Interspeech 2024
1-5 September 2024, Kos, Greece

SimpleSpeech: Towards Simple and Efficient Text-to-Speech with Scalar Latent Transformer Diffusion Models

Dongchao Yang, Dingdong Wang, Haohan Guo, Xueyuan Chen, Xixin Wu, Helen Meng

The Chinese University of Hong Kong, Hong Kong SAR, China
{dcyang,hmmeng}@se.cuhk.edu.hk

Abstract

In this study, we propose a simple and efficient Non-Autoregressive (NAR) text-to-speech (TTS) system based on diffusion, named SimpleSpeech. Its simpleness shows in three aspects: (1) It can be trained on the speech-only dataset, without any alignment information; (2) It directly takes plain text as input and generates speech through an NAR way; (3) It tries to model speech in a finite and compact latent space, which alleviates the modeling difficulty of diffusion. More specifically, we propose a novel speech codec model (SQ-Codec) with scalar quantization, SQ-Codec effectively maps the complex speech signal into a finite and compact latent space, named scalar latent space. Benefits from SQ-Codec, we apply a novel transformer diffusion model in the scalar latent space of SQ-Codec. We train SimpleSpeech on 4k hours of a speech-only dataset, it shows natural prosody and voice cloning ability. Compared with previous large-scale TTS models, it presents significant speech quality and generation speed improvement. Demos are released.[1]

Index Terms: TTS, Diffusion Models, Speech Codec

1. Introduction

Text-to-speech synthesis (TTS) aims to synthesize intelligible and natural speech given text, which has made great progress in the past years [1, 2]. Most previous TTS systems trained on small-scale high-quality labeled speech datasets. In terms of model training, they rely on a relatively complicated pipeline (*e.g.* prosody prediction, G2P conversion), and fine-grained alignment information (*e.g.* phone-level duration) is needed. However, the recently proposed approaches based on large-scale speech data significantly simplify the TTS system [3–7]. For instance, language model (LM) based TTS, *e.g.*VALL-E [4] uses a pre-trained audio codec model *e.g.* Encodec [8] to map the speech signal into a sequence of discrete tokens, then an auto-regressive language model is trained to translate the phoneme into speech tokens. It can generate more expressive speech, but also is troubled by the slow and unstable inference. To address these issues, Non-autoregressive (NAR) models, *e.g.* NaturalSpeech 2 [9], SoundStorm [10], and VoiceBox [11], are proposed. They have higher inference speed and better stability, but with the cost of 1) relying on phoneme-acoustic alignment information, and 2) a more complicated training process.

In this work, we propose a simple and efficient TTS system, SimpleSpeech, which does not rely on any alignment information and generates high-quality speech in an NAR way. To build such a TTS system, where we meet with the following research problems: (1) how to use the large-scale speech-only dataset to train a TTS model without any alignment information *e.g.* phoneme-level duration; (2) how to design a generative model that can generate high-quality speech in a NAR way. (3) how to solve the duration alignment problem without using a specific duration model when we train the NAR model; The main contributions of this study are summarized as follows:
(1) We demonstrate that large-scale unlabeled speech data can be used to build an NAR TTS system.
(2) We propose a novel generation model, a scalar latent transformer diffusion model, which models the speech data in a finite and compact latent space. More specifically, we propose a speech codec model (SQ-Codec) based on scalar quantization, which maps the complex speech signal into a finite and compact latent space, named scalar latent space. Then we apply the diffusion model in the scalar latent space of SQ-Codec.
(3) We propose to use sentence duration instead of phone-level duration for the NAR-based TTS model. The sentence duration is used to determine the length of the target sequence, then an in-context conditioning strategy is introduced to learn the fine-grained alignment between the condition and target sequence implicitly. Compared to previous works that predict the duration of each phoneme, the sentence duration gives more diversity in the generation process. More importantly, the sentence duration is easy to obtain in both the training and inference stages. Please refer to Section 2.2 finds more details.
(4) Extensive experimental results show the effectiveness of SimpleSpeech. We also conduct a lot of ablation studies to explore the effectiveness of each part in SimpleSpeech.

2. SimpleSpeech

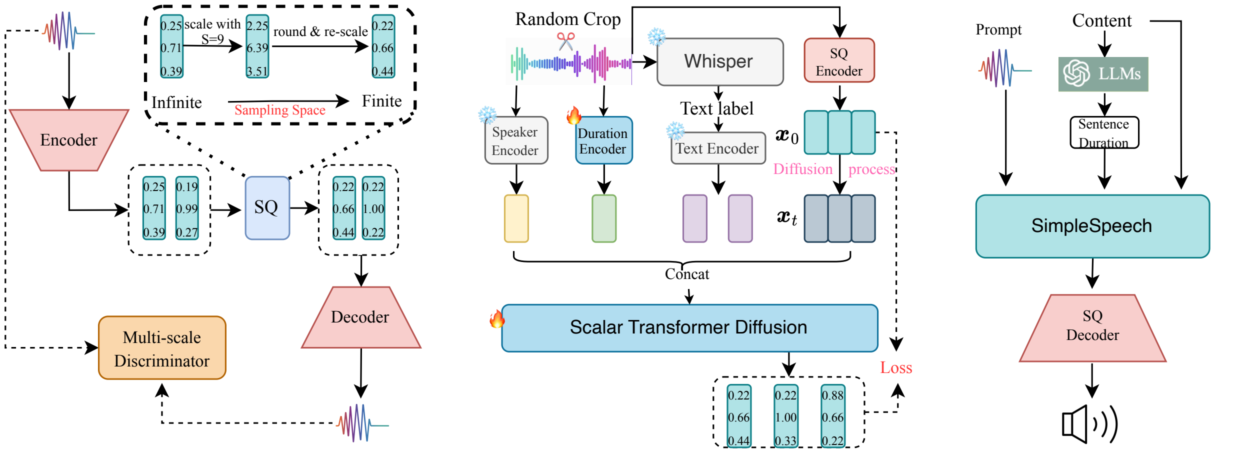
The overall architecture of the SimpleSpeech framework is demonstrated in Figure 1 (b), which mainly consists of two parts: SQ-Codec and Scalar Latent Transformer Diffusion. The detailed design of each part will be introduced in this section.

2.1. Text Encoder and Speaker Encoder

This work explores to use the large-scale speech-only datasets to train a TTS system. The first step is to obtain the text label for these speech samples. We propose to use the open available ASR model to get transcripts, Whisper-base model [12] is used in this study. To take advantage of the large language model and simply the traditional TTS frontend, we use a pre-trained language model to extract the textual representations. Then we directly take these textual representations as the conditional information for TTS. To realize zero-shot voice cloning, we use the 1st layer of XLSR-53 [13] to extract global embedding to represent the speaker timbre.

[1] https://simplespeech.github.io/simplespeechDemo. This work is partially supported by the Centre for Perceptual and Interactive Intelligence (CPII) Ltd., a CUHK-led under the InnoHK scheme of Innovation.

10.21437/Interspeech.2024-1392



(a) The overview of SQ-Codec

(b) The training illustration of SimpleSpeech

(c) The inference process of SimpleSpeech

Figure 1: *(a) shows the overview of the SQ-Codec. (b) and (c) give the illustration of SimpleSpeech in training and inference.*

2.2. Sentence duration

Previous works [2, 9] try to model the phone-level duration in TTS. In general, a duration predictor is built to predict the duration of each phone. Training such modules increases the complexity of data pre-processing and training pipeline. In this study, we propose to model the sentence-level duration prior by using the in-context learning of LLMs (gpt-3.5-turbo is used). Our motivation is that LLMs can easily estimate how much time to read a sentence based on the number of words in the sentence and the prior knowledge. The prompt for ChatGPT can be found on the demo page. After we obtain the sentence-level duration, we let the model learn the alignment between words and latent features implicitly. Such a design will bring more diversity to speech synthesis. In the training stage, we can directly get the duration based on the length of the waveform. We follow Stable Audio [14] uses a timing module to encode the duration into a global embedding. In the inference stage, the predicted duration by LLMs first determines the length of the noisy sequence, then input into the timing module.

2.3. SQ-Codec

Although residual vector quantization (RVQ) based audio codec models have shown effectiveness for audio compression, training a good codec model needs a lot of tricks and complicated loss design [15]. In this study, we propose to use scalar quantization [16, 17] to replace the residual vector quantization in audio codec models, which can be trained with the reconstruction loss and adversarial loss without any training tricks. Furthermore, we also find that the scalar quantization effectively maps the complex speech signal into a finite and compact latent space, which is suitable for diffusion model (refer to Section 2.4 and 3.2.4 for more details). Assuming $\boldsymbol{h} \in \mathcal{R}^{T*d}$ denotes the output features of Encoder in the codec model. T and d denote the number of frames and the dimension of each vector. For any vector $\boldsymbol{h}_i$, we use a parameter-free scalar quantization module to quantize $\boldsymbol{h}_i$ into a fixed scalar space:

$$\boldsymbol{h}_i = \text{torch.tanh}(\boldsymbol{h}_i), \quad \boldsymbol{s}_i = \text{torch.round}(\boldsymbol{h}_i * S)/S, \qquad (1)$$

where S is a hyper-parameter that determines the scope of scalar space. To get gradients through the rounding operation, we use a straight-through estimator like VQ-VAE [18]. We can see that the scalar quantization first uses a *tanh* function to map the value of features into $[-1, 1]$, then a *round* operation further reduces the value of range into 2*S+1 different numbers. We named such value domain as *scalar latent space*. We note that previous works [17, 19] also try to use scalar quantization as the image tokenizer for image generation. We claim that our implementation is different from theirs, and better adapts audio codec tasks in our experiments.

Encoder and Decoder Our encoder consists of 5 convolution blocks, each block includes 2 causal 1D-convolutional layers and one down-sample layer. The down-sample strides are set as $[2, 2, 4, 4, 5]$, resulting in 320 times down-sample along the time dimension. The decoder mirrors the encoder and uses transposed convolutions instead of stride convolutions.

Discriminator and Training Loss Following [6], a multi-scale discriminator is used. The training loss of SQ-Codec consists of two parts: (1) reconstruction loss $\mathcal{L}_{rec}$, which includes time domain and frequency domain losses: the L1 loss between the reconstructed waveform and the original waveform and the MSE loss for the STFT spectrogram. (2) adversarial loss, which is calculated based on the results of the discriminator.

2.4. Scalar Latent Transformer Diffusion Models

Based on the previous discussion, we can map the speech data into a *scalar latent space* based on our proposed SQ-Codec model. Inspired by the success of latent diffusion models [20] in both image and audio generation [9,21], we propose to model the speech data in the *scalar latent space*. Our motivation is that the sampling space of *scalar latent space* is simple because SQ effectively limits the value range of each element. The details of the scalar latent transformer diffusion model are as follows.

Transformer-based diffusion backbone U-Net backbone has been widely used in diffusion models, especially in the speech synthesis field [9,22–24]. We also noted that many transformer-based diffusion models, such as DiT [25] and Sora [26] have been used in image/video generation. Inspired by the success of the transformer-based audio language models [4–6] and DiT, we propose a transformer-based diffusion backbone for speech synthesis. Specifically, a GPT2-like transformer backbone is used: 12 attention layers, 8 attention heads, and the model dimension is 768.

In-Context Conditioning Inspired by LLMs and previous audio language models [4–6], we simply append the features of time step t and condition c as the prefix sequence in the input sequence. Such a condition way allows us to use a standard GPT-like structure without modification. After the final block, we remove the conditioning sequence from the output sequence.
Scalar Latent Diffusion Latent diffusion models (LDM) have been demonstrated to fit complex data distributions, including VAE latent features [9, 20–22], Mel-spectrogram features [27], and waveform [23, 24]. We speculate that these data distributions are very complex because their search space is infinite. Instead, SQ-Codec provides a finite and compact *scalar latent space*, thus we can consider modeling speech data in this space. Specifically, a network is trained to transfer the Gaussian distribution to the scalar latent space. We follow the training strategy of DDPM [28], and the mean squared error (MSE) loss is used. To make sure the final output belongs to the scalar latent space, we use the scalar quantization (SQ) operation to limit the final prediction.

$$\hat{\boldsymbol{x}}_0 = SQ(\theta(\boldsymbol{x}_T, T, \boldsymbol{c})) \tag{2}$$

where θ denotes the parameter of the neural network. T denotes the timestep, $\boldsymbol{x}_T$ denotes the sampling features from the Gaussian distribution. $\boldsymbol{c}$ denotes the condition information.

3. Experiments

3.1. Experimental setting

Dataset We create a dataset with only 4k hours of unlabeled English speech data sampled from the Multilingual LibriSpeech (MLS) dataset [29]. For the evaluation set, we follow E3TTS [24], evaluating the models on the LibriTTS test clean set [30].
Model training For the SQ-Codec model, we train it on the LibriTTS [30] dataset. We set the sample rate as 16k Hz. Unless specifically stated, we set $S = 9$ and $d = 32$. We train the SQ-Codec with a learning rate of 2e-3, and Adam optimizer is used. For the diffusion, we train it on our selected MLS dataset. The AdamW optimizer with a learning rate of 1e-4 is used.
Baselines For codec models, we compare with publicly available audio codec models, including Encodec [8], DAC [15], and HiFi-Codec [31]. For TTS models, to establish a standard for comparison, our study employs five baseline models, including VALL-EX [32], Pheme TTS [33], X-TTS [2], E3TTS [24], and NaturalSpeech 2 [9]. The first three models are industry baselines with publicly available pre-trained models. E3TTS and NaturalSpeech 2 are not publicly available, we directly compare them with their released demos. E3TTS is the most related to us, which proposes to directly predict waveform using a diffusion model. However, E3TTS is trained on fixed length speech (*e.g.* 10 seconds), such setting limits the flexibility of the model. Instead, SimpleSpeech supports the generation of a variable-length speech based on the sentence-level duration.

Evaluation metrics For Codec models, we use the following metrics: Perceptual Evaluation of Speech Quality (PESQ), Short-Time Objective Intelligibility (STOI), and structural similarity index measure (SSIM). We use objective and subjective metrics to assess the zero-shot synthesis ability. For objective metrics, word error rate (WER) is used to measure the robustness of TTS models. Speaker Similarity (SIM) is used to access the clone ability. DNSMOS [34] and Mel-Cepstral Distortion (MCD) are used to measure speech quality. To evaluate the generation speed, we calculate the average time of each model to generate 20 utterances in the same machine. For subjective metrics, we hire 30 listeners to conduct the MOS and Similar MOS (SMOS) evaluation.

Table 1: *Performance comparison between open-sourced audio codec models and ours. B denotes the Bitrate (kbps). H denotes the Hop-size. Evaluation is conducted on the LibriTTS test set.*

Model	Size (M)	H	B	PESQ	STOI	SSIM
Encodec	14	75	12	3.76	0.90	0.72
DAC	70	50	8	3.99	0.95	0.85
HiFiCodec	60	50	2	3.24	0.88	0.72
Ours	5	50	8	**4.16**	**0.95**	**0.86**

Table 2: *The influence of parameters S and d for SQ-Codec.*

Setting ID	S	d	PESQ	STOI	SSIM
1	9	32	4.16	0.95	0.86
2	2	18	3.33	0.88	0.71
3	1	32	2.57	0.84	0.62

3.2. Experimental Results

3.2.1. Performance of SQ-Codec model

In this study, the codec model plays an important role in the TTS system, its reconstruction performance determines the upper limit of speech generation. Thus, we first conduct experiments to validate the reconstruction performance of our proposed SQ-Codec model. Table 1 shows the performance comparison between previous audio codec models and our SQ-Codec. The results demonstrate that the SQ-Codec with the highest PESQ, STIO, and SSIM. Furthermore, the SQ-Codec model needs fewer model parameters.

3.2.2. The ablation study of SQ-Codec model

In this part, we conduct an ablation study to explore the influence of parameters S and d. When $S = 1$, we follow previous work [19] to quantize the value into -1 or 1. Although such a strategy gets good performance in image compression, we can see that it cannot work very well in audio compression. When we increase S and d, we can see the performance improvement.

3.2.3. Performance of SimpleSpeech

In this part, we conduct experiments to validate the performance of zero-shot speech synthesis. Table 3 shows the results.
Generation Quality One of the problems in previous large-scale TTS is their poor speech quality. In this study, we use better audio codec models and a novel generation model, we can see that both the subjective and objective metrics show SimpleSpeech obtains better speech quality. From the MOS and DNSMOS scores, we can see the generated samples by SimpleSpeech even surpass the ground truth speech.
Speaker Similarity Speaker similarity is an important metric to access the performance of voice cloning in TTS. [3] Previous works have demonstrated that by increasing the speech data to

[2] https://github.com/Plachtaa/VALL-E-X, https://github.com/PolyAI-LDN/pheme, https://github.com/coqui-ai/TTS

[3] The cosine similarity between the generated speech and prompt. The speaker encoder in Section 2.1 is used to extract global embedding.

Table 3: *The comparison with previous zero-shot TTS models. MOS and SMOS are presented with 95% confidence intervals.*

Model	MCD (↓)	SIM (↑)	WER (↓)	DNSMOS (↑)	Speed (seconds ↓)	MOS (↑)	SMOS (↑)
GroundTruth		-	2.7	3.82	-	4.39 ± 0.07	4.26 ± 0.10
VALL-EX	8.16	0.946	5.6	3.75	34.5	4.07 ± 0.12	**4.15 ± 0.20**
Pheme	8.56	**0.958**	4.6	3.70	3.3	3.89 ± 0.23	3.79 ± 0.29
X-TTS	9.93	0.949	5.6	3.80	2.77	4.19 ± 0.15	4.00 ± 0.13
SimpleSpeech (ours)	**7.51**	0.956	**3.3**	**3.90**	**1.6**	**4.45 ± 0.05**	4.14 ± 0.12

Table 4: *The Small-Scale Subjective Test*

Model	E3TTS	NaturalSpeech 2	SimpleSpeech
MOS	4.32 ± 0.08	4.26 ± 0.08	4.38 ± 0.10
SMOS	4.38 ± 0.17	4.38 ± 0.14	4.30 ± 0.11

Table 5: *The ablation study for each part in SimpleSpeech.*

Types	WER	SIM	MCD	DNSMOS
SimpleSpeech	3.3	0.956	7.51	3.90
Replacing the text encoder				
ByT5 → BERT	19.2	0.958	8.21	3.86
Replacing the diffusion backbone				
Transformer → Unet	5.75	0.949	8.16	3.86
Compared to VAE-based LDM				
SQ-Codec → VAE	5.1	0.952	9.04	3.49
Replacing the condition way				
IC → Cross-attention	9.6	0.958	8.29	3.90
Training Data				
MLS → LibriTTS	7.5	0.94	7.71	3.58

a large scale (*e.g.* 60k hours), the voice cloning ability will be improved. Although SimpleSpeech only trained on 4k hours of data, we can see that it also shows comparable performance to baselines trained with much more data.

Robustness We assess the robustness of TTS models by measuring the word error rate of generated speech (the HuBERT model [35] is used). We can see that SimpleSpeech shows better robustness than previous works, *e.g.* SimpleSpeech is more robust than VALL-EX. Similar to VALL-EX, SimpleSpeech does not need phone-level duration alignment information.

Generation Speed AR-based large TTS models, such as VALL-E and VALL-EX, try to generate the speech tokens in an AR way, which will cost a lot of time in the inference. In this work, our model first generates scalar latent features by NAR way, then these generated features can be used to recover the waveform using SQ-Codec's decoder. Furthermore, our SQ-Codec model is very small, so it further improves the generation speed. Our proposed model can get high-quality speech with only 100 DDPM steps, instead, E3TTS and NaturalSpeech 2 use 1000 steps and 150 steps to ensure sufficient generation quality. It also validates the efficiency of our proposed model.

3.2.4. The ablation study of SimpleSpeech

We conduct ablation studies to explore the influence of each part in SimpleSpeech. The experimental results are shown in Table 5, and the qualitative findings are as follows.

Text encoder From Table 5, we can see that using BERT [36] as the text encoder results in a decrease in robustness (WER from 3.3 to 19.2). We can see that ByT5 is critical for generating coherent speech. A potential explanation could be that ByT5 recognizes individual characters, whereas BERT relies on subword tokenization techniques (such as byte-pair encoding). Since words that are spelled similarly often sound alike, the ability to discern characters can enhance the model's ability.

Model Structure Unet-based structure [20, 21] have been widely used in diffusion models. In this study, we show that our proposed transformer-based diffusion models can bring better performance than the Unet-based model.

Latent Space We explore the influence of latent space, including VAE-latent space [20], and our proposed scalar latent space. For VAE-latent space, we follow Diffvoice [22] to train a VAE model. The latent space dimension is set as 20 for both mean and variance. We can see that our model brings a significant improvement over previous VAE-based LDM methods. We also explore applying LDM to the latent space of the DAC codec (following [9]), but we fail to train such a model. We conjecture that the latent dimension of DAC codec is 1024, it is hard for LDM to predict. We think that the dimension of latent space not only influences the reconstruction ability of the codec but also influences the performance of generation models.

Condition style We explore the influence of different types of condition ways, including cross-attention-based and our proposed in-context (IC) condition. We can see that our proposed in-context condition brings better performance than the cross-attention condition when the condition features do not include explicit duration information. Similar to LM-based generation methods, in-context condition makes our model learn the duration information by itself.

Data quantity We compare the performance gap when training SimpleSpeech on LibriTTS (400h) and our selected MLS (4000h) datasets. We notice that scaling the data quantity can significantly improve the robustness of SimpleSpeech.

4. Conclusion

In this study, we propose a simple and efficient TTS model named SimpleSpeech. SimpleSpeech can be trained on large-scale speech-only datasets without any additional data preprocessing, which significantly simplifies the efforts to train a TTS model. We propose a novel speech codec (SQ-Codec) based on scalar quantization. Then we apply a novel latent transformer diffusion model to the scalar latent space of SQ-Codec. Experimental results show the proposed model has better performance than the previous Unet-based latent diffusion model. Due to the NAR generation strategy, SimpleSpeech significantly improves the generation efficiency compared to previous LM-based TTS models. In the future, we will explore to scale the model and data size.

5. References

[1] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan *et al.*, "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions," in *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2018, pp. 4779–4783.

[2] Y. Ren, C. Hu, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "Fastspeech 2: Fast and high-quality end-to-end text to speech," in *International Conference on Learning Representations*, 2020.

[3] Z. Borsos, R. Marinier, D. Vincent, E. Kharitonov, O. Pietquin, M. Sharifi, D. Roblek, O. Teboul, D. Grangier, M. Tagliasacchi *et al.*, "Audiolm: a language modeling approach to audio generation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2023.

[4] 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.

[5] 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," *arXiv preprint arXiv:2302.03540*, 2023.

[6] 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," *arXiv preprint arXiv:2310.00704*, 2023.

[7] R. Huang, C. Zhang, Y. Wang, D. Yang, L. Liu, Z. Ye, Z. Jiang, C. Weng, Z. Zhao, and D. Yu, "Make-a-voice: Unified voice synthesis with discrete representation," *arXiv preprint arXiv:2305.19269*, 2023.

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

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

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

[11] 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," *Advances in neural information processing systems*, vol. 36, 2024.

[12] 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.

[13] A. Conneau, A. Baevski, R. Collobert, A. Mohamed, and M. Auli, "Unsupervised cross-lingual representation learning for speech recognition," *Interspeech*, 2020.

[14] Z. Evans, C. Carr, J. Taylor, S. H. Hawley, and J. Pons, "Fast timing-conditioned latent audio diffusion," *arXiv preprint arXiv:2402.04825*, 2024.

[15] 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.

[16] J. Ballé, V. Laparra, and E. P. Simoncelli, "End-to-end optimized image compression," in *International Conference on Learning Representations*, 2016.

[17] F. Mentzer, D. Minnen, E. Agustsson, and M. Tschannen, "Finite scalar quantization: Vq-vae made simple," in *The Twelfth International Conference on Learning Representations*, 2023.

[18] A. Van Den Oord, O. Vinyals *et al.*, "Neural discrete representation learning," *Advances in neural information processing systems*, vol. 30, 2017.

[19] L. Yu, J. Lezama, N. B. Gundavarapu, L. Versari, K. Sohn, D. Minnen, Y. Cheng, A. Gupta, X. Gu, A. G. Hauptmann *et al.*, "Language model beats diffusion-tokenizer is key to visual generation," in *The Twelfth International Conference on Learning Representations*, 2023.

[20] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 684–10 695.

[21] R. Huang, J. Huang, D. Yang, Y. Ren, L. Liu, M. Li, Z. Ye, J. Liu, X. Yin, and Z. Zhao, "Make-an-audio: Text-to-audio generation with prompt-enhanced diffusion models," *ICML*, 2023.

[22] Z. Liu, Y. Guo, and K. Yu, "Diffvoice: Text-to-speech with latent diffusion," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[23] Z. Kong, W. Ping, J. Huang, K. Zhao, and B. Catanzaro, "Diffwave: A versatile diffusion model for audio synthesis," *ICLR*, 2020.

[24] Y. Gao, N. Morioka, Y. Zhang, and N. Chen, "E3 tts: Easy end-to-end diffusion-based text to speech," in *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2023, pp. 1–8.

[25] W. Peebles and S. Xie, "Scalable diffusion models with transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 4195–4205.

[26] OpenAI, "Video generation models as world simulators," in *Technical report*, 2024.

[27] S. Liu, D. Su, and D. Yu, "Diffgan-tts: High-fidelity and efficient text-to-speech with denoising diffusion gans," *ICML WorkShop*, 2022.

[28] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.

[29] V. Pratap, Q. Xu, A. Sriram, G. Synnaeve, and R. Collobert, "Mls: A large-scale multilingual dataset for speech research," *InterSpeech*, vol. abs/2012.03411, 2020.

[30] 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.

[31] 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.

[32] 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.

[33] P. Budzianowski, T. Sereda, T. Cichy, and I. Vulić, "Pheme: Efficient and conversational speech generation," *arXiv preprint arXiv:2401.02839*, 2024.

[34] C. K. Reddy, V. Gopal, and R. Cutler, "Dnsmos: A non-intrusive perceptual objective speech quality metric to evaluate noise suppressors," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 6493–6497.

[35] 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.

[36] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.