DiFlow-TTS: Compact and Low-Latency Zero-Shot Text-to-Speech with Factorized Discrete Flow Matching

Anonymous ACL submission

Abstract

This paper introduces DiFlow-TTS, a novel zero-shot text-to-speech (TTS) system that employs discrete flow matching for generative speech modeling. We position this work as an entry point that may facilitate further advances in this research direction. Through extensive empirical evaluation, we analyze both the strengths and limitations of this approach across key aspects, including naturalness, expressive attributes, speaker identity, and inference latency. To this end, we leverage factorized speech representations and design a deterministic Phoneme-Content Mapper for modeling linguistic content, together with a Factorized Discrete Flow Denoiser that jointly models multiple discrete token streams corresponding to prosody and acoustics to capture expressive speech attributes. Experimental results demonstrate that DiFlow-TTS achieves strong performance across multiple metrics while maintaining a compact model size, up to 11.7 times smaller, and enabling low-latency inference that is up to 34 times faster than recent state-of-the-art baselines. Audio samples are available on our demo page: https://diflow-tts.github.io.

1 Introduction

Zero-shot text-to-speech (TTS) has made remarkable progress in recent years, with the goal of generating high-quality speech that faithfully replicates the voice of previously unseen speakers from only a few seconds of reference audio. Recent studies have explored autoregressive (AR) approaches, where speech is quantized into discrete codec tokens and modeled using language models (Zhang et al., 2023; Han et al., 2024; Meng et al., 2025; Song et al., 2024; Chen et al., 2024a; Peng et al., 2024; Ji et al., 2024a; Wang et al., 2025b; Chen et al., 2025). Although these models achieve strong performance, they typically require large-scale training data and suffer from slow inference.

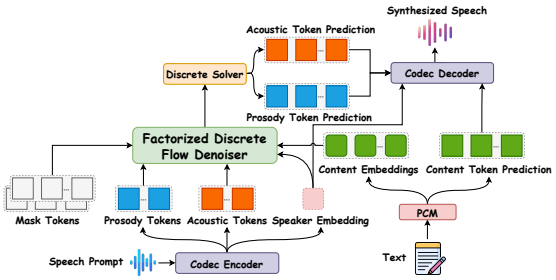

Figure 1: Overview of DiFlow-TTS. A Codec Encoder decomposes the speech prompt into speaker, prosody, and acoustic tokens, while the *Phoneme-Content Mapper* converts text into content embeddings. Conditioned on these, the *Factorized Discrete Flow Denoiser* generates prosody and acoustic tokens, and the Codec Decoder reconstructs the waveform.

Recent efforts to adapt these discrete codec tokens to generative paradigms have sparked growing interest in applying diffusion models within fully discrete settings (Yang et al., 2023; Wu et al., 2024; Yang et al., 2024; Ju et al., 2024). However, diffusion-based methods inherently couple the training and sampling processes, restricting sampling configurations to those established during training (Qin et al., 2025). Modifying components such as noise schedules or rate matrices requires retraining for each new configuration, resulting in considerable computational overhead. Flow matching, on the other hand, offers a more flexible and efficient alternative. Yet, most existing flow-based models for discrete data still follow a single design paradigm: the flow is defined over continuous representations derived from discrete inputs, rather than directly over discrete probability distributions. (Du et al., 2024b; Hieu et al., 2025; Wang et al., 2025a; Zuo et al., 2025a)

In response, we present DiFlow-TTS, illustrated in Figure 1, a novel zero-shot TTS framework that leverages discrete flow matching (DFM) tailored to discrete settings in the speech domain. This work aims to demonstrate discrete flow models

as a promising research direction for speech synthesis, supported by empirical experiments that reveal both the advantages and limitations of this approach. To address this gap, we take an initial step toward developing a framework that operates directly in the discrete space of factorized codec tokens by employing a pre-trained FACodec (Ju et al., 2024) as the target discrete representation. This choice is motivated by two key factors: **(1)** FACodec factorizes speech into prosody, content, and acoustic attributes, enabling modeling flexibility; and **(2)** it is pre-trained on a large-scale multi-speaker corpus, providing a robust and reliable codec foundation for our target data. Building on this, DiFlow-TTS explicitly models these factorized attributes within a compact and unified framework. Specifically, we design Phoneme-Content Mapper (PCM), which maps phoneme sequences to discrete speech tokens that represent the content of the utterance. This module further generates content embeddings that align closely with the semantic structure of the speech. These embeddings, along with auditory attributes extracted from the reference speech prompt, are then used to condition a Factorized Discrete Flow Denoiser (FDFD) module, allowing it to effectively clone the reference's speaking characteristics. Crucially, we design the model with separate prediction heads for the probability velocity of distinct speech aspects, specifically prosody and acoustic details, allowing it to simultaneously learn multiple attribute-specific distributions within a unified flow architecture. Our main contributions are as follows:

- We introduce DiFlow-TTS, a novel zero-shot TTS framework that learns probability flows directly in the discrete space of factorized codec tokens, establishing an initial baseline for applying DFM to speech generation.

- Unlike prior DFM applications that operate on homogeneous discrete sequences, we propose a reformulated DFM framework over a structured, factorized representation by introducing the Factorized Discrete Flow Denoiser. This design explicitly models individual speech attributes through a flow-prediction mechanism with dedicated heads for prosody and acoustic details. To our knowledge, this is the first work to decompose probability velocity fields across multiple discrete subspaces within a single discrete flow process.

- We show that DiFlow-TTS outperforms baseline models in naturalness, content accuracy, and prosody preservation, while retaining a compact architecture that is up to $11.7\times$ smaller and delivering low-latency inference with up to $34\times$ speedup over baselines, thereby making it suitable for deployment in resource-constrained and latency-critical environments.

2 Related Work

A growing trend in speech synthesis focuses on converting raw waveforms into discrete token representations using vector-quantized variational autoencoders (VQ-VAE), which was first introduced by (van den Oord et al., 2017) in the field of computer vision and later adapted to speech processing (Baevski et al., 2020; Hsu et al., 2021). These tokenized representations have demonstrated greater naturalness and robustness compared to conventional mel-spectrogram-based approaches. To effectively model sequences of discrete speech tokens, recent efforts have adapted large language models (LLMs) from the natural language processing domain (Zhang et al., 2023; Chen et al., 2024a; Han et al., 2024; Du et al., 2024b; Peng et al., 2024; Meng et al., 2025; Chen et al., 2025; Wang et al., 2025b). A notable example is VALL-E (Chen et al., 2025), which leverages a pre-trained neural codec to encode speech into discrete codec tokens and reformulates zero-shot TTS as a conditional codec language modeling task. During inference, it performs autoregressive continuation from the acoustic tokens of a speech prompt, enabling high-fidelity speaker-consistent voice synthesis.

Although AR models achieve impressive quality, they are inherently limited by slow inference speeds. This limitation has prompted a shift toward NAR paradigms (Shen et al., 2024; Ju et al., 2024; Du et al., 2024a; Lee et al., 2025; Jia et al., 2025). For example, NaturalSpeech 2 (Shen et al., 2024) uses diffusion (Ho et al., 2020; Song et al., 2021) to generate discrete acoustic tokens as continuous features. Its successor, NaturalSpeech 3 (Ju et al., 2024), further factorizes speech into subspaces of content, prosody, and acoustic details, employing multiple diffusion models to independently capture various acoustic characteristics. In parallel, flow matching (Lipman et al., 2023; Liu et al., 2023) has gained attention as a promising generative technique, producing strong results in various domains.

Yet, most existing speech-related flow matching applications operate in continuous spaces, using either mel-spectrogram representations or continuous representations derived from discrete tokens, and predict flows over these representations rather than directly over discrete representations. (Mehta et al., 2024; Guan et al., 2024; Yao et al., 2025; Zuo et al., 2025b,a; Hieu et al., 2025).

An emerging line of research seeks to extend iterative refinement techniques to discrete spaces by modeling generation dynamics with Markov chains. Discrete-space generative models have already proven effective in domains such as natural language (Lou et al., 2024; Shi et al., 2024; Sahoo et al., 2024), proteins (Campbell et al., 2024; Yi et al., 2025), vision (Austin et al., 2021; Chang et al., 2022; Shi et al., 2024; Fuest et al., 2025), code (Gat et al., 2024), and graphs (Qin et al., 2025). However, the application of DFM (Gat et al., 2024) to speech modeling, such as zero-shot TTS, remains largely unexplored. In this work, we propose a zero-shot TTS system with DFM, aiming to leverage the advantages of discrete modeling while preserving high-quality speech generation.

3 Methodology

Figure 2 shows the overall framework of DiFlow-TTS, which comprises three main modules: **(a)** *Speech Tokenizer*, **(b)** *Phoneme-Content Mapper*, and **(c)** *Factorized Discrete Flow Denoiser*. We detail each module in the following sections.

3.1 Preliminaries

Notation. Let a sequence x be an array of L tokens $(x^1, x^2, \ldots, x^L)$ drawn from a discrete vocabulary of size v, i.e., $x \in \mathcal{D} = [v]^L$ with $[v] = \{1, \ldots, v\}$. We further define the extended space $\mathcal{D}' = [v]^{nL}$ for the concatenation of n such sequences. To represent the point mass distributions over these sequences, we use the delta function $\delta_y(x) = \prod_{i=1}^{L} \delta_{y^i}(x^i)$, where $y \in \mathcal{D}, \delta_{y^i}(x^i) = 1$ if $x^i = y^i$, and 0 otherwise.

Discrete Flow Matching. We adopt DFM as the generative backbone for codec token modeling. The goal is to transport source samples $\mathbf{x}_0 \sim p$ to target samples $\mathbf{x}_1 \sim q$. Concretely, we instantiate the source distribution with all-mask tokens, while the target distribution is factorized into prosodic and acoustic components, enabling structured joint learning. During training, we employ a scheduler $\kappa_t \in [0, 1]$, a monotonically increasing function with boundary conditions $\kappa_0 = 0$ and $\kappa_1 = 1$, where $t \in [0, 1]$ denotes continuous time. This scheduler controls the interpolation, gradually shifting the distribution from source to target as κ_t increases. We follow the deterministic convex-interpolant assumption, under which the marginal path distribution is given by $p_t(\mathbf{x}^i|\mathbf{x}_t) = \delta_{\mathbf{x}_t}(\mathbf{x}^i)$. Following (Gat et al., 2024), we then construct a conditional probability path, referred to as the *mixture path*, which linearly interpolates between the source and target distributions: $p_t(\mathbf{x}^i|\mathbf{x}_0, \mathbf{x}_1) = (1-\kappa_t)\delta_{\mathbf{x}_0}(\mathbf{x}^i) + \kappa_t \delta_{\mathbf{x}_1}(\mathbf{x}^i)$. This formulation leads to a *conditional probability path*, which is governed by the probability velocity $\mathbf{u}_t$ defined as:

$$\mathbf{u}_t^i(\mathbf{x}^i, \mathbf{x}_t) = \frac{\dot{\kappa}_t}{1 - \kappa_t} \left[p_{1|t}(\mathbf{x}^i|\mathbf{x}_t, \mathbf{c}; \theta) - \delta_{\mathbf{x}_t}(\mathbf{x}^i) \right], \tag{1}$$

where $\dot{\kappa}_t$ is the time derivative of the scheduler κ_t, θ denotes learnable parameters of a probability denoiser, $p_{1|t}(\cdot|\mathbf{x}_t, \mathbf{c}; \theta)$ is the posterior distribution $\mathbf{x}_1$ given a partially corrupted sequence $\mathbf{x}_t$ and $\mathbf{c}$ representing a set of multimodal conditioning inputs. More details are provided in Appendix B.1.

3.2 Speech Tokenizer

The Speech Tokenizer module (Figure 2a) converts a raw input speech waveform into distinct token sequences. For this process, we employ FACodec (Ju et al., 2024), which factorizes the original speech signal $\mathbf{r}$ into disentangled token sequences representing prosody, content, and acoustic details and extracts the speaker identity:

$$\mathbf{r}^p, \mathbf{r}^c, \mathbf{r}^a, \mathbf{s} = \text{CodecEncoder}(\mathbf{r}), \tag{2}$$

where $\mathbf{r}^p \in [v]^{mL}$, $\mathbf{r}^c \in [v]^{nL}$, and $\mathbf{r}^a \in [v]^{kL}$ denote the token sequences for prosody, content, and acoustic details, respectively, and $\mathbf{s} \in \mathbb{R}^{D_{\text{spk}}}$ is the speaker embedding. Here, L is the token sequence length, while m, n, and k denote the number of residual vector quantization (RVQ) codebooks for prosody, content, and acoustic details, respectively, each with a vocabulary size of v. More details are provided in Appendix B.2.

3.3 Phoneme-Content Mapper

The PCM module (Figure 2b) aligns phoneme sequences derived from the text prompt with discrete content tokens. While its overall structure is inspired by conventional duration-based alignment mechanisms (Ren et al., 2021), we reformulate the alignment process to operate directly on

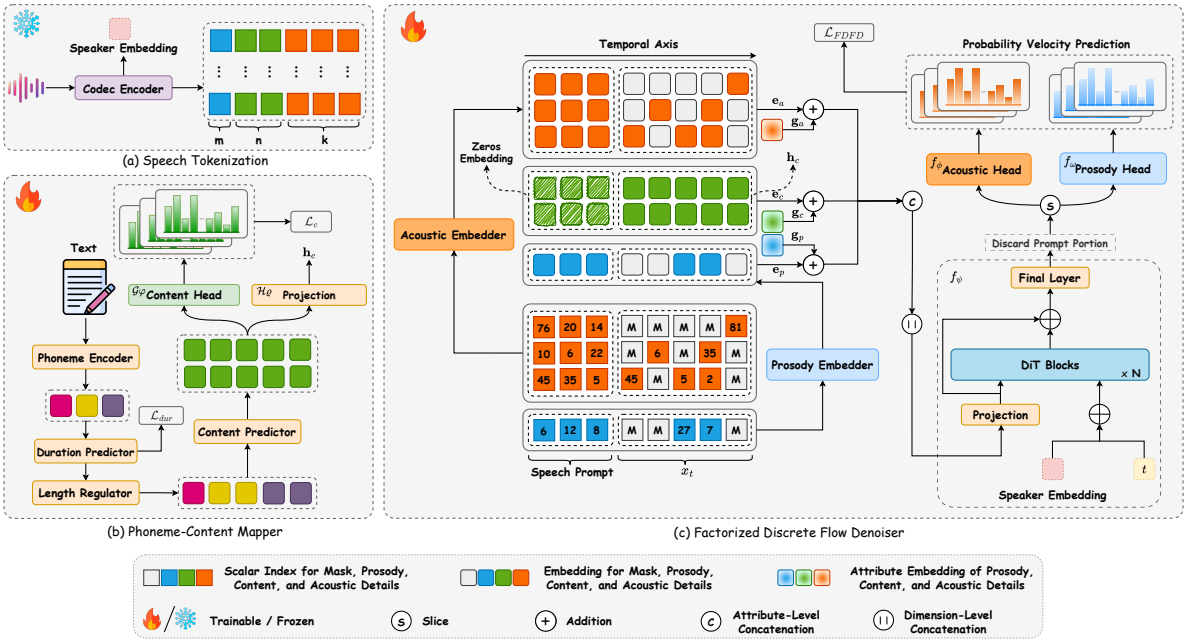

Figure 2: The detailed architecture of DiFlow-TTS comprises three main components: **(a)** *Speech Tokenizer*, which extracts factorized discrete tokens and a speaker embedding from a raw speech; **(b)** *Phoneme-Content Mapper*, which maps input phonemes to discrete content tokens and generates the corresponding content embeddings; and **(c)** *Factorized Discrete Flow Denoiser*, which performs discrete flow matching conditioned on the content embeddings, speaker embedding, and the discrete prosody and acoustic tokens derived from the reference speech prompt.

discrete codec tokens rather than continuous mel-spectrogram frames.

Given a text prompt, we convert it into a phoneme sequence and extract phoneme embeddings $\mathbf{p} \in \mathbb{R}^{N \times D}$ using a phoneme encoder, where N and D denote the sequence length and hidden dimension, respectively. To align phonemes with discrete speech tokens, a *Duration Predictor* estimates durations, indicating how many speech tokens correspond to each phoneme. This produces an integer-based alignment that maps each phoneme to a variable-length span in the speech-token sequence. Using these alignments, the *Length Regulator* upsamples phoneme embeddings based on the ground-truth (during training) or predicted (during inference) phoneme durations. The upsampled sequence, whose length is L, is then passed to the *Content Predictor* (Figure 6), which consists of multiple Feed-Forward Transformer (FFT) layers. These layers hierarchically extract n content representations, producing hidden states $\mathbf{h} \in \mathbb{R}^{n \times L \times D}$, which are then processed by two branches: a projection layer $\mathcal{H}_\varrho(\cdot)$ produces content embeddings, and a content head $\mathcal{G}_\varphi(\cdot)$ that outputs logits over a vocabulary of size v:

$$\begin{aligned} \mathbf{h}_c &= \mathcal{H}_\varrho(\mathbf{h}) \in \mathbb{R}^{n \times L \times D}, \\ p(\mathbf{x}^c|\mathbf{h};\varphi) &= \text{Softmax}(\mathcal{G}_\varphi(\mathbf{h})) \in \mathbb{R}^{n \times L \times v}. \end{aligned} \tag{3}$$

3.4 Factorized Discrete Flow Denoiser

The FDFD (Figure 2c) aims to generate the prosody and acoustic sequences of the synthesized speech by leveraging DFM and in-context learning, conditioned on a set of contextual inputs. In the following, we detail the key elements of this module.

Contextual Modeling. We now elaborate on the construction of the conditioning context $\mathbf{c}$ introduced in Eq. (1) and describe how it is integrated into our framework.

❶ **Inputs:** Given a reference speech prompt $\mathbf{r}$, we apply the operation in Eq. (2) to extract prosody (p) token sequences $\mathbf{r}^p \in [v]^{mL_r}$, acoustic ($a$) token sequences $\mathbf{r}^a \in [v]^{kL_r}$, and a speaker embedding $\mathbf{s} \in \mathbb{R}^{D_{\text{spk}}}$, where L_r denotes the temporal length of the reference prompt and D_{spk} is the hidden dimension of the speaker embedding. The corrupted input at timestep t, $\mathbf{x}_t \in [v]^{(m+k)L}$, is split into prosody tokens $\mathbf{x}_t^p \in [v]^{mL}$ and acoustic tokens $\mathbf{x}_t^a \in [v]^{kL}$. We then employ specific embedders for prosody and acoustic to map these sequences to D-dimensional hidden representations $\mathbf{e}_r^i$ (**reference**) and $\mathbf{e}_t^i$ (**corrupted**) for $i \in \{p, a\}$.

❷ **Conditioning:** We concatenate the reference and corrupted embeddings along the temporal dimension for each attribute, where the reference embeddings provide contextual information for the

corrupted embeddings. We further introduce the content (c) embedding (Eq. (3)) as the corrupted content representation ($\mathbf{e}_t^c = \mathbf{h}_c$) to enrich prosody and acoustic modeling, while the reference content representation is replaced with zeros ($\mathbf{e}_r^c = \mathbf{0}$), since reference speech content information is not used. The concatenated embeddings for each attribute is given by $\mathbf{e}_i = \mathbf{e}_r^i \oplus \mathbf{e}_t^i \in \mathbb{R}^{j \times (L_r+L) \times D}$, where $\oplus$ denotes the concatenation operator, and $(i, j) \in \{(p, m), (c, n), (a, k)\}$.

❸ **Constructing unified embedding:** We first introduce a learnable attribute-type embedding $\mathbf{g}_i \in \mathbb{R}^{1\times1\times D}$ for each attribute $i \in \{p, c, a\}$, allowing the model to explicitly differentiate among the corresponding attributes. Each such embedding is added to the embeddings of its corresponding attribute, yielding $\tilde{\mathbf{e}}_i = \mathbf{e}_i + \mathbf{g}_i$. The resulting embeddings are concatenated along the attribute dimension, yielding the final unified embedding $\mathbf{e} = \tilde{\mathbf{e}}_p \oplus \tilde{\mathbf{e}}_c \oplus \tilde{\mathbf{e}}_a \in \mathbb{R}^{(m+n+k)\times(L_r+L)\times D}$.

Factorized Flow Prediction. We flatten $\mathbf{e}$ and project it to $\mathbf{z} \in \mathbb{R}^{(L_r+L)\times D}$, which is then processed by a neural network $f_\psi : \mathbb{R}^{(L_r+L)\times D} \to \mathbb{R}^{(L_r+L)\times(m+k)D}$ composed of Diffusion Transformer (DiT) blocks (Peebles and Xie, 2023), a long skip connection, and a final transformation layer. To achieve speaker adaptation, we form a global conditioning vector $\mathbf{c}_g$ by summing the projected speaker embedding $\mathbf{s}$ and timestep embedding $\mathbf{t}$. This vector modulates the DiT features via adaptive layer normalization (AdaLN) (Peebles and Xie, 2023). After processing the DiT blocks via AdaLN, we apply a long skip connection by adding $\mathbf{z}$ to the final DiT output, followed by the final transformation layer comprising layer normalization, AdaLN-based modulation conditioned on $\mathbf{c}_g$, and a linear projection to dimension $(m + k)D$. We then discard the reference portion and permute the result to yield the final hidden representation $\mathbf{h}_{p,a} \in \mathbb{R}^{(m+k)\times L\times D}$.

To effectively enable the model to jointly attend to information from different representation subspaces, we propose a factorized flow prediction mechanism based on multi-head prediction. In this design, FDFD simultaneously models multiple aspects of speech, specifically prosody and acoustic details. Formally, we define two parallel heads: the *prosody head* $f_\phi(\cdot)$ and the *acoustic head* $f_\omega(\cdot)$, which independently predict probability distributions corresponding to prosody and acoustic attributes. We first slice the representation $\mathbf{h}_{p,a}$ into two parts: the prosody representation $\mathbf{h}_p \in \mathbb{R}^{m\times L\times D}$ and the acoustic representation $\mathbf{h}_a \in \mathbb{R}^{k\times L\times D}$. Each part is processed by its respective head, $f_\phi(\cdot)$ and $f_\omega(\cdot)$, producing logits of the shapes $\mathbb{R}^{m\times L\times v}$ and $\mathbb{R}^{k\times L\times v}$, respectively. These logits correspond to the categorical distributions predicted over the discrete token vocabulary for each attribute. Finally, the two outputs are concatenated along the attribute dimension, yielding a unified tensor of shape $\mathbb{R}^{(m+k)\times L\times v}$ serving as the estimated posterior distribution over $\mathbf{x}_1$.

3.5 Training Objective

Our training objective integrates three loss components, one for each module in the framework. First, we optimize the *Duration Predictor* using the Mean Squared Error (MSE) loss on the logarithmic scale, denoted as $\mathcal{L}_{dur}$, which compares the predicted and ground-truth durations. Second, for the *Content Predictor* defined in Eq. (3), we use a cross-entropy loss $\mathcal{L}_c$ between the predicted logits and the discrete content tokens obtained from the ground truth. Third, for the *FDFD* module, we learn a probabilistic denoiser $p_{1|t}$ trained to recover masked tokens under varying masking ratios. The objective is to minimize the cross-entropy loss: $\mathcal{L}_{FDFD}(\theta) = -\sum_{i\in\mathcal{T}} \mathbb{E}_{t\sim\mathcal{U}[0,1],(\mathbf{x}_0,\mathbf{x}_1),\mathbf{x}_t} \left[\log p_{1|t}(\mathbf{x}_1^i|\mathbf{x}_t, \mathbf{c}; \theta)\right]$, where $\mathcal{T} = [(m + k)L]$, $\mathbf{x}_t \sim p_t(\mathbf{x}|\mathbf{x}_0, \mathbf{x}_1)$, $\mathbf{x}_0 \sim p$, and $\mathbf{x}_1 \sim q$. Finally, the total loss is defined as:

$$\mathcal{L} = \lambda_{dur}\mathcal{L}_{dur} + \lambda_c\mathcal{L}_c + \lambda_{FDFD}\mathcal{L}_{FDFD}, \quad (4)$$

where λ_{dur}, λ_c, and λ_{FDFD} are hyperparameters weighting the loss terms. Appendix B.4 further describes our training and inference pipeline.

4 Experiments

We conduct extensive experiments to analyze the effectiveness and behaviors of DiFlow-TTS. Implementation details are reported in Appendix A.

4.1 Experimental Setup

Baselines. To ensure a best-effort comparison under publicly available checkpoints and code with standard evaluation settings, we compare against publicly available baselines spanning different modeling paradigms: **(i)** *Autoregressive models:* VoiceCraft (Peng et al., 2024), VALL-E (Chen et al., 2025); **(ii)** *Continuous flow matching/diffusion models:* NaturalSpeech 2 (Shen et al., 2024), F5-TTS (Chen et al., 2024b), OZSpeech

Type	Model	Data (hours)	UTMOS ↑	WER ↓	SIM-O ↑	F0		Energy	
						Accuracy ↑	RMSE ↓	Accuracy ↑	RMSE ↓
-	Ground Truth	-	4.10	0.02	-	-	-	-	-
(i)	VoiceCraft [†] ACL'24	GS (9K)	3.55	0.18	0.51	0.78	17.22	0.44	0.010
	VALL-E [⋄] TASLPRO'25	LT (500)	3.68	0.19	0.40	0.75	21.66	0.36	0.020
(ii)	NaturalSpeech 2 [‡] ICLR'24	LT (585)	2.38	0.09	0.31	0.80	15.62	0.25	0.020
	F5-TTS [⋄] ACL'25	LT (500)	3.76	0.24	0.52	0.80	13.78	0.67	0.010
	F5-TTS [†] ACL'25	E (100K)	3.72	0.09	0.66	0.83	12.66	0.66	0.010
	OZSpeech [†] ACL'25	LT (500)	3.15	**0.05**	0.40	0.81	11.96	0.67	0.010
(iii)	MaskGCT [†] ICLR'25	E (100K)	3.83	0.09	**0.67**	0.77	14.33	**0.75**	**0.007**
(iv)	**DiFlow-TTS**	**LT (470)**	**3.98**	**0.05**	0.45	**0.88**	**7.97**	0.73	**0.007**

Table 1: Performance on the *LibriSpeech test-clean* dataset using 3-second audio prompts. [⋄] means reproduced results. [†] and [‡] mean results inferred from official and unofficial checkpoints, respectively. The best and second best are **bold** and underlined, respectively. WER is reported as a value in the range $[0, 1]$. Abbreviation: E (Emilia), GS (GigaSpeech), LT (LibriTTS).

Model	Naturalness ↑	Intelligibility ↑	Similarity ↑
Ground Truth	4.42 ± 0.12	4.54 ± 0.11	4.29 ± 0.14
VoiceCraft	3.94 ± 0.17	4.08 ± 0.18	4.17 ± 0.15
VALLE-E	3.71 ± 0.17	3.96 ± 0.17	3.99 ± 0.15
NaturalSpeech 2	2.62 ± 0.20	3.25 ± 0.21	2.63 ± 0.18
F5-TTS	3.97 ± 0.17	4.16 ± 0.14	4.07 ± 0.16
OZSpeech	2.80 ± 0.23	3.42 ± 0.24	3.20 ± 0.22
MaskGCT	3.97 ± 0.16	4.14 ± 0.15	4.17 ± 0.15
DiFlow-TTS	$\mathbf{4.18 \pm 0.16}$	$\mathbf{4.41 \pm 0.13}$	$\mathbf{4.42 \pm 0.12}$

Table 2: MOS evaluation with 3-second audio prompts, including 95% confidence intervals. The best and second best are **bolded** and underlined, respectively.

(Hieu et al., 2025); **(iii)** *Masked generative model:* MaskGCT (Wang et al., 2025c). **(iv)** *Discrete flow matching model:* DiFlow-TTS. We refer readers to Appendix C.3 for detailed descriptions of all baselines, including their architectures, training data, checkpoints, and evaluation settings.

Dataset. We use a 470-hour subset of the *LibriTTS* dataset (Zen et al., 2019), which comprises multi-speaker English audio recordings, to train our method. For evaluation, we utilize the LibriSpeech test-clean dataset (Panayotov et al., 2015). Additional details are provided in Appendix C.1.

Evaluation Metrics. To evaluate model performance, we use a suite of *objective evaluation* metrics that target multiple aspects of speech synthesis. Naturalness and speech quality is measured with UTMOS, speaker similarity is assessed using SIM-O, content accuracy is quantified via the word error rate (WER), and prosody reconstruction is analyzed through pitch- and energy-based metrics. In addition, inference latency is evaluated using the real-time factor (RTF). Detailed descriptions of these metrics are provided in Appendix C.2.

Complementing these objective measures, we perform a *subjective evaluation* following the Mean Opinion Score (MOS) protocol. Specifically, 30 listeners rate the synthesized speech on a scale from 1 to 5 with respect to naturalness, intelligibility, and speaker similarity to the speech prompt.

4.2 Main Results

Comparison Results. Table 1 presents the performance of DiFlow-TTS with 128 function evaluations (NFE) using 3-second audio prompts, compared to baseline methods. For naturalness and speech quality as measured by UTMOS, DiFlow-TTS achieves the strongest performance despite being trained on only 470 hours of speech data, which is significantly smaller (by a factor of $1.1\times$ to $212.8\times$) than other baselines, highlighting the strength of our FDFD module in capturing prosodic and acoustic nuances even under limited data conditions. For content accuracy, DiFlow-TTS, along with OZSpeech, achieves SOTA performance in terms of WER, demonstrating the effectiveness of our method in producing speech with accurate linguistic content. For speaker similarity, DiFlow-TTS offers no clear advantage over baselines, likely due to its simple speaker conditioning in DiT blocks, which could be improved with more advanced strategies. **We highlight this limitation as a promising direction for future work beyond the scope of this study**. For prosody reconstruction, DiFlow-TTS outperforms across all metrics, with the sole exception of energy accuracy, where it trails MaskGCT by only 0.02, despite MaskGCT being trained on significantly more data (100K hours). These findings further confirm the ability of the FDFD module to model fine-grained prosodic attributes with high fidelity. To gain further insight into speech quality, we report the subjective MOS

Model	#Params ↓	NFE	RTF ↓	UTMOS ↑	WER ↓	SIM-O ↑	RMSE_{F0} ↓	RMSE_{E} ↓
VoiceCraft	830M	-	1.70	3.55	0.18	0.51	17.22	0.010
VALL-E	594M	-	0.86	3.68	0.19	0.40	21.66	0.020
NaturalSpeech 2	378M	200	1.66	2.38	0.09	0.31	15.62	0.020
F5-TTS	336M	32	0.26	3.72	0.09	0.66	12.66	0.010
OZSpeech	145M	1	**0.03**	3.15	**0.05**	0.40	11.96	0.010
MaskGCT	1.43B	50 + 45†	0.46	3.83	0.09	**0.67**	14.33	**0.007**
DiFlow-TTS-Small	**122M**	4	**0.03**	3.34	0.06	0.43	8.31	**0.007**
		16	0.05	**3.89**	**0.05**	0.45	8.58	0.008
DiFlow-TTS	164M	4	**0.03**	3.31	**0.05**	0.44	8.05	**0.007**
		16	0.07	3.86	**0.05**	0.45	**7.96**	**0.007**

† MaskGCT is a two-stage system that first predicts masked semantic tokens, then uses them to infer masked acoustic tokens.

Table 3: Comparison of model size and latency. The **#Params** exclude the neural codec or vocoder component, which is non-trainable. The best and second best are **bold** and underlined, respectively.

Model	UTMOS ↑	WER ↓	SIM-O ↑	F0		Energy	
				Accuracy ↑	RMSE ↓	Accuracy ↑	RMSE ↓
DiFlow-TTS	3.978	**0.048**	**0.454**	**0.884**	**7.972**	**0.735**	**0.007**
- w/o Attribute Embedding	**3.983**	0.060	0.444	0.869	9.289	0.712	0.008
- w/o Speaker Embedding	3.902	0.057	0.378	0.681	20.868	0.615	0.010
- w/o Content Embedding	3.077	0.063	0.333	0.867	8.878	0.698	0.008
- w/o Multi-head Prediction	3.939	0.057	0.442	0.876	8.474	0.726	**0.007**

Table 4: Ablation study results showing the effect of removing each component from the DiFlow-TTS, with NFE set to 128. The best and second best are **bold** and underlined, respectively.

evaluations in Table 2. Overall, DiFlow-TTS consistently outperforms SOTA methods across every MOS dimension, providing strong evidence of its well-balanced performance in generating natural and intelligible speech with high speaker similarity. It is worth noting that even though DiFlow-TTS ranks third on SIM-O (see Table 1), an embedding-space proxy that may penalize artifacts inaudible to humans, it best captures the perceptual identity cues (e.g., pitch, timbre, prosody) that human listeners value, indicating superior speaker faithfulness where it matters most. These results are especially notable given the model's training data efficiency. For the other MOS metrics, the results are consistent with the findings in Table 1.

Model Size & Latency Analysis. Table 3 compares the model size and latency between DiFlow-TTS and the baselines for the 3-second audio prompt setting. The RTF metric, measured in seconds, shows that all baselines except OZSpeech experiences latency in the order of hundreds of miliseconds. In contrast, DiFlow-TTS, along with OZSpeech, has latency that is an order of magnitude smaller. To highlight efficiency, we further construct a smaller variant of DiFlow-TTS (denoted as *DiFlow-TTS-Small*) by reducing the number of attention heads $(12 \rightarrow 8)$ and DiT layers $(12 \rightarrow 8)$, resulting in a 122M-parameter model. This small variant achieves the best results in both speed and size. For comparison, OZSpeech, optimized for the 1-NFE setting, achieves the same RTF (0.03) as DiFlow-TTS-Small with 4 NFEs, yet delivers significantly lower performance across all metrics except for WER, while DiFlow-TTS with 4 NFEs (RTF = 0.03) also achieves comparable performance to our small variant. Furthermore, DiFlow-TTS-Small with 16 NFEs (RTF = 0.05) achieves competitive performance in naturalness, intelligibility, speaker similarity, and prosody error while being only marginally slower than OZSpeech (by 0.02s in RTF) yet 5.2× to 34.0× faster than the other baselines, with a model size 1.2× to 11.7× smaller than all baselines. The results of DiFlow-TTS with 16 NFEs further reinforce these findings, demonstrating a strong balance between model size, speed, and speech quality.

4.3 Ablation Studies and Analyses

Effect of Each Component. To assess the impact of each component in DiFlow-TTS, we perform an ablation study by systematically removing or modifying key elements: **(1)** removing the attribute-type embeddings used to distinguish prosody, content, and acoustic streams; **(2)** excluding the speaker embedding from the conditioning process (i.e., not in-

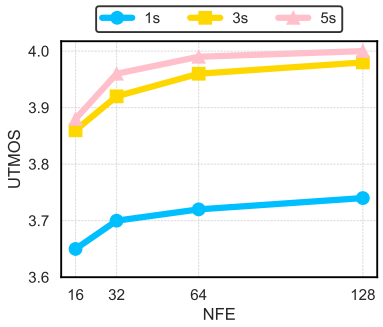

Figure 3: UTMOS vs. NFE for different prompt durations.

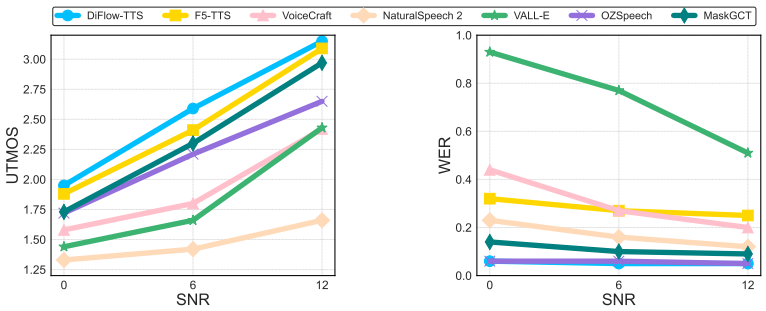

Figure 4: Performance across different SNR levels in terms of UTMOS (left) and WER (right).

jecting it into the DiT blocks); **(3)** disabling the use of content embeddings in the FDFD module; and **(4)** replacing the multi-head prediction architecture with a single-head prediction. As shown in Table 4, we observe a slight degradation in all metrics except UTMOS when the attribute-type embeddings are removed. This suggests that while these embeddings enhance overall fidelity and prosody modeling, they may introduce minor redundancies that subtly affect perceived naturalness. A more pronounced decline in speaker similarity and prosody-related metrics is observed when speaker embedding is excluded from the FDFD module. This highlights that prosody is not only content-dependent but also strongly influenced by speaker identity; without speaker conditioning, the FDFD module produces extraneous prosodic variations, resulting in reduced speaker adaptation and overall synthesis quality. When content embeddings from the PCM branch are removed from FDFD, we observe substantial degradation across metrics related to naturalness and speaker similarity. This demonstrates the critical role of content embeddings in conditioning FDFD to generate appropriate prosody and support speaker adaptation. Lastly, replacing the multi-head prediction mechanism with a single-head alternative leads to minor performance drops across all metrics, indicating that the multi-head design enhances prediction diversity and robustness in prosody and acoustic modeling.

Prompt Duration Analysis. To gain further insight into model behavior, Figure 3 illustrates the relationship between UTMOS, which strongly correlates with human perceptual evaluations, and NFE across different prompt durations. Overall, longer prompts lead to higher UTMOS scores, indicating improved reconstruction quality and a greater sensitivity of the model to prompt length. Additionally, increasing the NFE from 16 to 128 consistently improves performance for all prompt durations. In particular, the highest performance is achieved with a 5-second prompt and an NFE of 128.

Noisy Prompt Analysis. We analyze the robustness of DiFlow-TTS under noisy audio prompts using UTMOS and WER, a challenging scenario since most models are trained on clean speech. The noisy prompts are generated from the *LibriSpeech test-clean* set with additive noise augmentation. As shown in Figure 4, all models are highly sensitive to noise, showing sharp degradation in both UTMOS (left) and WER (right) as the prompt SNR decreases (see Table 1 for clean-prompt reference). DiFlow-TTS, however, consistently achieves the highest UTMOS across all noise levels, demonstrating its ability to synthesize high-fidelity speech under noisy conditions. For WER, it shows little to no degradation across SNR levels, a trend also observed in OZSpeech, while other baselines suffer significant performance drops.

5 Conclusion

We introduce DiFlow-TTS, a novel zero-shot TTS framework that brings discrete flow matching into the domain of speech generation. By combining a PCM for accurate content modeling with a FDFD that separately models prosody and acoustic attributes, DiFlow-TTS achieves strong performance in naturalness, intelligibility, prosody, and inference efficiency, while exhibiting limitations in speaker similarity as reflected by comprehensive objective and subjective evaluations. These results establish DiFlow-TTS as a compelling solution for efficient, high-fidelity zero-shot speech synthesis, well-suited to resource-constrained and latency-sensitive applications, and highlight discrete flow models as a promising direction for future generative speech research.

Limitations

Our study investigates the viability of discrete flow matching for speech synthesis. Although our proposed method substantially improves the quality of generated speech, it still exhibits several limitations. In particular, the current design does not fully meet expectations in terms of voice cloning quality, we acknowledge that our current strategy for speaker conditioning is not yet optimal. Since our framework relies on FACodec (Ju et al., 2024), which explicitly disentangles speech attributes and speaker identity, integrating it effectively requires additional mechanisms to inject speaker information. In contrast, codecs such as EnCodec (Défossez et al., 2023) embed speaker information directly within their vector quantizers, enabling models to implicitly learn speaker characteristics. In future work, we plan to extend our framework to support alternative codec models or to replace global AdaLN conditioning with a cross-attention mechanism specifically designed for timbre embeddings, allowing the model to capture local speaker characteristics more effectively. Developing more robust methods for reproducing speaker timbre from real-world audio in discrete settings, particularly in zero-shot TTS scenarios, remains an open research direction.

Potential Risks

Our work focuses on advancing text-to-speech technology, which, while beneficial, carries potential risks of misuse such as voice spoofing, impersonation, or spreading misleading content. To ensure ethical compliance, all experiments are conducted exclusively on publicly available datasets with appropriate licenses, where speakers have explicitly consented to their voices being used for research. No private or unauthorized data are employed. We acknowledge that the ability to closely mimic a speaker's voice raises important concerns regarding privacy, security, and trust. To mitigate these risks, it is essential to pair progress in TTS with robust detection systems for synthetic speech and to establish mechanisms for reporting and addressing suspected misuse.

References

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. 2021. Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems*.

Alexei Baevski, Steffen Schneider, and Michael Auli. 2020. vq-wav2vec: Self-supervised learning of discrete speech representations. In *International Conference on Learning Representations*.

Andrew Campbell, Jason Yim, Regina Barzilay, Tom Rainforth, and Tommi S. Jaakkola. 2024. Generative flows on discrete state-spaces: Enabling multimodal flows with applications to protein co-design. In *ICML*.

Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T. Freeman. 2022. Maskgit: Masked generative image transformer. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Guoguo Chen, Shuzhou Chai, Guan-Bo Wang, Jiayu Du, Wei-Qiang Zhang, Chao Weng, Dan Su, Daniel Povey, Jan Trmal, Junbo Zhang, Mingjie Jin, Sanjeev Khudanpur, Shinji Watanabe, Shuaijiang Zhao, Wei Zou, Xiangang Li, Xuchen Yao, Yongqing Wang, Zhao You, and Zhiyong Yan. 2021. Gigaspeech: An evolving, multi-domain asr corpus with 10,000 hours of transcribed audio. In *Interspeech 2021*, pages 3670–3674.

Sanyuan Chen, Shujie Liu, Long Zhou, Yanqing Liu, Xu Tan, Jinyu Li, Sheng Zhao, Yao Qian, and Furu Wei. 2024a. Vall-e 2: Neural codec language models are human parity zero-shot text to speech synthesizers. *Preprint*, arXiv:2406.05370.

Sanyuan Chen, Chengyi Wang, Yu Wu, Ziqiang Zhang, Long Zhou, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, Lei He, Sheng Zhao, and Furu Wei. 2025. Neural codec language models are zero-shot text to speech synthesizers. *IEEE Transactions on Audio, Speech and Language Processing*, pages 1–15.

Yushen Chen, Zhikang Niu, Ziyang Ma, Keqi Deng, Chunhui Wang, Jian Zhao, Kai Yu, and Xie Chen. 2024b. F5-tts: A fairytaler that fakes fluent and faithful speech with flow matching. *Preprint*, arXiv:2410.06885.

Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. 2023. High fidelity neural audio compression. *Transactions on Machine Learning Research*. Featured Certification, Reproducibility Certification.

Chenpeng Du, Yiwei Guo, Feiyu Shen, Zhijun Liu, Zheng Liang, Xie Chen, Shuai Wang, Hui Zhang, and Kai Yu. 2024a. Unicats: A unified context-aware text-to-speech framework with contextual vq-diffusion and vocoding. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(16):17924–17932.

Zhihao Du, Yuxuan Wang, Qian Chen, Xian Shi, Xiang Lv, Tianyu Zhao, Zhifu Gao, Yexin Yang, Changfeng Gao, Hui Wang, and 1 others. 2024b. Cosyvoice 2: Scalable streaming speech synthesis with large language models. *arXiv preprint arXiv:2412.10117*.

Michael Fuest, Vincent Tao Hu, and Björn Ommer. 2025. Maskflow: Discrete flows for flexible and efficient long video generation. *arXiv preprint arXiv:2502.11234*.

Itai Gat, Tal Remez, Neta Shaul, Felix Kreuk, Ricky T. Q. Chen, Gabriel Synnaeve, Yossi Adi, and Yaron Lipman. 2024. Discrete flow matching. In *Advances in Neural Information Processing Systems*, volume 37, pages 133345–133385. Curran Associates, Inc.

Wenhao Guan, Qi Su, Haodong Zhou, Shiyu Miao, Xingjia Xie, Lin Li, and Qingyang Hong. 2024. Reflow-tts: A rectified flow model for high-fidelity text-to-speech. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 10501–10505. IEEE.

Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and Ruoming Pang. 2020. Conformer: Convolution-augmented transformer for speech recognition. In *Interspeech 2020*, pages 5036–5040.

Zhifang Guo, Yichong Leng, Yihan Wu, Sheng Zhao, and Xu Tan. 2022. Prompttts: Controllable text-to-speech with text descriptions. *Preprint*, arXiv:2211.12171.

Bing Han, Long Zhou, Shujie Liu, Sanyuan Chen, Lingwei Meng, Yanming Qian, Yanqing Liu, Sheng Zhao, Jinyu Li, and Furu Wei. 2024. Vall-e r: Robust and efficient zero-shot text-to-speech synthesis via monotonic alignment. *arXiv preprint arXiv:2406.07855*.

Haorui He, Zengqiang Shang, Chaoren Wang, Xuyuan Li, Yicheng Gu, Hua Hua, Liwei Liu, Chen Yang, Jiaqi Li, Peiyang Shi, Yuancheng Wang, Kai Chen, Pengyuan Zhang, and Zhizheng Wu. 2024. Emilia: An extensive, multilingual, and diverse speech dataset for large-scale speech generation. In *2024 IEEE Spoken Language Technology Workshop (SLT)*, pages 885–890.

Nghia Huynh Nguyen Hieu, Ngoc Son Nguyen, Huynh Nguyen Dang, Thieu Vo, Truong-Son Hy, and Van Nguyen. 2025. OZSpeech: One-step zero-shot speech synthesis with learned-prior-conditioned flow matching. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 21500–21517, Vienna, Austria. Association for Computational Linguistics.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851. Curran Associates, Inc.

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. 2021. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, 29:3451–3460.

Shengpeng Ji, Ziyue Jiang, Hanting Wang, Jialong Zuo, and Zhou Zhao. 2024a. MobileSpeech: A fast and high-fidelity framework for mobile zero-shot text-to-speech. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13588–13600, Bangkok, Thailand. Association for Computational Linguistics.

Shengpeng Ji, Jialong Zuo, Minghui Fang, Ziyue Jiang, Feiyang Chen, Xinyu Duan, Baoxing Huai, and Zhou Zhao. 2024b. Textrolspeech: A text style control speech corpus with codec language text-to-speech models. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 10301–10305.

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

Zeqian Ju, Yuancheng Wang, Kai Shen, Xu Tan, Detai Xin, Dongchao Yang, Eric Liu, Yichong Leng, Kaitao Song, Siliang Tang, Zhizheng Wu, Tao Qin, Xiangyang Li, Wei Ye, Shikun Zhang, Jiang Bian, Lei He, Jinyu Li, and Sheng Zhao. 2024. NaturalSpeech 3: Zero-shot speech synthesis with factorized codec and diffusion models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 22605–22623. PMLR.

Keon Lee, Dong Won Kim, Jaehyeon Kim, Seungjun Chung, and Jaewoong Cho. 2025. DiTTo-TTS: Diffusion transformers for scalable text-to-speech without domain-specific factors. In *The Thirteenth International Conference on Learning Representations*.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. 2023. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*.

Xingchao Liu, Chengyue Gong, and qiang liu. 2023. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations*.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. 2022. A convnet for the 2020s. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11966–11976.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Aaron Lou, Chenlin Meng, and Stefano Ermon. 2024. Discrete diffusion modeling by estimating the ratios

of the data distribution. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 32819–32848. PMLR.

Michael McAuliffe, Michaela Socolof, Sarah Mihuc, Michael Wagner, and Morgan Sonderegger. 2017. Montreal forced aligner: Trainable text-speech alignment using kaldi. In *Interspeech 2017*, pages 498–502.

Shivam Mehta, Ruibo Tu, Jonas Beskow, Éva Székely, and Gustav Eje Henter. 2024. Matcha-tts: A fast tts architecture with conditional flow matching. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 11341–11345.

Lingwei Meng, Long Zhou, Shujie Liu, Sanyuan Chen, Bing Han, Shujie Hu, Yanqing Liu, Jinyu Li, Sheng Zhao, Xixin Wu, Helen M. Meng, and Furu Wei. 2025. Autoregressive speech synthesis without vector quantization. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1287–1300, Vienna, Austria. Association for Computational Linguistics.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: An asr corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5206–5210.

William Peebles and Saining Xie. 2023. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205.

Puyuan Peng, Po-Yao Huang, Shang-Wen Li, Abdelrahman Mohamed, and David Harwath. 2024. VoiceCraft: Zero-shot speech editing and text-to-speech in the wild. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12442–12462, Bangkok, Thailand. Association for Computational Linguistics.

Yiming Qin, Manuel Madeira, Dorina Thanou, and Pascal Frossard. 2025. Defog: Discrete flow matching for graph generation. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*.

Yi Ren, Chenxu Hu, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu. 2021. Fastspeech 2: Fast and high-quality end-to-end text to speech. In *International Conference on Learning Representations*.

Takaaki Saeki, Detai Xin, Wataru Nakata, Tomoki Koriyama, Shinnosuke Takamichi, and Hiroshi Saruwatari. 2022. Utmos: Utokyo-sarulab system for voicemos challenge 2022. *Preprint*, arXiv:2204.02152.

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

Kai Shen, Zeqian Ju, Xu Tan, Eric Liu, Yichong Leng, Lei He, Tao Qin, sheng zhao, and Jiang Bian. 2024. Naturalspeech 2: Latent diffusion models are natural and zero-shot speech and singing synthesizers. In *The Twelfth International Conference on Learning Representations*.

Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis Titsias. 2024. Simplified and generalized masked diffusion for discrete data. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Yakun Song, Zhuo Chen, Xiaofei Wang, Ziyang Ma, and Xie Chen. 2024. Ella-v: Stable neural codec language modeling with alignment-guided sequence reordering. *Preprint*, arXiv:2401.07333.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. 2021. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. 2024. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. 2017. Neural discrete representation learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6309–6318, Red Hook, NY, USA. Curran Associates Inc.

Kaidi Wang, Wenhao Guan, Ziyue Jiang, Hukai Huang, Peijie Chen, Weijie Wu, Qingyang Hong, and Lin Li. 2025a. Discl-vc: Disentangled discrete tokens and in-context learning for controllable zero-shot voice conversion. *arXiv preprint arXiv:2505.24291*.

Xinsheng Wang, Mingqi Jiang, Ziyang Ma, Ziyu Zhang, Songxiang Liu, Linqin Li, Zheng Liang, Qixi Zheng, Rui Wang, Xiaoqin Feng, and 1 others. 2025b. Spark-tts: An efficient llm-based text-to-speech model with single-stream decoupled speech tokens. *arXiv preprint arXiv:2503.01710*.

Yuancheng Wang, Haoyue Zhan, Liwei Liu, Ruihong Zeng, Haotian Guo, Jiachen Zheng, Qiang Zhang, Xueyao Zhang, Shunsi Zhang, and Zhizheng Wu. 2025c. MaskGCT: Zero-shot text-to-speech with masked generative codec transformer. In *The Thirteenth International Conference on Learning Representations*.

Zhichao Wu, Qiulin Li, Sixing Liu, and Qun Yang. 2024. Dctts: Discrete diffusion model with contrastive learning for text-to-speech generation. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 11336–11340.

Dongchao Yang, Songxiang Liu, Rongjie Huang, Chao Weng, and Helen Meng. 2024. Instructtts: Modelling expressive tts in discrete latent space with natural language style prompt. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 32:2913–2925.

Dongchao Yang, Jianwei Yu, Helin Wang, Wen Wang, Chao Weng, Yuexian Zou, and Dong Yu. 2023. Diffsound: Discrete diffusion model for text-to-sound generation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:1720–1733.

Jixun Yao, Yang Yuguang, Yu Pan, Ziqian Ning, Jianhao Ye, Hongbin Zhou, and Lei Xie. 2025. Stablevc: Style controllable zero-shot voice conversion with conditional flow matching. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 25669–25677.

Kai Yi, Kiarash Jamali, and Sjors HW Scheres. 2025. All-atom inverse protein folding through discrete flow matching. In *Forty-second International Conference on Machine Learning*.

Heiga Zen, Viet Dang, Rob Clark, Yu Zhang, Ron J. Weiss, Ye Jia, Zhifeng Chen, and Yonghui Wu. 2019. Libritts: A corpus derived from librispeech for text-to-speech. In *Interspeech 2019*, pages 1526–1530.

Xueyao Zhang, Liumeng Xue, Yicheng Gu, Yuancheng Wang, Jiaqi Li, Haorui He, Chaoren Wang, Ting Song, Xi Chen, Zihao Fang, Haopeng Chen, Junan Zhang, Tze Ying Tang, Lexiao Zou, Mingxuan Wang, Jun Han, Kai Chen, Haizhou Li, and Zhizheng Wu. 2024. Amphion: An open-source audio, music and speech generation toolkit. In *IEEE Spoken Language Technology Workshop, SLT 2024*.

Ziqiang Zhang, Long Zhou, Chengyi Wang, Sanyuan Chen, Yu Wu, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, and 1 others. 2023. Speak foreign languages with your own voice: Cross-lingual neural codec language modeling. *arXiv preprint arXiv:2303.03926*.

Jialong Zuo, Shengpeng Ji, Minghui Fang, Ziyue Jiang, Xize Cheng, Qian Yang, Wenrui Liu, Guangyan Zhang, Zehai Tu, Yiwen Guo, and 1 others. 2025a. Enhancing expressive voice conversion with discrete pitch-conditioned flow matching model. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Jialong Zuo, Shengpeng Ji, Minghui Fang, Mingze Li, Ziyue Jiang, Xize Cheng, Xiaoda Yang, Chen Feiyang, Xinyu Duan, and Zhou Zhao. 2025b. Rhythm controllable and efficient zero-shot voice conversion via shortcut flow matching. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16203–16217, Vienna, Austria. Association for Computational Linguistics.

A Implementation Details

The FDFD uses a scheduler κ_t drawn from a family of cubic polynomials. In our implementation, we set $\kappa_t = t^2$. The module employs DiT blocks (Peebles and Xie, 2023) with a hidden size of 768, 12 layers, and 12 attention heads, further enhanced with rotary position embedding (RoPE) (Su et al., 2024). The number of quantizers used in FaCodec (Ju et al., 2024) is $m = 1$ for prosody, $n = 2$ for content, and $k = 3$ for acoustic tokens, each with a vocabulary size of 1024. Figure 5 illustrates the architecture of our DiT block (Peebles and Xie, 2023), which incorporates global conditioning through adaptive normalization. The global conditioning vector, formed by combining time and speaker embeddings, is processed by a Multi-Layer Perceptron (MLP) to generate scale and shift parameters that modulate the input in both the multi-head self-attention (MHSA) and feed-forward stages. The feed-forward network in DiT uses a width multiplier of 4, and the speaker embedding dimension is 256. The *Phoneme-to-Discrete Content Aligner*, shown in Figure 6, has a hidden dimension of 768 and integrates a variance adapter (Ren et al., 2021) with an encoder hidden size of 256, a filter size of 1024, a kernel size of 3, and a dropout rate of 0.5. It employs a hierarchical stack of Feed-Forward Transformer (FFT) blocks, where each level models dependencies conditioned on the outputs of the previous layer. Given phoneme embeddings, the model produces n contextual representations that capture progressively richer features through the stacked FFT layers. Both the text encoder and decoder used to generate content tokens and embeddings adopt the same FFT-based architecture, consisting of 2 layers, 4 attention heads, a hidden size of 256, an output dimension of 768, convolutional filter sizes of 1024 with kernel sizes $[9, 1]$, a dropout rate of 0.2, and a maximum sequence length of 5000.

We train the model on 4× NVIDIA A100 GPUs for 315K steps with a batch size of 16, using AdamW (Loshchilov and Hutter, 2019) with a learning rate of 1×10^{-4}, weight decay of 0.01, and 200K warm-up steps. For Eq. (4), the overall objective combines duration, content, and denois-

ing losses, weighted by $\lambda_{dur} = 0.5$, $\lambda_c = 1.0$, and $\lambda_{FDFD} = 1.0$, respectively.

B Method Details

B.1 Source and Target Distributions

In this section, we elaborate on the source and target distributions of DFT in our setting, as detailed in the following paragraphs.

Source Distribution. Following (Gat et al., 2024), we instantiate the source distribution p to assign all probability mass to sequences in which every token is the mask token `[MASK]`, that is, $p(x) = \delta_{\texttt{[MASK]}}(x)$. This implies that the source distribution places all probability mass in the sequence where every token is the mask token `[MASK]`.

Target Distribution. In conventional DFM settings, the target sequence $\mathbf{x}_1$ is treated as a monolithic sequence. In contrast, we propose to factorize $\mathbf{x}_1$ into two structured components that are learned jointly. This formulation allows us to construct a probability velocity over a structured target space composed of two parts. To this end, we define the target distribution q as follows:

Definition B.1. Let $\mathbf{x}_1^p \sim q_p$ and $\mathbf{x}_1^a \sim q_a$ denote the random variables corresponding to the prosody and acoustic details sequences, respectively. These sequences are in spaces $[v]^{mL}$ and $[v]^{kL}$. The full target sequence is then defined as $\mathbf{x}_1 = \mathbf{x}_1^p \oplus \mathbf{x}_1^a \in [v]^{(m+k)L}$, where $\oplus$ denotes the concatenation of the sequence. Assuming the independence between the two components, the joint target distribution is factorized as $q(x) = q_p(x^p) \cdot q_a(x^a)$, where $x = x^p \oplus x^a$.

B.2 Factorized Neural Speech Codec

The Factorized Neural Speech Codec (FACodec) (Ju et al., 2024) disentangles speech waveforms into distinct attributes, which are content, prosody, acoustic details, and timbre, enabling precise representation for zero shot text-to-speech (TTS) tasks. Given a speech input $x \in \mathbb{R}^C$, a speech encoder, implemented with convolutional blocks, transforms it into a pre-quantization latent representation:

$$h = \text{Encoder}(x) \in \mathbb{R}^{T \times D}, \tag{5}$$

where T represents the downsampled temporal dimension and D denotes the latent feature dimension.

Three factorized vector quantizers (FVQs), denoted $\mathcal{Q}_p$, $\mathcal{Q}_c$, and $\mathcal{Q}_a$ for prosody, content, and acoustic details, respectively, convert h into discrete token sequences. Each FVQ, defined as $\mathcal{Q}_i = \{q_i^j\}_{j=1}^{N_i}$ for $i \in \{p, c, a\}$, consists of N_i quantizers, where $q_i^j \in \mathbb{R}^d$ is the j-th quantizer with hidden dimension d and a codebook size of 1024. Specifically, $N_p = 1$, $N_c = 2$, and $N_a = 3$. These quantizers produce discrete codes:

$$z = g_p(h) \oplus g_c(h) \oplus g_a(h) \in \mathbb{R}^{T \times 6}, \tag{6}$$

where $g_p(h) \in \mathbb{R}^{T \times 1}$, $g_c(h) \in \mathbb{R}^{T \times 2}$, and $g_a(h) \in \mathbb{R}^{T \times 3}$ map the latent h to prosody, content, and acoustic detail tokens, respectively. The concatenated output z forms a unified representation of the speech attributes.

The timbre attribute is extracted by passing the hidden representation h through a series of Conformer blocks (Gulati et al., 2020), followed by a temporal pooling layer. This process yields a timbre-specific embedding $z_t \in \mathbb{R}^D$. Given both z and z_t, the neural codec decoder reconstructs the speech waveform as follows:

$$y = \text{CodecDecoder}(z, z_t). \tag{7}$$

Building upon the structure of 7, which accepts z and z_t as input and is pre-trained on a large-scale multi-speaker corpus to support robust zero-shot TTS, we propose a method to model and generate a six-dimensional sequence representation $\tilde{z} \in \mathbb{R}^{T \times 6}$. This representation is restricted to lie within the latent subspace of the pre-trained FACodec and is designed to encode prosody, content, and acoustic information in a manner aligned with z. Finally, $\tilde{z}$ is passed to f_{dec} along with the timbre embedding z_t to synthesize the output waveform $\tilde{y}$.

B.3 Reference Prompt Selection during Training

During training, a crucial step is selecting an appropriate speech prompt to condition the FDFD module. Specifically, we randomly sample an arbitrary segment whose length is 30% of the total temporal length of the ground-truth sequence. This segment serves as the reference prompt, ensuring that prosodic and acoustic characteristics are preserved to guide the FDFD module effectively.

B.4 Training and Inference Procedures

To provide a clearer understanding of DiFlow-TTS, we detail the algorithmic procedures for training and inference in Algorithms 1 and 2, respectively.

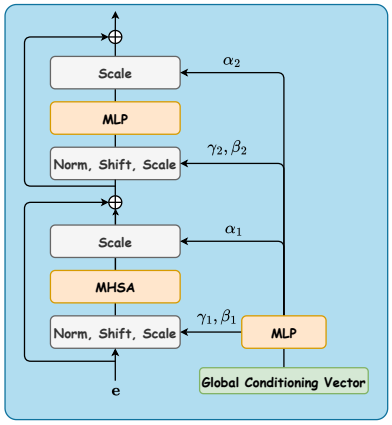

Figure 5: The detailed architecture of the DiT block.

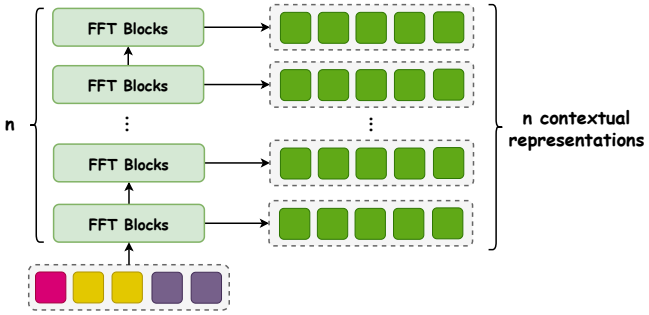

Figure 6: The detailed architecture of the content predictor.

C Evaluation Details

C.1 Dataset Details

Training Dataset. We preprocess the LibriTTS (Zen et al., 2019) dataset for training as follows. The silent segments at the beginning and end of each utterance are removed. We retain audio clips ranging from 1.0 to 16.6 seconds in duration that contain utterances with more than three words. From FACodec, we extract the ground-truth representations, which include a speaker embedding and six sequences of discrete tokens: one for prosody, two for content, and three for acoustic details in order. To obtain the ground truth of phoneme-level text and corresponding discrete speech tokens, we use the Montreal Forced Aligner (MFA) (McAuliffe et al., 2017) to align each audio with its target transcription, producing the duration of each phoneme in the audio. We then multiply these durations by 80, which represents the number of tokens per second in FACodec, to determine the number of speech tokens corresponding to each phoneme.

Evaluation dataset. The evaluation protocol follows VALL-E (Chen et al., 2025). Specifically, the LibriSpeech *test-clean* subset is filtered to retain utterances with durations between 4 and 10 seconds, resulting in a total of 2.2 hours of audio. For each utterance, a prompt is randomly sampled from another utterance spoken by the same speaker, from which a segment of 1, 3, or 5 seconds is extracted to serve as the prompt. To ensure consistency with prior work, we obtain the evaluation split used in our experiments from (Hieu et al., 2025) through communication with the authors.

C.2 Metrics Details

We assess each system utilizing the following objective evaluation metrics:

- **RTF** (Real-Time Factor) serves as a critical indicator of system efficiency, especially in applications that require real-time processing. It quantifies the duration needed to generate one second of speech. RTF evaluations for all models are conducted in a complete end-to-end configuration on a single NVIDIA 80GB A100 GPU.
- **UTMOS** (Saeki et al., 2022) is a deep learning framework designed to gauge the naturalness and general quality of speech by estimating mean opinion scores (MOS). This approach mitigates the resource-intensive nature of traditional subjective assessments, using sophisticated neural networks to produce predictions that strongly correlate with human perceptual evaluations.
- **SIM-O** is a metric used to quantify the similarity of the speakers. It evaluates the resemblance between the synthesized speech and the original prompt. This metric is derived from the cosine similarity of the speaker embeddings obtained through WavLM-TDCNN[1] applied to the audio waveforms. SIM-O spans a range of -1 to 1, where higher values indicate stronger speaker similarity.
- **WER** (Word Error Rate) is utilized to appraise the robustness of speech synthesis sys-

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

Algorithm 1 DiFlow-TTS Training

Input: Model $\mathcal{M}$, Dataset $\mathcal{D} = \{X^1, ..., X^M\}$, where each X^i consists of phonemes $\mathbf{P}$, durations $\mathbf{d}$, ground-truth speech $\mathbf{y}$, and reference speech prompt $\mathbf{r}$.

Output: Trained Model $\mathcal{M}$

while $\mathcal{M}$ not converged **do**
  Sample $X \sim \mathcal{D}$
  Extract prosody tokens $\mathbf{y}^p$, content tokens $\mathbf{y}^c$, and acoustic tokens $\mathbf{y}^a$ from $\mathbf{y}$, as defined in Eq. (2)
  Extract only prosody tokens $\mathbf{r}^p$, acoustic tokens $\mathbf{r}^a$, and speaker embedding $\mathbf{s}$ from $\mathbf{r}$, as defined Eq. (2)
  $\mathbf{x}_1 \leftarrow \mathbf{y}^p \oplus \mathbf{y}^a$ ▷ Prosody + acoustic tokens as target
  $\mathbf{p} \leftarrow \text{PhonemeEncoder}(\mathbf{P})$
  $\hat{\mathbf{d}} \leftarrow \text{DurationPredictor}(\mathbf{p})$
  $\mathbf{p}_{\text{up}} \leftarrow \text{LengthRegulator}(\mathbf{p}, \mathbf{d})$
  $\mathbf{h} \leftarrow \text{ContentPredictor}(\mathbf{p}_{\text{up}})$
  Obtain $\mathbf{h}_c$ and $p(\cdot|\mathbf{h}; \varphi)$ as defined in Eq. (3)
  $\mathcal{L}_{\text{dur}} \leftarrow \text{MSE}(\mathbf{d}, \hat{\mathbf{d}})$
  $\mathcal{L}_c \leftarrow \text{CE}(\mathbf{y}^c, p(\cdot|\mathbf{h}; \varphi))$
  Sample $t \sim \mathcal{U}(0, 1)$
  Sample $\mathbf{x}_t \sim p_{t|1}(\mathbf{x}_t \mid \mathbf{x}_1)$ ▷ Noising
  Obtain $\mathbf{h}_{p,a}$ using $\mathbf{x}_t$, $\mathbf{h}_c$, $\mathbf{r}^p$, $\mathbf{r}^a$, $\mathbf{s}$, t as defined in Section 3 of the paper
  $\mathbf{h}_p, \mathbf{h}_a = \text{Slice}(\mathbf{h}_{p,a})$
  $p_{1|t}(\cdot|\mathbf{x}_t, \mathbf{c}; \theta) \leftarrow f_\phi(\mathbf{h}_p) \oplus f_\omega(\mathbf{h}_a)$ ▷ Denoising prediction
  $\mathcal{L}_{\text{FDFD}} \leftarrow \text{CE}(\mathbf{x}_1, p_{1|t}(\cdot|\mathbf{x}_t, \mathbf{c}; \theta))$
  $\mathcal{L} \leftarrow \lambda_{\text{dur}}\mathcal{L}_{\text{dur}} + \lambda_c\mathcal{L}_c + \lambda_{\text{FDFD}}\mathcal{L}_{\text{FDFD}}$
  Optimizer.step($\mathcal{L}$)
end while

tems, focusing on the precision of word pronunciation. An automatic speech recognition (ASR) model[2] transcribes the generated speech, which is then compared to the textual prompt. The employed ASR model is a connectionist temporal classification (CTC)-based HuBERT, pre-trained on LibriLight, and fine-tuned on the 960-hour LibriSpeech training dataset. WER is reported as a value in the range $[0, 1]$, where lower values indicate better performance.

- **Prosody Accuracy & Error** metrics evaluate the congruence between the synthesized speech and the audio prompt, focusing on pitch (F0) and energy contours. Accuracy is determined following the framework outlined in PromptTTS (Guo et al., 2022) and TextrolSpeech (Ji et al., 2024b), by classifying F0 and energy into three tiers such as high, normal and low, which are relative to their mean values [3]. Furthermore, the Root Mean Square Error (RMSE) is calculated to measure deviations in F0 and the energy between the synthesized output and the reference prompts.

C.3 Baselines Details

We compare our model with previous zero-shot TTS baselines, including:

- **VoiceCraft** (Peng et al., 2024) is a token infilling neural codec language model built on a Transformer decoder architecture, incorporating a two-step token rearrangement procedure that applies causal masking for bidirectional-context autoregressive generation and delayed stacking for multi-codebook efficiency, trained autoregressively with a loss function that weights earlier codebooks more heavily. We use the official code and the pre-trained checkpoint[4], trained on 9K hours of the GigaSpeech dataset (Chen et al., 2021).

- **F5-TTS** (Chen et al., 2024b) is a fully non-autoregressive (NAR) TTS system based on flow matching with DiT architecture, where text inputs are padded with filler tokens to align with speech lengths, bypassing the need for duration models, text encoders, or phoneme alignment. It contributes refinements to text representation using ConvNeXt (Liu et al., 2022) for better speech alignment and an inference-time Sway Sampling strategy that improves generation efficiency. In our experiments, we use samples obtained through communication with the authors of (Hieu et al., 2025), reproduced using 500 hours of the LibriTTS dataset. We additionally perform inference using the official code [5] and a pre-trained checkpoint [6] trained on 100K hours of the Emilia dataset (He et al., 2024).

[2] https://huggingface.co/facebook/hubert-large-ls960-ft

[3] https://github.com/jishengpeng/TextrolSpeech

[4] https://huggingface.co/pyp1/VoiceCraft/blob/main/830M_TTSEnhanced.pth

[5] https://github.com/SWivid/F5-TTS

[6] https://huggingface.co/SWivid/F5-TTS

- **NaturalSpeech 2** (Shen et al., 2024) is a latent diffusion model designed for zero-shot TTS, capable of generating high-fidelity audio from diverse text inputs. It utilizes a neural audio codec and a latent diffusion framework to produce natural-sounding speech and singing without requiring speaker-specific training data. We use the Amphion toolkit (Zhang et al., 2024) and the pre-trained weight[7], trained on 585 hours of the LibriTTS dataset.

- **VALL-E** (Chen et al., 2025) is a neural codec language model that treats TTS synthesis as a conditional language modeling task using discrete codes derived from an off-the-shelf neural audio codec, pre-trained on 60,000 hours of English speech data to enable in-context learning. In our experiments, we use samples provided through communication with the authors of (Hieu et al., 2025). The model is reproduced using the Amphion toolkit[8] (Zhang et al., 2024) and trained on 500 hours of the LibriTTS dataset.

- **OZSpeech** (Hieu et al., 2025) is a zero-shot TTS system that employs optimal transport conditional flow matching with one-step sampling, conditioned on a learned prior derived from disentangled, factorized speech components represented in token format to model individual attributes. It contributes a novel framework that bypasses traditional multi-step sampling processes by leveraging the learned prior for direct generation from text prompts, thereby reducing computational demands and enhancing precise attribute disentanglement in speech synthesis. In our experiments, we use samples provided through communication with the authors, trained on 500 hours of the LibriTTS dataset.

- **MaskGCT** (Wang et al., 2025c) is a fully NAR zero-shot TTS model structured as a two-stage generative codec transformer, where the first stage predicts semantic tokens from input text using representations from a speech self-supervised learning model, and the second stage generates acoustic tokens conditioned on these semantic tokens via a mask-and-predict paradigm. It contributes an efficient training approach that learns to infill masked tokens based on prompts and conditions, enabling parallel inference for tokens of arbitrary length without explicit text-speech alignment or phone-level duration modeling, thus resolving key limitations in prior autoregressive and NAR TTS frameworks. We use the official code and the pretrained checkpoint [9], trained on English and Chinese data from Emilia (He et al., 2024), each with 50K hours of speech (totaling 100K hours). Since the baseline requires the total speech length, we use the ground-truth duration during inference.

[7] https://huggingface.co/amphion/naturalspeech2_libritts/tree/main/checkpoint

[8] https://github.com/open-mmlab/Amphion

NFE	RTF ↓	UTMOS ↑
1	0.022	2.904
2	0.025	2.908
4	0.031	3.313
8	0.043	3.698
16	0.066	3.864
32	0.112	3.923
64	0.207	3.958
128	0.394	3.978

Table 5: Performance of DiFlow-TTS vs. NFE count with 3-second audio prompts.

D Additional Analysis

Effect of NFE. We investigate the impact of varying NFE from 1 to 128 on DiFlow-TTS performance to explore the trade-off between inference efficiency and synthesis quality, as presented in Table 5. Increasing NFE markedly improves UTMOS, indicating that the FDFD module benefits from additional refinement steps to generate more natural speech. In particular, performance stabilizes around 32 NFE, with optimal audio quality observed at 64 NFE, and only marginal improvements beyond this point. Although RTF naturally increases with NFE, the overall latency remains competitive (see Table 3). These results demonstrate its effective trade-off between quality and efficiency.

Attribute-type embeddings. We conduct a finer-grained ablation to isolate the contribution of each attribute-specific embedding. The additional results are reported in the Table 6. We observe that removing either the prosody attribute embedding or the acoustic attribute embedding leads to noticeable performance degradation, particularly in natural-

[9] https://huggingface.co/amphion/MaskGCT

Model	UTMOS ↑	WER ↓	SIM-O ↑	F0		Energy	
				Accuracy ↑	RMSE ↓	Accuracy ↑	RMSE ↓
DiFlow-TTS	**3.978**	**0.048**	**0.454**	**0.884**	**7.972**	0.735	**0.007**
- w/o Prosody Attribute Embedding	3.918	0.062	0.445	0.846	9.547	0.734	0.007
- w/o Acoustic Attribute Embedding	3.929	0.060	0.447	0.868	9.216	0.741	0.007

Table 6: Ablation study on attribute-type embeddings.

ness and metrics closely tied to these attributes. Attribute-type embeddings guide the model in distinguishing which attribute each token corresponds to when feeding them into the DiT. When an attribute embedding is removed, the model is forced to treat all attributes uniformly, which results in entanglement and reduced synthesis quality.

Prompt Duration. To investigate in detail the influence of prompt duration on zero-shot speech synthesis, we conducted a comprehensive evaluation of DiFlow-TTS with 128 NFE across different prompt lengths: 1 second, 3 seconds (as reported in the paper), and 5 seconds. As shown in Table 7, increasing the prompt duration consistently improves all aspects of speech quality across models. Specifically, DiFlow-TTS, along with OZSpeech, achieves the lowest WER across all prompt lengths, demonstrating superior content preservation. In terms of naturalness and overall quality, our method attains SOTA performance, achieving the highest UTMOS score (4.00) with a 5-second prompt. Notably, this is achieved using only 470 hours of training data, whereas VoiceCraft (9K hours) and MaskGCT (100K hours) obtain lower UTMOS scores of 3.58 and 3.89, respectively. For speaker similarity, our method does not show a clear advantage over baseline models, though we note that these baselines also exhibit trade-offs under limited training data. Regarding pitch and energy accuracies and errors, which reflect prosody reconstruction ability, DiFlow-TTS consistently ranks as the best or second-best performer across prompt lengths. Overall, DiFlow-TTS strikes a favorable balance among naturalness, prosody, model size, and speaker similarity.

Algorithm 2 DiFlow-TTS Inference

Input: The phonemes $\mathbf{P}$, and reference speech prompt $\mathbf{r}$, the number of sampling step N, and step size $\Delta_t = \frac{1}{N}$.

Output: Synthesized speech $\hat{a}$.

Extract prosody tokens $\mathbf{r}^p$, acoustic tokens $\mathbf{r}^a$, and speaker embedding $\mathbf{s}$ from $\mathbf{r}$, as defined in Eq. (2)
$\mathbf{p} \leftarrow \text{PhonemeEncoder}(\mathbf{P})$
$\hat{\mathbf{d}} \leftarrow \text{DurationPredictor}(\mathbf{p})$
$\mathbf{p}_{\text{up}} \leftarrow \text{LengthRegulator}(\mathbf{p}, \hat{\mathbf{d}})$
$\mathbf{h} \leftarrow \text{ContentPredictor}(\mathbf{p}_{\text{up}})$
Obtain $\mathbf{h}_c$ and $p(\cdot|\mathbf{h}; \varphi)$ using $\mathbf{h}$ as defined in Eq. (3)
Sample $\mathbf{x}_0 \sim p(\mathbf{x}_0)$
for $t = 0$ **to** $1 - \Delta_t$ **with step** Δ_t **do**
    Obtain $\mathbf{h}_{p,a}$ using $\mathbf{x}_t$, $\mathbf{h}_c$, $\mathbf{r}^p$, $\mathbf{r}^a$, $\mathbf{s}$, t as defined in 3 of the paper
    $\mathbf{h}_p, \mathbf{h}_a = \text{Slice}(\mathbf{h}_{p,a})$
    $p_{1|t}(\cdot|\mathbf{x}_t, \mathbf{c}; \theta) \leftarrow f_\phi(\mathbf{h}_p) \oplus f_\omega(\mathbf{h}_a)$ ▷ Denoising prediction
    Sample $\mathbf{x}_1^i \sim p_{1|t}^i(\cdot|\mathbf{x}_t, \mathbf{c}; \theta)$
    $\mathbf{u}_t^i(x^i|\mathbf{x}_t^i, \mathbf{x}_1^i) \leftarrow \frac{\dot{\kappa}_t}{1-\kappa_t}\left[\delta_{\mathbf{x}_1^i}(x^i) - \delta_{\mathbf{x}_t^i}(x^i)\right]$ ▷ Probability velocity as defined in 1
    $\lambda^i \leftarrow \sum_{x^i \neq \mathbf{x}_t^i} \mathbf{u}_t^i(x^i|\mathbf{x}_t^i, \mathbf{x}_1^i)$
    Sample $z^i \sim \mathcal{U}(0, 1)$
    if $z^i \leq 1 - \exp(-\Delta_t \lambda^i)$ **then**
        Sample $\mathbf{x}_{t+\Delta_t}^i \sim \frac{1}{\lambda^i}\mathbf{u}_t(\cdot|\mathbf{x}_t^i, \mathbf{x}_1^i)(1 - \delta_{\mathbf{x}_t^i}(\cdot))$ ▷ Transition to a new token; self-transitions are disallowed
    else
        Sample $\mathbf{x}_{t+\Delta_t}^i \sim \delta_{\mathbf{x}_t}(\cdot)$ ▷ No transition; retain current token
    end if
end for
$\mathbf{x}^p, \mathbf{x}^a = \text{Split}(\mathbf{x}_t)$
$\mathbf{x}^c \leftarrow \arg\max_x \text{softmax}(p(x \mid \mathbf{h}; \varphi))$
$\mathbf{x} = \mathbf{x}^p \oplus \mathbf{x}^c \oplus \mathbf{x}^a$
$\hat{a} \leftarrow \text{CodecDecoder}(\mathbf{x}, \mathbf{s})$

Type	Model	Data (hours)	UTMOS ↑	WER ↓	SIM-O ↑	F0		Energy	
						Accuracy (↑)	RMSE ↓	Accuracy ↑	RMSE ↓
-	Ground Truth	-	4.10	0.02	-	-	-	-	-
				1s Prompt					
(i)	VoiceCraft [†]	GS (9K)	3.45	0.16	0.31	0.61	31.57	0.52	**0.010**
	VALL-E [⋄]	LT (500)	<u>3.61</u>	0.21	0.24	0.55	37.87	0.40	0.020
(ii)	NaturalSpeech 2 [‡]	LT (585)	2.12	0.12	0.20	<u>0.69</u>	26.48	0.39	0.020
	F5-TTS	-	-	-	-	-	-	-	-
	OZSpeech [†]	LT (500)	3.17	**0.05**	0.30	0.62	<u>27.70</u>	0.49	0.020
(iii)	MaskGCT [†]	E (100K)	3.60	<u>0.10</u>	**0.36**	0.63	29.63	**0.60**	<u>0.013</u>
(iv)	**DiFlow-TTS**	**LT (470)**	**3.74**	**0.05**	<u>0.34</u>	**0.82**	**13.00**	<u>0.55</u>	**0.010**
				3s Prompt					
(i)	VoiceCraft [†]	GS (9K)	3.55	0.18	0.51	0.78	17.22	0.44	<u>0.010</u>
	VALL-E [⋄]	LT (500)	3.68	0.19	0.40	0.75	21.66	0.36	0.020
(ii)	NaturalSpeech 2 [‡]	LT (585)	2.38	<u>0.09</u>	0.31	0.80	15.62	0.25	0.020
	F5-TTS [⋄]	LT (500)	3.76	0.24	0.52	0.80	13.78	0.67	<u>0.010</u>
	F5-TTS [†]	E (100K)	3.72	<u>0.09</u>	<u>0.66</u>	<u>0.83</u>	12.66	0.66	<u>0.010</u>
	OZSpeech [†]	LT (500)	3.15	**0.05**	0.40	0.81	11.96	0.67	<u>0.010</u>
(iii)	MaskGCT [†]	E (100K)	<u>3.83</u>	0.09	**0.67**	0.77	14.33	**0.75**	**0.007**
(iv)	**DiFlow-TTS**	**LT (470)**	**3.98**	**0.05**	0.45	**0.88**	**7.97**	<u>0.73</u>	**0.007**
				5s Prompt					
(i)	VoiceCraft [†]	GS (9K)	3.58	0.19	0.56	0.81	14.48	0.46	0.010
	VALL-E [⋄]	LT (500)	3.72	0.19	0.46	0.79	18.20	0.41	0.010
(ii)	NaturalSpeech 2 [‡]	LT (585)	2.33	0.09	0.35	0.84	13.13	0.28	0.020
	F5-TTS [⋄]	LT (500)	3.71	0.32	0.57	0.83	11.20	0.68	0.010
	F5-TTS [†]	E (100K)	3.78	<u>0.07</u>	<u>0.72</u>	<u>0.86</u>	<u>10.54</u>	0.68	0.009
	OZSpeech [†]	LT (500)	3.15	**0.05**	0.39	0.83	12.05	0.67	<u>0.010</u>
(iii)	MaskGCT [†]	E (100K)	<u>3.89</u>	0.09	**0.74**	0.81	11.82	**0.77**	<u>0.005</u>
(iv)	**DiFlow-TTS**	**LT (470)**	**4.00**	**0.05**	0.48	**0.89**	**8.04**	<u>0.73</u>	**0.007**

Table 7: Performance on the *LibriSpeech test-clean* dataset across different audio prompt lengths. [⋄] means reproduced results. [†] and [‡] mean results inferred from official and unofficial checkpoints, respectively. The best and second best are **bold** and <u>underlined</u>. Abbreviation: E (Emilia), GS (GigaSpeech), LT (LibriTTS). Note that F5-TTS does not support the 1-second prompt setting in the official code, which requires prompts longer than 3 seconds.