StableTTS: Towards Efficient Denoising Acoustic Decoder for Text to Speech Synthesis with Consistency Flow Matching

Zhiyong Chen* *Shanghai University* Shanghai, China zhiyongchen@shu.edu.cn

Xinnuo Li* *New York University* New York, USA xl5454@nyu.edu

Shuhang Wu, Zhi Yang *Shanghai University* shuhang_wu@outlook.com zhi.yang03@outlook.com

Zhiqi Ai *Shanghai University* Shanghai, China aizhiqi-work@shu.edu.cn

Shugong Xu *Shanghai University* Shanghai, China shugong@shu.edu.cn

***Abstract*—Current state-of-the-art text-to-speech (TTS) systems predominantly utilize denoising-based acoustic decoders with language models (LLMs) or with non-autoregressive front-ends, known for their superior performance in generating high-fidelity spectrum. In this study, we introduce an efficient TTS system that incorporates Consistency Flow Matching denoising training. This training approach significantly enhances the training efficiency and operational performance of denoising-based acoustic decoders in existing TTS or voice conversion systems, with no additional cost in the training process—a free lunch. To efficiently compare with other denoising strategies, we align with the latest advancements in the implementation of non-autoregressive-based TTS systems and build an efficient DiT-based TTS architecture. Our comprehensive evaluations against various denoising-based methods affirm the efficiency of our proposed system[1].**

***Index Terms*—component, formatting, style, styling, insert.**

I. Introduction

Text-to-speech synthesis (TTS) aims to generate high-quality speech from text inputs, ensuring clarity and intelligibility. With advancements in deep learning, TTS research has seen significant improvements in recent years.

Recent advances in TTS systems can be categorized into LLM-based autoregressive models and diffusion-based non-autoregressive models. Leveraging the in-context learning capabilities of large language models (LLMs), several studies have applied LLMs to model discrete audio tokens from neural codecs [14], [22]–[24]. These approaches have demonstrated remarkable performance in zero-shot TTS, capable of cloning timbre and prosody with extremely high audio quality when combined with neural codec models.

A major component in TTS systems combining LLMs or other non-autoregressive (NAR) content encoders is the denoising acoustic decoder, which commonly employs denoising probabilistic models (DDPM) [10], flow-matching-based models [11], Schrödinger bridge methods [12], or rectified flow-based methods [30]. Flow-matching-based methods have proven to be highly efficient in TTS systems, benefiting from an effective large language model front-end.

Optimal Transport Conditional Flow Matching (OT-CFM) [25] can be considered a specific type of flow matching with a defined trajectory and is widely used in current state-of-the-art TTS systems [20]. It also combines effectively with LLM-based content encoders, such as in CosyVoice [30], as well as in zero-shot voice conversion systems like Seed-VC [31]. However, OT-CFM is not an optimal and efficient training method and may suffer from cumulative errors, as analyzed in [2]. We propose to optimize the flow trajectory in a more flexible and effective manner by enforcing the self-consistency property in the flow velocity and flow endpoints during the training of the acoustic decoder. This approach ultimately results in faster generation with fewer sampling steps in inference, improved generation quality, and better training effectiveness. Unlike ReFlow [30], the Consist-FM decoder does not require additional sample generation steps or increased disk usage, and even better efficiency [2]—a free-lunch method based on existing architectures.

Our contributions in this work are as follows:

- We introduce StableTTS, a TTS system that utilizes Consistency Flow Matching to optimize the training efficiency of the denoising acoustic decoder, thereby achieving optimal training and inference sampling efficiency.
- We have enhanced the existing non-autoregressive TTS architecture by integrating a diffusion transformer for both the content encoder and acoustic decoder to prepare an effective and efficient architecture for comparison with other denoising methods. We believe such findings can be easily adapted to existing LLM-based content front-end TTS architectures.

II. Related Work

Modern autoregressive TTS models treat TTS as a language modeling task. Vall-E [7] utilizes neural codec codes as intermediate representations instead of mel-spectrograms. BASETTS [13] deploys a 1-billion-parameter autoregressive Transformer model and confirms the capabilities of large TTS models. CosyVoice [14] introduces a novel codec-based voice synthesizer that combines an LLM for text-to-token generation and a conditional flow-matching model for token-to-speech

*Contributed equally to this work.

[1]https://zhiyongchengreat.github.io/stabletts_ccfm

synthesis. MELL-E [15] proposes a continuous-valued token-based language modeling approach that autoregressively generates continuous mel-spectrogram frames directly from text.

Denoising models for acoustic decoders have emerged as a powerful approach in both autoregressive and non-autoregressive TTS models. They provide robust frameworks for learning complex high-dimensional data distributions (Mel spectrum or audio latents) through continuous-time diffusion processes. Diff-TTS [16] was the first to apply diffusion probabilistic models (DPMs) for acoustic modeling. Grad-TTS [10] introduces a score-based decoder for generating mel-spectrograms, showcasing the potential of diffusion models in TTS. To enhance generation speed, Fast Grad-TTS [17] explores specific sampling methods to accelerate inference. CoMoSpeech [18] introduces a consistency model-based approach, enabling high-quality audio generation in just a single diffusion sampling step. LightGrad [19] incorporates a lightweight U-Net diffusion decoder with a training-free fast sampling technique, reducing both model parameters and inference latency.

Flow-matching techniques have also gained attention in the TTS domain. Matcha-TTS [1] and P-Flow [21] employ an ODE-based decoder that generates high-quality output in fewer synthesis steps. VoiceFlow [1] uses rectified flow resampling techniques for more efficient synthesis.

In recent flow-matching methods for learning flows, it is often necessary to approximate the transformation between two distributions. However, this process is computationally intensive and introduces additional approximation errors, as seen in rectified flow [35] and consistency models [36]. To overcome these challenges, Consistency Flow Matching [2] learns a multisegment flow with constraints on both velocity and endpoints.

III. Preliminary on the optimal-transport flow-matching based TTS system

The flow-matching based method is used as the acoustic decoder for an encoder-decoder-based zero-shot TTS system. Our overall architecture for comparing denosing decoders is demonstrated in Fig. 1. We use a Diffusion Transformer [26] as the text-encoder as the content generation mainly for efficient and fair comparison.

Let x_1 denote an observation in the mel spectrum, sampled from an unknown data distribution $q(x)$. A probability density path is a time-dependent probability density function $p_t(x)$. One way to generate samples from the data distribution $q(x)$ is to construct a probability density path $p_t(x)$, to transform noise $x_0 \sim p_0(x) = N(0, I)$, the prior normal distribution, such that $p_1(x)$ approximates the data distribution $q(x)$. Flow matching first defines a vector field $\mathbf{v}_t$, which generates the flow ϕ_t through the ODE:

$$\frac{d}{dt}\phi_t(x) = \mathbf{v}_t(\phi_t(x)); \phi_0(\mathbf{x}) = x. \tag{1}$$

We can sample from the approximated data distribution $p_1(x)$ by solving the ODE initial value problem in Eq. (1)

Suppose there exists a known vector field $\mathbf{u}_t$ that generates a probability path p_t. The flow matching loss is defined as:

$$L_{FM}(\theta) = E_{t,p_t(x)}||\mathbf{u}(x,t) - \mathbf{v}(x,t;\theta)||^2 \tag{2}$$

where $\mathbf{v}(x,t;\theta)$ is a neural network with parameters θ. To make training tractable by training with x_0 and x_1, conditional flow matching with optimal transport (OT-CFM) considers:

$$L_{OTCFM}(\theta) = E_{t,q(x_1),p_0(x_0)}||\mathbf{u}_t^{OT}(x|x_0,x_1) - \mathbf{v}_t(x|\mu;\theta)||^2 \tag{3}$$

where $\mathbf{x}$ is sampled from:

$$x \sim N[(1-t)x_0 + tx_1, \sigma] \tag{4}$$

$$\mathbf{u}_t^{OT}(x|x_0,x_1) = x_1 - x_0 \tag{5}$$

The neural network for estimating the vector field is conditioned on the content sequence embeddings μ, as in [11].

IV. Consistency Flow Matching for Denoising Acoustic Modeling

A. Consistency Flow Matching for Acoustic Decoding

In this paper, we propose using the consistency flow matching [2] for the denoising acoustic model. The core of the consistency training contains two constraints. One is to directly constrain the velocity vector field to be consistent in the transport in each training segment. The second constraint ensures that, starting from an arbitrary time t with data point x_t, and moving in the direction of the current velocity for a duration of $1-t$, the resulting ending point will be consistent. We omit the notations for conditions on μ and θ, and define the method:

$$L_{consistFM}(\theta) = E_t(EndConsist + VeloConsist) \tag{6}$$

$$EndConsist = \|\mathbf{f}(x_t) - \mathbf{f}(x_{t+\Delta t})\|^2 \tag{7}$$

$$VeloConsist = \|\mathbf{v}(x_t) - \mathbf{v}(x_{t+\Delta t})\|^2 \tag{8}$$

$$\mathbf{f}(x_t) = x_t + (1-t)\mathbf{v}(x_t) \tag{9}$$

$$x_t = (1-t)x_0 + tx_1 \tag{10}$$

For samples $x_1 \sim q(x)$ and $x_0 \sim p_0(x)$, equation (6) demonstrates the optimization loss for training the consistency flow matching based denoising acoustic model. Δt denotes a time interval which is very small. The training process consists of matching the endpoints and velocity at both $\mathbf{x}$ and $\mathbf{x}_{t+\Delta t}$.

B. Multi-segment Consistency Flow Matching and Initial Pretraining

To further enhance the transfer flexibility and wish to better matching the NFEs in inference sampling, we train Consist-FM with multi-segment strategy. Additionally, we perform two-stage training for this multi-segment Consist-FM [2].

In the first initial pretraining stage, for samples $x_1 \sim q(x)$ and $x_0 \sim p_0(x)$, $t \sim Uniform(0,1)$. We define K segments and linear sampling K endpoints $(x_0, x_{1/K}, ..., x_{k/K}, ..., x_1)$. The initial training stage uses a velocity-consistent flow matching, which is similar to OT-CFM:

$$L_{ConsistInit}(\theta) = E_{k,t}||\mathbf{u}^{init}(x_t) - \mathbf{v}(x_t)||^2 \tag{11}$$

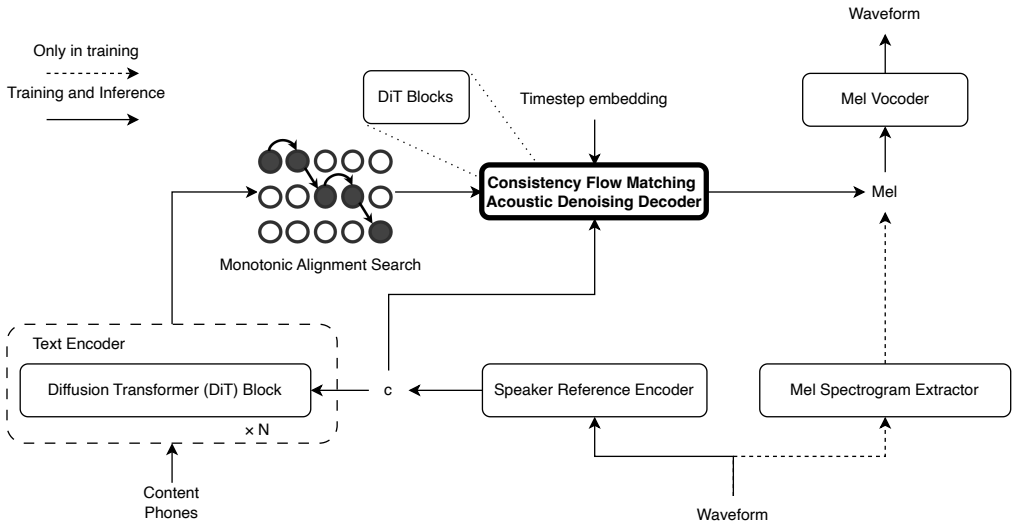

Fig. 1. Overall architecture for StableTTS.

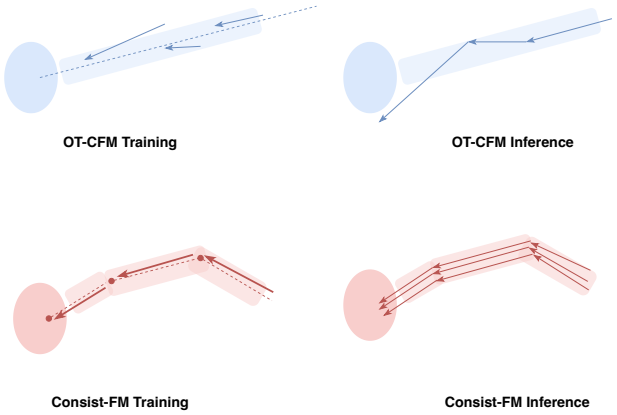

Fig. 2. ConsistFM and OT-CFM Method Comparison

for those $t \in [(k-1)/K, k/K]$ and $x_t \in [x_{(k-1)/K}, x_{k/K}]$,

$$\mathbf{u}^{init}(x_t) = (x_{k/K} - x_t)/(k/T - t). \tag{12}$$

In the second stage for consistency flow matching training, we use multi-segment consistency flow matching loss. Similarly to Eq. (6), for those samples $t \in [(k-1)/K, k/K]$ and $x_t \in [x_{(k-1)/K}, x_{k/K}]$,

$$L_{ConsistFMK}(\theta) = E_{t,k}(EndConsist_k + VeloConsist_k) \tag{13}$$

$$EndConsist_k = \|\mathbf{f}_k(x_t) - \mathbf{f}_k(x_{t+\Delta t})\|^2 \tag{14}$$

$$VeloConsist_k = \|\mathbf{v}_k(x_t) - \mathbf{v}_k(x_{t+\Delta t})\|^2 \tag{15}$$

$$\mathbf{f}_k(x_t) = x_t + (k/K - t)\mathbf{v}_k(x_t) \tag{16}$$

Here, $\mathbf{f}_k$ and $\mathbf{v}_k$ are the endpoint estimator and velocity estimator for segment k. An conceptual drawing in Fig. 2.

C. Denoising Acoustic Decoder on Efficient TTS architecture

Our evaluation backbone architecture for StableTTS consists of a speaker encoder, a duration predictor, a text encoder, a denoising- based acoustic decoder, and a Mel vocoder [32]. The text encoder and flow-matching decoder are entirely composed of diffusion transformer blocks. After the reference audio y_{mel} passes through the style encoder [27], it generates a global feature vector g related to the speaker's identity and style. To implementing zero-shot TTS, we inject the speaker information g into the model, g is processed through MLP layer to obtain γ and β. These are then used as scaling and shifting parameters to inject the style information into the transformer in the form of adaptive layer normalization. A FiLM layer [28] is inserted before each of the diffusion transformer block in the acoustic decoder to condition the timestep information. These implementation is similar with the original DiT [34]. The Snake activation function is used in the feed-forward network in the FFN layer. We concat noise and content embeddings as input to the denosing acousctic decoder. We found that the training process remains smooth when we shift from OT-CFM to ConsistFM. The implementation of our architecture and the denoising decoder can be found on the project website.

V. Experiments

A. Dataset and experimental setting

We train StableTTS and the baseline systems fairly on the LibriTTS multi-speaker TTS corpus, training all models for 300 epochs. Testing is conducted on the LibriTTS dataset to evaluate the synthesis quality using 10 speakers for audio fidelity and speaker similarity assessments. For objective evaluations, we use a pretrained speaker verification model [33]. We compute the Speaker Encoder Cosine Similarity (SECS)

TABLE I
COMPARING DENOISING ACOUSTIC DECODER BASED ON THE SAME ARCHITECTURE

Model	MCD↓	SECS↑	DNSMOS↑	Denoising Method	NFE↓	RTF(CPU)↓
Ground Truth	0.00	0.86	3.82	DDPM	-	-
GradTTS+ [10]	8.49	0.72	3.47	DDPM	1000	2.20
MachaTTS+ [11]	7.21	0.81	3.80	OT-CFM	35	0.47
Reflow+ [3]	9.61	0.53	2.23	Reflow	35	0.47
StableTTS	**7.18**	**0.82**	**3.85**	ConsistFM	35	0.47
GradTTS+ [10]	9.81	0.60	2.65	DDPM	10	0.15
MachaTTS+ [11]	7.47	0.79	3.79	OT-CFM	10	0.10
StableTTS	**7.32**	**0.82**	**3.88**	ConsistFM	**10**	**0.10**
MachaTTS+ [11]	8.10	0.77	3.76	OT-CFM	6	0.08
StableTTS	**7.85**	**0.77**	**3.80**	ConsistFM	**6**	**0.08**

TABLE II
INFERENCE COST EVALUATION FOR STABLETTS WITH DENOISING ACOUSTIC MODEL-BASED SYSTEMS

Model	Stage	NFE↓	MCD↓	SECS↑	DNSMOS↑
Ground Truth	-	-	0.00	0.86	3.82
StableTTS-6Seg	Init	10	7.38	0.79	3.85
StableTTS-6Seg	Consist	10	**7.32**	**0.82**	**3.88**
StableTTS-4Seg	Init	10	7.45	0.80	3.80
StableTTS-4Seg	Consist	10	7.38	0.81	3.82
StableTTS-2Seg	Init	10	7.44	0.80	3.81
StableTTS-2Seg	Consist	10	7.40	0.81	3.87

by comparing speaker embeddings from both synthesized and ground-truth speech samples. We measure the Mel-cepstral Distortion (MCD) to gauge structural disparities. To avoid subjective bias, we follow recent research using DNSMOS [29] to evaluate the audio P808-MOS objective score for audio quality assessment. ConsistFM converges as easily as OT-CFM in our experiments, with no degradation in training speed.

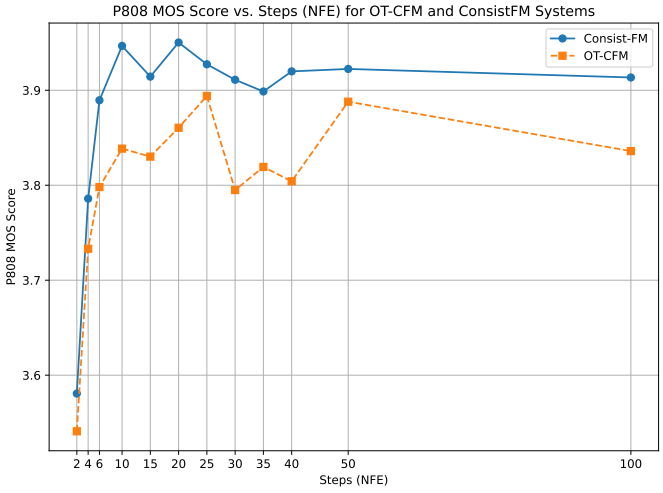

Fig. 3. ConsistFM compare with OT-CFM at different sampling steps (NFEs)

B. Results

We evaluated the proposed systems alongside several state-of-the-art denoising decoders for TTS systems, as shown in Table I. GradTTS+ and MatchaTTS+ are widely used methods with diffusion denoising probabilistic models and OT-CFM, and we adapt them to our backbone for a fair comparison. StableTTS with ConsistFM outperforms on all metrics with very low NFEs, as well as at normal operational NFEs. With NFE equal to 6 and RTF at 0.08, ConsistFM can still generate better quality audio. The ReFlow-based method [30] does not work properly with our backbone, and it requires resampling and massive storage for features, making it not as concise as ConsistFM.

Further comparisons of systems using the initial OT-CFM strategy and second-stage ConsistFM training are shown in Table II. StableTTS generally performs better in the second-stage training with consistency flow matching, demonstrating the effectiveness of multi-stage training. We also compare different segmentation settings. Performance is generally better when using a segment length equal to 6. Moreover, we compare the consistency flow matching acoustic decoder with the optimal-transport flow matching-based decoder with different NFEs in Fig.3, showing that the proposed ConsistFM strategy generally performs better for all NFEs.

VI. CONCLUSION

In this paper, we introduced StableTTS, an efficient text-to-speech (TTS) system that incorporates Consistency Flow Matching denoising training—a ”free lunch” training strategy on a highly efficient backbone. By aligning with the latest advancements in non-autoregressive TTS systems, we constructed an efficient DiT-based TTS architecture to effectively compare our method with other denoising strategies, including DDPM, OT-CFM, and others. The Consistency Flow Matching training significantly enhances the performance and efficiency of denoising-based acoustic decoders, improving both training efficiency and sampling efficiency during inference. Our comprehensive evaluations against various denoising-based methods affirm the effectiveness of our proposed system in boosting denoising decoder performance and achieving faster inference times. Future work will focus on scaling our method to larger datasets and integrating it with existing front-end LLMs based architectures to further improve performance in content and prosody modeling.

REFERENCES

[1] Mehta, Shivam, et al. "Matcha-TTS: A fast TTS architecture with conditional flow matching." ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2024.

[2] Yang, Ling, et al. "Consistency Flow Matching: Defining Straight Flows with Velocity Consistency." arXiv preprint arXiv:2407.02398 (2024).

[3] Casanova, Edresson, et al. "Yourtts: Towards zero-shot multi-speaker tts and zero-shot voice conversion for everyone." International Conference on Machine Learning. PMLR, 2022.

[4] Lee, Sang-Hoon, et al. "Hierspeech++: Bridging the gap between semantic and acoustic representation of speech by hierarchical variational inference for zero-shot speech synthesis." arXiv preprint arXiv:2311.12454 (2023).

[5] Camb-ai. MARS5-TTS. GitHub, 2023, https://github.com/Camb-ai/MARS5-TTS/tree/master. Accessed 13 Sept. 2024.

[6] Shen, Kai, et al. "NaturalSpeech 2: Latent Diffusion Models are Natural and Zero-Shot Speech and Singing Synthesizers." The Twelfth International Conference on Learning Representations.

[7] Wang, Chengyi, et al. "Neural codec language models are zero-shot text to speech synthesizers." arXiv preprint arXiv:2301.02111 (2023).

[8] Peng, Puyuan, et al. "VoiceCraft: Zero-Shot Speech Editing and Text-to-Speech in the Wild." arXiv preprint arXiv:2403.16973 (2024).

[9] Casanova, Edresson, et al. "XTTS: a Massively Multilingual Zero-Shot Text-to-Speech Model." arXiv preprint arXiv:2406.04904 (2024).

[10] Popov, Vadim, et al. "Grad-tts: A diffusion probabilistic model for text-to-speech." International Conference on Machine Learning. PMLR, 2021.

[11] Mehta S, Tu R, Beskow J, et al. Matcha-TTS: A fast TTS architecture with conditional flow matching[C]//ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2024: 11341-11345.

[12] Chen, Zehua, et al. "Bridge-TTS: Text-to-Speech Synthesis with Schrodinger Bridge."

[13] Łajszczak, Mateusz, et al. "BASE TTS: Lessons from building a billion-parameter text-to-speech model on 100K hours of data." arXiv preprint arXiv:2402.08093 (2024).

[14] Du, Zhihao, et al. "Cosyvoice: A scalable multilingual zero-shot text-to-speech synthesizer based on supervised semantic tokens." arXiv preprint arXiv:2407.05407 (2024).

[15] Meng, Lingwei, et al. "Autoregressive Speech Synthesis without Vector Quantization." arXiv preprint arXiv:2407.08551 (2024).

[16] Jeong, Myeonghun, et al. "Diff-tts: A denoising diffusion model for text-to-speech." arXiv preprint arXiv:2104.01409 (2021).

[17] Vovk, Ivan, et al. "Fast Grad-TTS: Towards Efficient Diffusion-Based Speech Generation on CPU." Interspeech. 2022.

[18] Ye, Zhen, et al. "Comospeech: One-step speech and singing voice synthesis via consistency model." Proceedings of the 31st ACM International Conference on Multimedia. 2023.

[19] Chen, Jie, et al. "Lightgrad: Lightweight diffusion probabilistic model for text-to-speech." ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2023.

[20] Guo, Yiwei, et al. "Voiceflow: Efficient text-to-speech with rectified flow matching." ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2024.

[21] Kim, Sungwon, et al. "P-flow: a fast and data-efficient zero-shot TTS through speech prompting." Advances in Neural Information Processing Systems 36 (2024).

[22] Wang C, Chen S, Wu Y, et al. Neural codec language models are zero-shot text to speech synthesizers[J]. arXiv preprint arXiv:2301.02111, 2023.

[23] Chen S, Liu S, Zhou L, et al. VALL-E 2: Neural Codec Language Models are Human Parity Zero-Shot Text to Speech Synthesizers[J]. arXiv preprint arXiv:2406.05370, 2024.

[24] Borsos Z, Sharifi M, Vincent D, et al. Soundstorm: Efficient parallel audio generation[J]. arXiv preprint arXiv:2305.09636, 2023.

[25] Liu X, Gong C. Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow[C]//NeurIPS 2022 Workshop on Score-Based Methods.

[26] Peebles W, Xie S. Scalable diffusion models with transformers[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2023: 4195-4205.

[27] Min D, Lee D B, Yang E, et al. Meta-stylespeech: Multi-speaker adaptive text-to-speech generation[C]//International Conference on Machine Learning. PMLR, 2021: 7748-7759.

[28] Perez E, Strub F, De Vries H, et al. Film: Visual reasoning with a general conditioning layer[C]//Proceedings of the AAAI conference on artificial intelligence. 2018, 32(1).

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

[30] Guo, Yiwei, et al. "Voiceflow: Efficient text-to-speech with rectified flow matching." ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2024.

[31] Plachtaa. "GitHub - Plachtaa/Seed-vc: State-of-the-Art Zero-shot Voice Conversion and Singing Voice Conversion With in Context Learning." GitHub, github.com/Plachtaa/seed-vc/tree/main.

[32] Siuzdak, Hubert. "Vocos: Closing the gap between time-domain and Fourier-based neural vocoders for high-quality audio synthesis." arXiv preprint arXiv:2306.00814 (2023).

[33] Resemble-Ai. "GitHub - Resemble-ai/Resemblyzer: A Python Package to Analyze and Compare Voices With Deep Learning." GitHub, github.com/resemble-ai/Resemblyzer.

[34] Peebles, William, and Saining Xie. "Scalable diffusion models with transformers." Proceedings of the IEEE/CVF International Conference on Computer Vision. 2023.

[35] X. Liu, C. Gong, et al., "Flow straight and fast: Learning to generate and transfer data with rectified flow," in The Eleventh International Conference on Learning Representations, 2022.

[36] Y.Song, P.Dhariwal, M.Chen, andI.Sutskever,Consistency models, in International Conferenceon Machine Learning, pp. 32211–32252, PMLR, 2023.