALL-E	0.926	6.69	3.29	13.78
VALL-E R	0.921	4.52	2.85	8.37
VALL-E R(Greedy)	0.925	8.25	3.77	19.33
VALL-E BOTH	0.923	4.26	3.30	6.26
OURS	0.924	**3.16**	**2.51**	**4.52**

The evaluation results are detailed in Table 1. For speaker similarity, all synthetic methods achieve comparable SECS scores that remain within 1% of the GT, demonstrating effective preservation of voice characteristic across different synthesis approaches. In terms of robustness, our method exhibits substantial improvements across all subsets. The p-values between our system and other VALL-E variants from Wilcoxon signed-rank tests are all smaller than 10^{-5}, which demonstrates that our system performs significantly better than others. On the meta subset, our system achieves a CER of 2.51%, representing a 23.7% reduction compared to VALL-E's 3.29% and approaching GT performance. For the challenging hard subset, our method yields a CER of 4.52%, which is 67.2% lower than VALL-E. These advancements result in a 52.7% overall CER reduction across all test scenarios.

Table 2: *The robustness on Seed-TTS test-zh hard cases.*

Method	CER	Sub	Del	Ins
VALL-E	13.78	5.02	8.19	0.58
VALL-E R	8.37	4.95	2.99	0.43
VALL-E R(Greedy)	19.33	5.83	11.77	1.74
VALL-E BOTH	6.07	5.50	**0.49**	**0.08**
OURS	**4.52**	**3.99**	**0.47**	**0.06**

To analyze robustness improvements on challenging cases in detail, we introduce three CER-derived sub-metrics: Substitution (Sub), Deletion (Del), and Insertion (Ins). These metrics

[1] https://huggingface.co/pyannote/speaker-diarization

[2] https://huggingface.co/Dongchao/UniAudio/blob/main/universal_model.zip

[3] https://github.com/lifeiteng/vall-e

[4] https://www.modelscope.cn/models/iic/speech_paraformer-large_asr_nat-zh-cn-16k-common-vocab8404-pytorch

[5] https://huggingface.co/microsoft/wavlm-base-plus-sv3

specifically characterize error patterns in TTS systems: Sub reflects mispronunciations, Del indicates word omission, while Ins captures word repetition or hallucination.

As listed in Table 2, there are two primary error patterns in baseline VALL-E: Sub errors stemming from similar pronunciations in tongue-twister phrases, and severe Del errors caused by alignment failures on repeated characters, words, or phrases. By introducing phoneme identity prediction and a monotonic alignment strategy, VALL-E R effectively improves robustness and reduces the Del error to 2.99%. However, the severe degradation observed in its greedy decoding variant demonstrates the inability to fundamentally resolve alignment confusion in text repetition scenarios through sole reliance on phoneme identity prediction. Compared to VALL-E and VALL-E R, both our method and VALL-E BOTH achieve breakthrough improvements in Del and Ins errors by leveraging predicted phoneme positions to establish explicit acoustic-phoneme alignment, where the monotonically increasing position guides acoustic code attends exclusively to one phoneme and then effectively prevents skipping/repetition errors. Notably, our position-only approach also achieves the lowest Sub errors, as joint phoneme and position prediction in VALL-E BOTH introduces pronunciation errors when predicted phonemes conflict with their positional assignments.

4.2.2. *Subjective evaluations*

For subjective evaluation, we conduct Comparative Mean Opinion Score (CMOS) and Speaker Similarity Mean Opinion Score (SMOS) tests to assess the naturalness and speaker similarity of the synthesized speech, respectively. CMOS measures the comparative naturalness of the synthesized speech against a reference speech from GT, with a scale ranging from -3 (much worse than the reference) to 3 (much better than the reference). SMOS evaluates how closely the synthesized speech resembles the original prompt, using a scale from 1 to 5. In each subjective test, 15 native Mandarin listeners are asked to evaluate 30 samples from each system.

Table 3: *Subjective evaluation results on cross sentence zero-shot TTS.*

Method	CMOS↑	SMOS↑
GT	0.00	4.46 ± 0.11
VALL-E	-0.06	4.28 ± 0.22
VALL-E R	-0.04	4.27 ± 0.20
OURS	0.14	4.38 ± 0.20

The evaluation results, as detailed in Table 3, demonstrate that our system outperforms the other two VALL-E-based systems and validate the effectiveness of our approach. In terms of speaker similarity, our method notably surpasses VALL-E and VALL-E R by approximately 0.1 in SMOS results. Regarding naturalness, our system not only achieves the best result, but also performs slightly better than the GT. It should be noted that the speech prompts are from unprofessional speakers, which sometimes contains undesirable breaks and hesitations that negatively impacted the naturalness of GT.

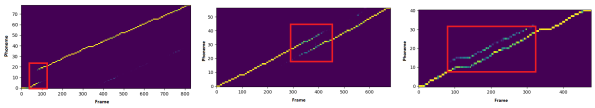
4.3. Alignment analysis

To verify that the robustness improvement of our model is attributed to the enhanced alignment between phonemes and acoustic codes, we first visualized the attention weights of all attention heads across all layers. This visualization allowed us to identify the head exhibiting the most distinct diagonal alignment patterns between phonemes and acoustic codes. We then selected this specific head for detailed analysis of its alignment results on the Seed-TTS test-zh hard samples.

We identified three alignment errors based on the phoneme-speech alignment properties outlined in [14], each violating a different property:

- Phoneme skipping: violates completeness, which requires that each phoneme be attended by at least one acoustic code.
- Phoneme repetition: violates monotonicity, requiring that if phoneme A precedes B in the sequence, their corresponding acoustic codes maintain this order.
- One-to-many alignment: violates locality, stipulating that each acoustic code should correspond to only one phoneme.

We provide concrete examples of these alignment errors in Figure 3.



(a) Phone skipping (b) Phone repetition (c) One-to-many error

Figure 3: *The visualized attention maps of alignment error examples.*

Table 4: *Occurrence times of three alignment errors on Seed-TTS test-zh hard cases.*

Method	Skipping	Repetition	One-to-many
VALL-E	1656	813	259
VALL-E R	978	941	242
OURS	**0**	**0**	**0**

We quantified the occurrences of each alignment error and presented the results in Table 4. We observed that VALL-E and VALL-E R are confused about which phoneme to attend to when encountering multiple identical phonemes, as discussed in Section 4.2.1, leading to these phoneme skipping and repetition alignment errors. In contrast to these systems, our approach completely eliminates all three types of alignment errors, achieving zero error occurrences. This indicates that the proposed phoneme position prediction mechanism effectively addresses alignment issues, thereby enhancing the robustness and accuracy of the TTS system. Speech samples with corresponding alignment visualizations are available online[6].

5. Conclusion

In this paper, we introduce a robust zero-shot TTS model that integrates neural codec language modeling with phoneme position prediction. The position of phonemes corresponding to each acoustic code, along with the code itself, are predicted and then fed back into the model. This additional positional information guides the TTS model to accurately align the acoustic codes with the corresponding phonemes. Experimental results demonstrate that our system effectively eliminates alignment errors, achieving a 52.7% relative reduction in CER compared to VALL-E.

For future work, we will consider extending our method to other languages and applying to other decoder-only models that predict codes extracted from speech, conditioned on tokens derived from text.

[6]https://chunhui-lu.github.io/interspeech25/

6. References

[1] J. Wei, Y. Tay, R. Bommasani, C. Raffel, B. Zoph *et al.*, "Emergent abilities of large language models," *Transactions on Machine Learning Research*, 2022.

[2] A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle *et al.*, "The llama 3 herd of models," *arXiv preprint arXiv:2407.21783*, 2024.

[3] C. Wang, S. Chen, Y. Wu, Z. Zhang, L. Zhou *et al.*, "Neural codec language models are zero-shot text to speech synthesizers," *arXiv preprint arXiv:2301.02111*, 2023.

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

[5] C. Du, Y. Guo, H. Wang, Y. Yang, Z. Niu *et al.*, "VALL-T: Decoder-only generative transducer for robust and decoding-controllable text-to-speech," *arXiv preprint arXiv:2401.14321*, 2024.

[6] B. Han, L. Zhou, S. Liu, S. Chen, L. Meng *et al.*, "VALL-E R: Robust and efficient zero-shot text-to-speech synthesis via monotonic alignment," *arXiv preprint arXiv:2406.07855*, 2024.

[7] D. Xin, X. Tan, K. Shen, Z. Ju, D. Yang *et al.*, "RALL-E: Robust codec language modeling with chain-of-thought prompting for text-to-speech synthesis," *arXiv preprint arXiv:2404.03204*, 2024.

[8] S. Chen, S. Liu, L. Zhou, Y. Liu, X. Tan *et al.*, "VALL-E 2: Neural codec language models are human parity zero-shot text to speech synthesizers," *arXiv preprint arXiv:2406.05370*, 2024.

[9] E. Kharitonov, D. Vincent, Z. Borsos, R. Marinier, S. Girgin *et al.*, "Speak, read and prompt: High-fidelity text-to-speech with minimal supervision," *ACL*, vol. 11, pp. 1703–1718, 2023.

[10] R. Huang, C. Zhang, Y. Wang, D. Yang, J. Tian *et al.*, "Make-A-Voice: Revisiting voice large language models as scalable multilingual and multitask learners," in *ACL*, 2024, pp. 10 929–10 942.

[11] J. Kim, K. Lee, S. Chung, and J. Cho, "CLaM-TTS: Improving neural codec language model for zero-shot text-to-speech," in *The Twelfth International Conference on Learning Representations*, 2024.

[12] L. Meng, L. Zhou, S. Liu, S. Chen, B. Han *et al.*, "Autoregressive speech synthesis without vector quantization," *arXiv preprint arXiv:2407.08551*, 2024.

[13] Y. Song, Z. Chen, X. Wang, Z. Ma, G. Yang *et al.*, "TacoLM: Gated attention equipped codec language model are efficient zero-shot text to speech synthesizers," in *INTERSPEECH*, 2024.

[14] X. Tan, T. Qin, F. Soong, and T.-Y. Liu, "A survey on neural speech synthesis," *arXiv preprint arXiv:2106.15561*, 2021.

[15] J.-X. Zhang, Z.-H. Ling, and L.-R. Dai, "Forward attention in sequence-to-sequence acoustic modeling for speech synthesis," in *ICASSP*, 2018, pp. 4789–4793.

[16] E. Battenberg, R. Skerry-Ryan, S. Mariooryad, D. Stanton, D. Kao *et al.*, "Location-relative attention mechanisms for robust long-form speech synthesis," in *ICASSP*, 2020, pp. 6194–6198.

[17] M. He, Y. Deng, and L. He, "Robust sequence-to-sequence acoustic modeling with stepwise monotonic attention for neural tts," in *INTERSPEECH*, 2019, pp. 1293–1297.

[18] J. Shen, Y. Jia, M. Chrzanowski, Y. Zhang, I. Elias *et al.*, "Non-attentive tacotron: Robust and controllable neural TTS synthesis including unsupervised duration modeling," *arXiv preprint arXiv:2010.04301*, 2020.

[19] I. Elias, H. Zen, J. Shen, Y. Zhang, Y. Jia *et al.*, "Parallel tacotron: Non-autoregressive and controllable TTS," in *ICASSP*, 2021, pp. 5709–5713.

[20] C. Yu, H. Lu, N. Hu, M. Yu, C. Weng *et al.*, "DurIAN: Duration informed attention network for speech synthesis." in *Interspeech*, 2020, pp. 2027–2031.

[21] H. Wang, C. Du, Y. Guo, S. Wang, X. Chen, and K. Yu, "Attention-constrained inference for robust decoder-only text-to-speech," *arXiv preprint arXiv:2404.19723*, 2024.

[22] Y. Song, Z. Chen, X. Wang, Z. Ma, and X. Chen, "ELLA-V: Stable neural codec language modeling with alignment-guided sequence reordering," *arXiv preprint arXiv:2401.07333*, 2024.

[23] A. Holtzman, J. Buys, L. Du, M. Forbes, and Y. Choi, "The curious case of neural text degeneration," in *ICLR*, 2020.

[24] L. Ma, D. Guo, K. Song, Y. Jiang, S. Wang *et al.*, "WenetSpeech4TTS: A 12,800-hour mandarin TTS corpus for large speech generation model benchmark," in *INTERSPEECH*, 2024.

[25] H. Bredin, "pyannote.audio 2.1 speaker diarization pipeline: principle, benchmark, and recipe," in *INTERSPEECH*, 2023, pp. 1983–1987.

[26] A. Plaquet and H. Bredin, "Powerset multi-class cross entropy loss for neural speaker diarization," in *INTERSPEECH*, 2023, pp. 3222–3226.

[27] D. Yang, J. Tian, X. Tan, R. Huang, S. Liu *et al.*, "UniAudio: Towards universal audio generation with large language models," in *Forty-first International Conference on Machine Learning*, 2024.

[28] 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.

[29] Z. Gao, S. Zhang, I. McLoughlin, and Z. Yan, "Paraformer: Fast and accurate parallel transformer for non-autoregressive end-to-end speech recognition," in *INTERSPEECH*, 2022, pp. 2063–2067.

[30] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu *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.