DECODING ORDER MATTERS IN AUTOREGRESSIVE SPEECH SYNTHESIS

Minghui Zhao, Anton Ragni

School of Computer Science, University of Sheffield, 211 Portobello, Sheffield S1 4DP, UK

ABSTRACT

Autoregressive speech synthesis often adopts a left-to-right order, yet generation order is a modelling choice. We investigate decoding order through masked diffusion framework, which progressively unmasks positions and allows arbitrary decoding orders during training and inference. By interpolating between identity and random permutations, we show that randomness in decoding order affects speech quality. We further compare fixed strategies, such as `l2r` and `r2l` with adaptive ones, such as Top-K, finding that fixed-order decoding, including the dominating left-to-right approach, is suboptimal, while adaptive decoding yields better performance. Finally, since masked diffusion requires discrete inputs, we quantise acoustic representations and find that even 1-bit quantisation can support reasonably high-quality speech.

***Index Terms*—** speech synthesis, discrete diffusion model, order-agnostic autoregressive decoding

1. INTRODUCTION

Autoregressive generation has long been central to speech synthesis, from earlier systems such as Wavenet [1] and Tacotron 2 [2] to more recent approaches that model discretised acoustic features with a language model (e.g., [3, 4, 5]). In these systems, speech is generated sequentially, with each frame or sample conditioned on previously produced outputs, hardly ever, if ever, not in a left-to-right order that mirrors the flow of natural speech.

From a modelling perspective, however, left-to-right generation is not necessarily optimal. Speech exhibits dependencies that extend beyond a simple causal chain: pauses and emphasis often depend on global context, while coarticulation reflects interactions between both past and future phones. Even when the model is conditioned on the full utterance, for example through the phone sequence, the decoding order can influence how effectively these dependencies are captured. Exploring alternatives to left-to-right generation is therefore important not only for assessing the widely adopted and unchallenged convention, but also for improving synthesis quality.

Viewing decoding order as a modelling choice, consider an autoregressive approximation to the ground-truth distribution $p(\boldsymbol{y}_{1:T})$ for a length-T sequence. Let S_T denote the set of all permutations of $\{1, \cdots, T\}$. For any $\boldsymbol{\sigma} \in S_T$, the chain rule yields

$$p(\boldsymbol{y}_{1:T} \mid \boldsymbol{\sigma}) = \prod_{t=1}^{T} p(\boldsymbol{y}_{\boldsymbol{\sigma}(t)} \mid \boldsymbol{y}_{\boldsymbol{\sigma}(<t)}), \tag{1}$$

where $\boldsymbol{\sigma}(< t) := \{\boldsymbol{\sigma}(1), \ldots, \boldsymbol{\sigma}(t-1)\}$, t indexes the factorisation step and $\boldsymbol{\sigma}(t)$ is the position in the sequence, which typically equals to t but in this work could be any value in range $[1, T]$. Because different orders expose different contexts, the resulting factorisation can vary in how well it approximates $p(\boldsymbol{y}_{1:T})$. Left-to-right is the special case $\boldsymbol{\sigma}^{l2r}(t) = t$, but nothing in (1) requires this choice. Moreover, decoding order can be adaptive, with $\boldsymbol{\sigma}$ chosen dynamically. In principle, for fixed orders, separate autoregressive models could be trained under different fixed orders, but this is inefficient, since there are $T!$ such orders, and it prevents direct comparison in the same framework.

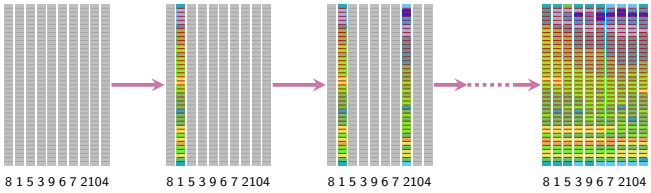

Fig. 1. Intermediate mel-spectrograms shown from left to right as generation progresses in a random order. The number beneath each frame indicates its order in generation and the rightmost frames are the final output.

The masked diffusion model (MDM) [6, 7, 8] is an interesting, recently proposed, framework that supports arbitrary decoding orders $\boldsymbol{\sigma} \in S_T$. During training, the model predicts randomly masked tokens from the visible ones, so in expectation every token is equally likely to be masked and reconstructed. This *order-agnostic* training ensures that no particular order of information revealing is favoured. At inference (Figure 1), we can choose any generation order $\boldsymbol{\sigma}$ and unmask each position conditioned on what has already be revealed, which allows us to assess the effect that any order has on the quality of the generated result. Recent experiments on reasoning and vision tasks show that different generation orders can produce different outcomes [8, 9], suggesting that the choice of $\boldsymbol{\sigma}$ shapes how dependencies are captured and motivating the exploration of non-left-to-right factorisations.

This paper investigates how decoding order shapes autoregressive speech synthesis. We show that randomness in order affects speech quality, fixed left-to-right decoding is suboptimal, and adaptive strategies perform better. Building on this, we propose a duration-guided decoding scheme. We further extend decoding from single to multiple frames, and establish the decoding schedule as a critical modelling choice.

2. METHOD

Autoregressive sequence decoding can be characterised by two design choices, the order in which frames are generated and the update size k, i.e., how many elements are decoded at each step. While many models impose a fixed schedule (e.g., left-to-right, one by one), the Masked Diffusion Model (MDM) predicts probabilities for

This work was supported by the UKRI AI Centre for Doctoral Training in Speech and Language Technologies (SLT) and their Applications funded by UK Research and Innovation [grant number EP/S023062/1].

all positions at each step, allowing the decoding schedule to be freely specified along both dimensions. To study the effect of order, we constrain the model to generate one position at a time ($k = 1$ for every step) without replacement, making it directly comparable to conventional autoregressive models, as in [7].

The following sections describe the inference and training procedures. Assume that $\mu \in \mathbb{R}^{n_f \times T}$ is a prior derived from text that has the desired temporal length. Conditioned on $\boldsymbol{\mu}$, we generate $\boldsymbol{y}$ of the same dimensions. In our implementation, generation proceeds along the T dimension, though it could in principle be extended to a flattened dimension of length $n_f \times T$. As MDMs are designed for discrete data type, we need to quantise Mel-spectrograms used as an acoustic representation in this work. We then discuss the decoding strategies implemented in this paper.

2.1. Inference

The inference procedure is outlined in Algorithm 1. At inference, a decoding order $\boldsymbol{\sigma}$ is sampled uniformly from S_T, and generation begins with all frames masked (set as $\mathbf{0}$). For each step $t = 1, \ldots, T$ and position $i \in \{1, \ldots, T\}$, we define the indicator masks

$$\boldsymbol{m}_i = \mathbb{1}[\boldsymbol{\sigma}(i) < t], \qquad \boldsymbol{n}_i = \mathbb{1}[\boldsymbol{\sigma}(i) = t], \tag{2}$$

where $\boldsymbol{m}$ marks previously generated frames and $\boldsymbol{n}$, marks the current position to decode. Conditioned on $\boldsymbol{\mu}$ and the available context $\boldsymbol{m} \odot \boldsymbol{y}$, the model outputs the parameters of a categorical distribution $\mathcal{C}(\boldsymbol{y}|\boldsymbol{f}_\theta(\boldsymbol{m} \odot \boldsymbol{y}, \boldsymbol{\mu}, \boldsymbol{m}))$, which approximates $p(\boldsymbol{y}_{\boldsymbol{\sigma}(t)} \mid \boldsymbol{y}_{\boldsymbol{\sigma}(<t)})$. An update $\boldsymbol{y}'$ is then drawn, with positions sampled independently, and only the frame indicated by $\boldsymbol{n}$ is updated.

Algorithm 1 Order-agnostic Sampling

Require: Conditioning input $\boldsymbol{\mu}$, Network $\boldsymbol{f}_\theta$
Ensure: Sample $\boldsymbol{y}$
  Initialize $\boldsymbol{y} = \mathbf{0}$
  Sample $\boldsymbol{\sigma} \sim \mathcal{U}(S_T)$
  for t in $\{1, \ldots, T\}$ **do**
    $\boldsymbol{m} \leftarrow \big(\mathbb{1}[\boldsymbol{\sigma}(i) < t]\big)_{i=1}^{T}$
    $\boldsymbol{n} \leftarrow \big(\mathbb{1}[\boldsymbol{\sigma}(i) = t]\big)_{i=1}^{T}$
    $\boldsymbol{y}' \sim \mathcal{C}(\boldsymbol{y}|\boldsymbol{f}_\theta(\boldsymbol{m} \odot \boldsymbol{y}, \boldsymbol{\mu}, \boldsymbol{m}))$
    $\boldsymbol{y} \leftarrow (\mathbf{1} - \boldsymbol{n}) \odot \boldsymbol{y} + \boldsymbol{n} \odot \boldsymbol{y}'$
  end for

2.2. Order-agnostic training

Algorithm 2 provides the training procedure. During training, the model learns to reconstruct masked frames $\boldsymbol{y}_{\boldsymbol{\sigma}(\geq t)}$ from the observed ones $\boldsymbol{y}_{\boldsymbol{\sigma}(<t)}$. The objective is the evidence lower bound (ELBO) on the log-likelihood [7]:

$$\log p(\boldsymbol{y}) \geq d \cdot \mathbb{E}_{t,\boldsymbol{\sigma}} \left[\frac{1}{d - t + 1} \sum_{k \in \boldsymbol{\sigma}(\geq t)} \log p(\boldsymbol{y}_k \mid \boldsymbol{y}_{\boldsymbol{\sigma}(<t)}) \right], \tag{3}$$

where $t \sim \mathcal{U}(1, \ldots, T)$ and $\boldsymbol{\sigma}$ is uniformly sampled from $\mathcal{U}(S_T)$. Naively optimising the chain rule likelihood (Eq. (1)) requires full sequential autoregression over all positions, which is computationally expensive. The formulation in Eq.(3) instead permits computing all masked terms k in parallel for a sampled pair $(t, \boldsymbol{\sigma})$, yielding an efficient and order-agnostic training objective. To approximate the expectation, we use a single-sample Monte Carlo estimate per step.

Algorithm 2 Order-agnostic Training

Require: Data point $\boldsymbol{y}$, Conditioning input $\boldsymbol{\mu}$, Network $\boldsymbol{f}_\theta$
Ensure: $\mathcal{L}$
  Sample $t \sim \mathcal{U}(1, \ldots, T)$
  Sample $\boldsymbol{\sigma} \sim \mathcal{U}(S_T)$
  Compute $\boldsymbol{m} \leftarrow (\boldsymbol{\sigma} < t)$
  $\boldsymbol{l} \leftarrow (\mathbf{1} - \boldsymbol{m}) \odot \log \mathcal{C}(\boldsymbol{y}|\boldsymbol{f}_\theta(\boldsymbol{m} \odot \boldsymbol{y}, \boldsymbol{\mu}, \boldsymbol{m}))$
  $\mathcal{L}_t \leftarrow \frac{1}{T-t+1}\text{sum}(\boldsymbol{l})$
  $\mathcal{L} \leftarrow T \cdot \mathcal{L}_t$

2.3. Quantisation

Unlike previous works that rely on learned speech tokens (e.g., [3, 4]), we apply a single linear quantiser shared across all frequency bins. Specifically, for a value $y \in [a, b]$, the quantised index $\hat{y}$ is given by:

$$\hat{y} = \text{round}\left(\frac{y - a}{b - a} \cdot (Q - 1) \right), \quad \hat{y} \in \{0, 1, \ldots, Q - 1\}, \tag{4}$$

where Q denotes the number of quantisation bins. Since the model predicts whole frames, with each of the n_f bins sampled from its own distribution over Q values, a frame can be viewed as a token from a vocabulary of size Q^{n_f}. This space is extremely large even for $Q = 2$ with $n_f = 80$.

2.4. Decoding strategies

We refer to the procedure in Algorithm 1 as the `default` decoding that applies uniformly random ordering. We refer to fixed orders left-to-right as `l2r` and right-to-left `r2l`.

2.4.1. Controlling order stochasticity

We investigated the effect of order randomness by varying the degree of stochasticity. Startingfrom $\boldsymbol{\sigma}^{l2r}$, we introduce randomness by applying a sequence of swaps between two randomly chosen positions. Theoretically, performing $T \log T$ swaps yields a distribution that is nearly uniform [8]. We control the number of swaps as a coarse measure of randomness by scaling $T \log T$ with a factor $\beta \in (0, 1)$, yielding orders close to`l2r` as $\beta \to 0$ and to `default` as $\beta \to 1$.

2.4.2. Top-K probability

At step t, for each undecoded position $k \in \boldsymbol{\sigma}(\geq t)$ we compute a confidence score

$$s_k(t) = \sum_{i=1}^{n_f} \max_{j \in \{0, \ldots, Q-1\}} \log p(\boldsymbol{y}_{k,i} = j \mid \boldsymbol{y}_{\boldsymbol{\sigma}(<t)}), \tag{5}$$

which sums the maximum log-probabilities across all n_f bins. The next position is chosen as

$$k_t^\star = \arg \max_{k \in \boldsymbol{\sigma}(\geq t)} s_k(t), \tag{6}$$

so the t-th entry of the decoding order is updated $\boldsymbol{\sigma}(t) \leftarrow k_t^\star$. The equations present the $K = 1$ case; for $K > 1$ we take the top K indices in Eq. (6) at each step. Note that this strategy only selects positions and not the actual values for each position.

2.4.3. *Duration-guided decoding*

While inspecting the adaptively determined Top-K orders (see Section 4.3), we observe that the model tends to decode contiguous positions: adjacent frames are decoded consecutively. This pattern suggests that the algorithm prefers to decode semantically coherent regions together. Motivated by this, we propose a segment-wise decoding scheme: we use the duration predictor to decided the segments and then select the segment with the highest average confidence score (see Eq. (5)), approximated here by the model's predicted probabilities, which may not be well calibrated. The frames within the chosen segment are then updated one by one in a random order, although adaptive strategies could also be considered.

3. EXPERIMENTS

3.1. Experimental setup

We use LJSpeech dataset [10], a public-domain corpus containing approximately 13, 100 clips (1-10s) of read speech by a single female native English speaker. We adopt the same dataset split as Grad-TTS [11]. The model architecture, based on Grad-TTS[11], comprises a text encoder, a duration predictor and a decoder. The encoder produces a latent text representation, upsampled by predicted durations to yield $\boldsymbol{\mu}$, which serves as the prior for decoding. The decoder is and MDM that we train and assess in a number of ways to investigate different orders.

We apply linear quantisation with $Q = 100$ levels to balance learning simplicity and reconstruction fidelity, unless stated otherwise. For each time-frequency bin, the decoder predicts parameters for a mixture of 5 logistic components, sampled independently across Mel bins. At inference, the mixture component is chosen via the Gumbel-Max trick with temperature t_1 and the bin value is then drawn from the selected logistic distribution with temperature t_2. HiFi-GAN [12] is used as the vocoder.

3.2. Evaluation metrics

We evaluate with Mel-Cepstral Distortion (MCD), $\log F_0$, UTMOSv2 [13], and Mean Opinion Score (MOS), all on 50 sampled test-split audios. MCD and $\log F_0$ are computed against vocoded references, while MOS was rated by 10 Amazon Mechanical Turk Master Workers per audio, self-reported as native English speakers.

4. RESULTS

We first justify the choice of Q, then examine the effect of randomness, the impact of decoding order, and finally the role of update size K in Top-K decoding.

4.1. Effect of discretisation on speech quality

We conducted preliminary experiments using a public HiFi-GAN checkpoint to vocode Mel-spectrograms quantised at different levels. As shown in Figure 2, 10-class quantisation preserves high quality, while even 2-class (1-bit) Mel-spectrograms yield partially intelligible speech with some recognisable words, suggesting that Mel-spectrograms are redundant and HiFi-GAN can reconstruct speech even from heavily quantised inputs. To examine this more directly, we trained and evaluated on 1-bit representations, achieving an average MCD of 4.21, and an average of $\log F0$ of 0.211. Based on these findings, we adopt 100-class quantisation, which simplifies training in the discrete space while preserving speech quality.

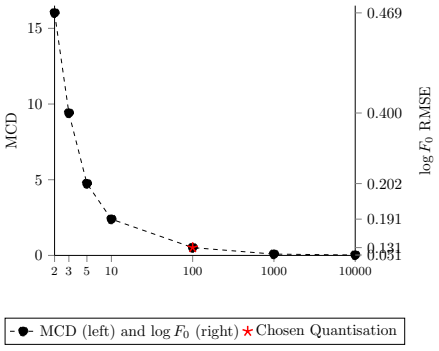

Fig. 2. Evaluation on quantisation levels

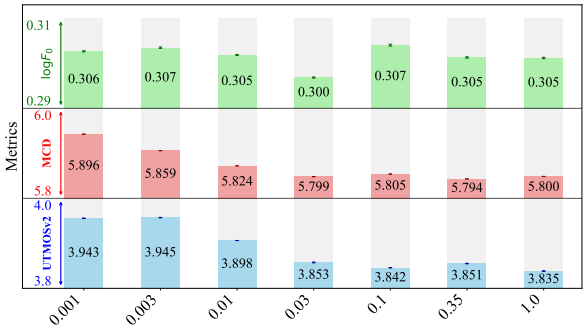

Fig. 3. Evaluation on orders with controlled randomness

4.2. Controlling randomness in decoding order

As described in Section 2.4.1, we control decoding-order randomness by varying β. We test seven values: `[0.001, 0.003, 0.01, 0.03, 0.1, 0.35, 1.0]`. Small β values approximate `l2r`, while larger values approach the `default` strategy. Each setting is run five times and statistics are averaged across runs. Results are shown in Figure 3. As randomness increases, MCD improves while UTMOS degrades. $\log F_0$ shows a different pattern: it reaches its lowest value at intermediate β, where the order is neither strictly sequential nor fully random but still preserves local cluster of consecutive frames. This suggests that pitch is best preserved with partial ordering. Overall, these results highlight that different metrics respond differently to decoding-order randomness.

4.3. The impact of decoding order

We re-train Grad-TTS on full utterances, since the original model is trained on segments. As baselines, we use two decoding strategies: fixed 100-step decoding (`gradtts-100`) and length-based decoding (`gradtts-length`), where the number of steps matches the predicted number of frames. Both `top1` and `top1*` follow Section 2.4.2, but `top1` selects the most likely value, while `top1*` samples from the distribution.

The automatic evaluation metrics are reported in Figure 4. Overall, duration-based decoding performs best, yielding low MCD and $\log F_0$ alongside relatively high UTMOS, with `top1*` performing comparably. Among sequential strategies, `r2l` outperforms `l2r` in UTMOS and MCD, with similar $\log F_0$, demonstrating that left-to-right decoding is not optimal. The Grad-TTS baselines achieve the highest UTMOS among all models but suffer from high MCD, with `gradtts-length` also showing a significantly

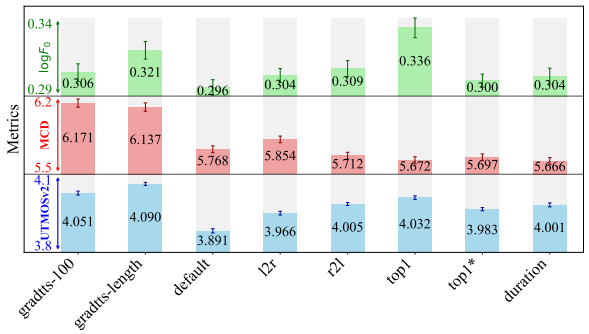

Fig. 4. Evaluation results for single-frame decoding strategies

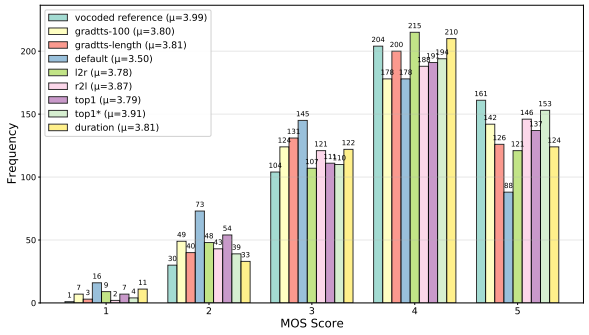

Fig. 5. Breakdown of MOS scores

worse $\log F_0$. `top1` performs similar to `top1*` in UTMOS and MCD, but `top1*` is clearly better in $\log F_0$. MOS results shows a similar trend to the automatic metrics (Figure 5). `top1*` ranks highest after the vocoded reference. `r2l` again surpasses `l2r`. The Grad-TTS baselines achieve the highest scores on automatic metrics but rank lower in subjective evaluations, which may be related to their higher MCD. The duration-based strategy is comparable to the Grad-TTS baselines. Finally, `l2r` and `top1` yield relatively low MOS, but are still higher than the `default` setting. Across metrics, the adaptive strategies `top1*` and duration-based decoding performed best, while deterministic `top1` lagged behind, likely due to over-smoothed outputs from always selecting the most probable value.

4.4. Adaptive decoding with Top$K$

We then evaluated the adaptive Top-K decoding strategy with varying K, assigning each time-frequency bin its most likely value. Unlike previous experiments, here K frames are updated simultaneously at each step. Results show that increasing K improves MCD and $\log F_0$ but reduces UTMOS (Figure 6). We attribute the MCD and $\log F_0$ gains to decoding multiple frames with shared context, which enhances spectral and pitch consistency, while the drop in UTMOS reflects diminished naturalness. Considering order and update size, an optimal decoding schedule may be learned with reinforcement learning, rather than fixed heuristics.

5. RELATED WORK

5.1. Combining diffusion and autoregressive models

The notion of autoregression in this paper differs from recent efforts to make diffusion models semi-autoregressive. For example, [14] uses a block-based approach with left-to-right autoregression across blocks and parallel prediction within blocks. Similar approaches have been explored in speech synthesis, notably in models such as DiffAR [15], ARDiT [16] and DiTAR [17]. In this paper, we integrate the iterative refinement of diffusion models with autoregressive generation, following [7, 8].

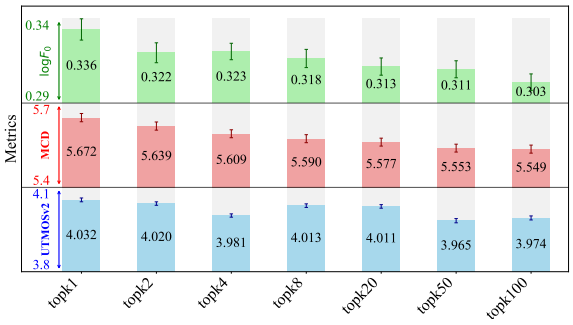

Fig. 6. Evaluation results for Top$K$ decoding

5.2. Discrete speech tokens

Discrete speech tokens are usually obtained via vector quantisation (e.g., VQ-VAE [18]) or clustering [19]. In contrast, we apply uniform scalar quantisation directly to Mel-spectrograms and find that an off-the-shelf HiFi-GAN can decode the dequantised values without retraining, remarkably even at the extreme 1-bit setting. For images, Mentzer *et al.* proposed FSQ-VQ [20], which also uses scalar quantisation but maps the resulting vector to a single token ID for downstream modelling. In our setting, such per-frame tokenisation would cause a combinatorial vocabulary explosion. Instead, we sample bins independently per frame, which likely contributes to degraded quality at higher sampling temperatures, as within-frame frequency correlations are not modelled. Future work could reduce the number of bins per frame to make FSQ-style tokenisation tractable or incorporate joint sampling across bins.

5.3. Vocoders

Vocoder input have evolved from hand-crafted features (e.g. F_0 in statistical parametric vocoders [21]) to Mel-spectrograms in neural vocoders [1, 12], and more recently to discrete latent tokens in codec models [3, 4, 5]. Quantisation experiments suggest vocoders need not preserve exact mel values, but only their relative distribution. Whether this holds true for other neural vocoders remain unexplored.

6. CONCLUSION

In this paper, we examine the role of decoding order in autoregressive speech synthesis. We show that left-to-right order can be suboptimal in speech synthesis, despite its universal adoption. Our results show that adaptive orders generally yield better performance, though identifying an optimal update schedule requires further exploration. We also found that the degree of decoding-order randomness affects synthesis quality. Finally, we demonstrate that speech tokens can be obtained directly through simple quantisation without training, and that these tokens integrate effectively with standard vocoders such as HiFi-GAN.

7. REFERENCES

[1] Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew W. Senior, and Koray Kavukcuoglu, "Wavenet: A generative model for raw audio," *CoRR*, vol. abs/1609.03499, 2016.

[2] Jonathan Shen, Ruoming Pang, Ron J. Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, R. J. Skerry-Ryan, Rif A. Saurous, Yannis Agiomyrgiannakis, and Yonghui Wu, "Natural TTS synthesis by conditioning wavenet on mel spectrogram predictions," *CoRR*, vol. abs/1712.05884, 2017.

[3] Neil Zeghidour, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi, "Soundstream: An end-to-end neural audio codec," 2021.

[4] Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi, "High fidelity neural audio compression," 2022.

[5] Chengyi Wang, Sanyuan Chen, Yu Wu, Ziqiang Zhang, Long Zhou, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, Lei He, Sheng Zhao, and Furu Wei, "Neural codec language models are zero-shot text to speech synthesizers," 2023.

[6] Subham Sekhar Sahoo, Marianne Arriola, Aaron Gokaslan, Edgar Mariano Marroquin, Alexander M Rush, Yair Schiff, Justin T Chiu, and Volodymyr Kuleshov, "Simple and effective masked diffusion language models," in *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[7] Emiel Hoogeboom, Alexey A. Gritsenko, Jasmijn Bastings, Ben Poole, Rianne van den Berg, and Tim Salimans, "Autoregressive diffusion models," 2022.

[8] Jaeyeon Kim, Kulin Shah, Vasilis Kontonis, Sham M. Kakade, and Sitan Chen, "Train for the worst, plan for the best: Understanding token ordering in masked diffusions," in *Forty-second International Conference on Machine Learning*, 2025.

[9] Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He, "Autoregressive image generation without vector quantization," in *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[10] Keith Ito and Linda Johnson, "The lj speech dataset," `https://keithito.com/LJ-Speech-Dataset/`, 2017.

[11] Vadim Popov, Ivan Vovk, Vladimir Gogoryan, Tasnima Sadekova, and Mikhail Kudinov, "Grad-tts: A diffusion probabilistic model for text-to-speech," in *Proceedings of the 38th International Conference on Machine Learning*, Marina Meila and Tong Zhang, Eds. 18–24 Jul 2021, vol. 139 of *Proceedings of Machine Learning Research*, pp. 8599–8608, PMLR.

[12] Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae, "Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis," *CoRR*, vol. abs/2010.05646, 2020.

[13] Kaito Baba, Wataru Nakata, Yuki Saito, and Hiroshi Saruwatari, "The t05 system for the VoiceMOS Challenge 2024: Transfer learning from deep image classifier to naturalness MOS prediction of high-quality synthetic speech," in *IEEE Spoken Language Technology Workshop (SLT)*, 2024.

[14] Marianne Arriola, Subham Sekhar Sahoo, Aaron Gokaslan, Zhihan Yang, Zhixuan Qi, Jiaqi Han, Justin T Chiu, and Volodymyr Kuleshov, "Block diffusion: Interpolating between autoregressive and diffusion language models," in *The Thirteenth International Conference on Learning Representations*, 2025.

[15] Roi Benita, Michael Elad, and Joseph Keshet, "Diffar: Denoising diffusion autoregressive model for raw speech waveform generation," 2024.

[16] Zhijun Liu, Shuai Wang, Sho Inoue, Qibing Bai, and Haizhou Li, "Autoregressive diffusion transformer for text-to-speech synthesis," 2024.

[17] Dongya Jia, Zhuo Chen, Jiawei Chen, Chenpeng Du, Jian Wu, Jian Cong, Xiaobin Zhuang, Chumin Li, Zhen Wei, Yuping Wang, and Yuxuan Wang, "DiTAR: Diffusion transformer autoregressive modeling for speech generation," in *Forty-second International Conference on Machine Learning*, 2025.

[18] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu, "Neural discrete representation learning," 2018.

[19] Yiwei Guo, Zhihan Li, Hankun Wang, Bohan Li, Chongtian Shao, Hanglei Zhang, Chenpeng Du, Xie Chen, Shujie Liu, and Kai Yu, "Recent advances in discrete speech tokens: A review," 2025.

[20] Fabian Mentzer, David Minnen, Eirikur Agustsson, and Michael Tschannen, "Finite scalar quantization: VQ-VAE made simple," in *The Twelfth International Conference on Learning Representations*, 2024.

[21] Keiichi Tokuda, Yoshihiko Nankaku, Tomoki Toda, Heiga Zen, Junichi Yamagishi, and Keiichiro Oura, "Speech synthesis based on hidden markov models," *Proceedings of the IEEE*, vol. 101, no. 5, pp. 1234–1252, 2013.