GOGO: GROUP-WISE GRANULARITY-ORDERED CODEC FOR STABLE AND EFFICIENT SPEECH GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Current speech language models require their core component, the speech codec, to discretize continuous speech signals into tokens that not only capture high-level cues for autoregressive modeling but also preserve sufficient acoustic details for perceptual quality. To address this need, we propose Gogo, a group-wise granularity-ordered codec that quantizes each group of frames into tokens arranged from coarse to fine, where coarse tokens encode high-level abstractions and fine tokens progressively recover low-level details. Building on the granularity-ordering property of Gogo, we introduce GogoSpeech, a two-stage speech language model that performs speech generation by first constructing a coarse speech backbone at an extremely low token rate and then enriching the backbone with fine-grained acoustic details. Considering the inherently non-uniform information distribution in speech signals, we further design a Group Relative Policy Optimization (GRPO)-trained token allocator that adaptively allocates token budgets to groups based on group-wise complexity. Experimental results demonstrate that Gogo delivers state-of-the-art reconstruction performance across most metrics at a token rate of 47. Moreover, evaluations on zero-shot text-to-speech tasks show that GogoSpeech enables efficient generation by adaptively reducing the average token rate, and attains state-of-the-art results in long-form speech generation.

1 INTRODUCTION

Large language models (LLMs) such as the GPT series (Brown et al., 2020; OpenAI, 2024) have demonstrated remarkable capabilities across diverse text-based tasks. Their success has inspired growing efforts to extend the LLM paradigm to the speech modality, leading to the development of speech language models (SLMs) capable of understanding and generating spoken language (Ye et al., 2025b; Fang et al., 2025; Défossez et al., 2024). A common pipeline for SLMs first discretizes continuous speech into sequences of tokens via an audio codec, then models both text and speech tokens in an autoregressive framework. The effectiveness of this approach hinges critically on the codec's ability to produce tokens that concurrently contain high-level cues (*e.g.*, content, semantics, and structural attributes) for autoregressive modeling (Ye et al., 2025a) and sufficient acoustic details (*e.g.*, low-level acoustic fluctuations) for perceptual quality preservation.

Conventional audio codecs, originally designed for compression and transmission, adopt a frame-wise quantization scheme (Kleijn et al., 2021; Valin et al., 2012; Défossez et al., 2022). While this enables high-fidelity reconstruction, its strong locality bias limits the codec's ability to capture high-level cues needed by SLMs. To address this limitation, recent works have augmented codecs with self-supervised representations (Défossez et al., 2024; Zhang et al., 2024; Li et al., 2025) or automatic speech recognition (ASR) features (Du et al., 2024b; Jo et al., 2025; Zeng et al., 2025) to explicitly inject high-level linguistic and semantic information into the quantization process. However, the fundamental frame-wise quantization paradigm remains unbroken, inherently limiting the ability to learn high-level information. Besides, little attention has been given to the non-uniform information distribution in speech (Dieleman et al., 2021; Voran, 2024). Current approaches generally allocate one same bitrate to all segments, which leads to redundant coding and low generation efficiency, especially in less complex segments like silence where a low coding rate would suffice.

To address the aforementioned limitations, we redesign both the codec and SLM framework. As depicted in Figure 1, we first propose Gogo, a group-wise granularity-ordered codec that processes

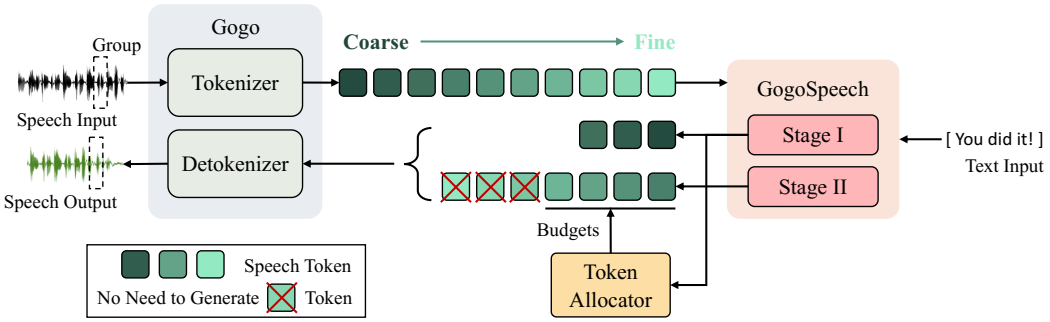

Figure 1: System overview. Only one group is plotted for simplicity. The shading of a token reflects the granularity of the information it encodes. Best viewed in color.

contiguous frames as groups and generates tokens in a coarse-to-fine order, where coarse tokens capture high-level information and fine tokens are used to progressively restore low-level acoustic details. Building on the granularity-ordering property of Gogo, we develop GogoSpeech, a two-stage SLM for speech generation. In the first stage, a high-level speech backbone, which serves as a coarse indicator of the target speech, is predicted at a extremely low feature rate (around 14 Hz). This reduced feature rate enhances the stability of autoregressive prediction and mitigates error accumulation (Arora et al., 2022; He et al., 2021). In the second stage, fine-grained details are incrementally recovered conditioned on the speech backbone. The feature rate in the second stage is restored to a standard level to ensure high-fidelity synthesis with precise details. To further enhance efficiency, we propose a Group Relative Policy Optimization (GRPO)-trained (Shao et al., 2024) token allocator that dynamically assigns token budgets to every group based on its complexity, thereby aligning computational resources with the non-uniform information density of speech signals.

Our main contributions in this work are fivefold: (1) proposing a new speech codec, namely Gogo, featuring group-by-group and granularity-ordered tokenization that better addresses the representational requirements of SLMs; (2) constructing a new speech language model, namely GogoSpeech, enabling staged speech generation from high-level abstractions to fine acoustic details; (3) developing a GRPO-trained token allocator that optimizes efficiency and quality by aligning token budgets with group-wise complexity; (4) conducting reconstruction experiments to show that Gogo achieves superior performance compared to state-of-the-art codecs; (5) performing experiments on text-to-speech (TTS) to show that GogoSpeech achieves better performance with higher stability and efficiency. Demo samples can be found at `https://anonymous.4open.science/w/gogo`.

2 METHODS

2.1 GOGO

The proposed Gogo, as shown in Figure 2, comprises three main components: an encoder for learning speech representations, a flow-based generative model (Lipman et al., 2023; Tong et al., 2024) for mel-spectrogram reconstruction, and a vocoder for converting spectrograms into waveforms. Additionally, Gogo integrates an ASR module and an autoregressive (AR) prior (Wang et al., 2025; Yang et al., 2025) to enhance the suitability of the learned speech tokens for downstream generation.

2.1.1 WORKFLOW

Given an input waveform w, we first extract its mel-spectrogram $x \in \mathbb{R}^{n_f \times d}$, where n_f denotes the number of frames and d the number of mel bins. The spectrogram is then partitioned along the temporal axis into multiple non-overlap groups $x^i \in \mathbb{R}^{g \times d}$, where g denotes the group size, $i \in [1, n_g]$ denotes the group index, and $n_g = \lceil \frac{n_f}{g} \rceil$ denotes the total number of groups. The last group is zero-padded if necessary. Subsequently, each group is concatenated with n_q learnable queries $q^i \in \mathbb{R}^{n_q \times d}$, which yields $z^i \in \mathbb{R}^{(g+n_q) \times d}$. The extended sequences z^i are encoded by Transformer (Vaswani et al., 2017) encoder, after which the x^i part is discarded and finite scalar quantiza-

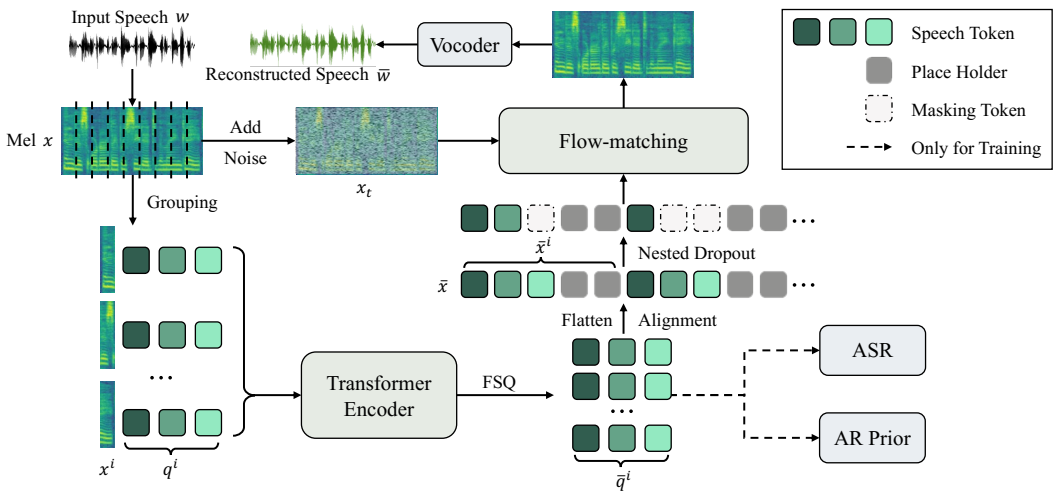

Figure 2: Architecture of Gogo. Here the group size g is set to 5 and the number of speech queries n_q assigned to each group is set to 3 for demonstration. Best viewed in color.

tion (FSQ) (Mentzer et al., 2024) is applied to the positions corresponding to the learnable queries, producing speech token indices $s^i \in \mathbb{R}^{n_q}$ and their corresponding embeddings $\bar{q}^i \in \mathbb{R}^{n_q \times d_h}$, where d_h denotes the hidden dimension. Formally, the tokenization workflow in Gogo is given by:

$$\begin{aligned} x &= \text{Mel}(w), \\ x^1, x^2, \cdots, x^{n_g} &= \text{Grouping}(x), \\ z^i &= \text{Cat}(x^i, q^i), \\ \bar{q}^i, s^i &= \text{FSQ}(\text{Encoder}(z^i)), \end{aligned} \tag{1}$$

where $\text{Mel}(\cdot)$ denotes the mel extraction operation, $\text{Grouping}(\cdot)$ denotes the grouping operation, and $\text{Cat}(\cdot)$ denotes the concatenation operation.

For reconstruction, $\bar{q}^i$ are first padded with $(g - n_q)$ placeholder tokens to match the original group length g, resulting in aligned features $\bar{x}^i \in \mathbb{R}^{g \times d_h}$. All groups' $\bar{x}^i$ are concatenated along the time axis to generate $\bar{x} \in \mathbb{R}^{n_f \times d_h}$, which is then fed into a flow-matching model to predict the mel-spectrogram. The final waveform $\bar{w}$ is recovered using a pretrained Vocos (Siuzdak, 2023) vocoder. Formally, the reconstruction workflow in Gogo is given by:

$$\begin{aligned} \bar{x}^i &= \text{Cat}(\bar{q}^i, \text{Placeholders}), \\ \bar{x} &= \text{Cat}(\bar{x}^1, \bar{x}^2, \cdots, \bar{x}^{n_g}), \\ \bar{w} &= \text{Vocoder}(\text{Flow}(\bar{x})). \end{aligned} \tag{2}$$

2.1.2 Training Objectives

Gogo leverages conditional flow matching (CFM) (Lipman et al., 2023), which extends the framework of continuous normalizing flows (Chen et al., 2018) to learn a time-dependent vector field that transports a simple prior distribution, *e.g.*, the standard normal distribution, to the distribution of the mel-spectrograms conditioned on the features $\bar{x}$. Given a mel-spectrogram x_1 and a Gaussian noise $x_0 \sim \mathcal{N}(0, I)$, we first interpolate between them and produce a noisy spectrogram $x_t = (1-t)x_0 + tx_1$ using a flow time $t \sim \mathcal{U}(0, 1)$. The conditional vector field for this linear interpolation is $v(x_0, x_1, t) = \partial x_t / \partial t = x_1 - x_0$. The flow-matching model, parameterized by θ, takes $(x_t, \bar{x}, t)$ as input and predicts a velocity vector $v_\theta(x_t, \bar{x}, t)$. Formally, the CFM objective is defined as a simple vector field regression loss:

$$\mathcal{L}_{\text{CFM}} = \mathbb{E}_{t, p(x_0), q(x_1)} \left[\| v_\theta(x_t, \bar{x}, t) - v(x_0, x_1, t) \|_2^2 \right]. \tag{3}$$

We further introduce two auxiliary modules, AR prior and ASR module, to encourage the speech queries to capture temporal dependencies and linguistic information within each group, respectively.

The corresponding training objectives, $\mathcal{L}_{\text{AR}}$ and $\mathcal{L}_{\text{ASR}}$, are formally defined in Appendix B. Finally, the objective for training Gogo can be written as:

$$\mathcal{L}_{\text{Gogo}} = \lambda_{\text{CFM}}\mathcal{L}_{\text{CFM}} + \lambda_{\text{AR}}\mathcal{L}_{\text{AR}} + \lambda_{\text{ASR}}\mathcal{L}_{\text{ASR}}, \tag{4}$$

where λ_{CFM}, λ_{AR}, and λ_{ASR} are coefficients employed to balance different loss components.

2.1.3 Granularity Ordering

To enforce a coarse-to-fine ordering in the learned speech queries q^i and tokens s^i, we introduce three techniques: asymmetric masking, nested dropout (Rippel et al., 2014), and loss balancer.

Asymmetric masking is introduced to control the information flow within the encoder. Specifically, the mel features x^i are allowed to attend to each other but not to the speech queries q^i, while each speech query can attend to all mel features. Moreover, the j-th speech query is permitted to attend to the k-th speech query only if $j \geq k$. This design ensures that each speech query is generated by progressively incorporating finer-grained information conditioned on all preceding query.

Nested dropout randomly drops tokens in a nested fashion during training. Specifically, we uniformly sample the number of tokens to retain, $n_k \in \{1, \ldots, n_q\}$, and replace the last $(n_q - n_k)$ tokens with masking tokens m. This mechanism drives Gogo to prioritize encoding high-level abstractions and essential structural attributes into the earlier coarse tokens to minimize $\mathcal{L}_{\text{Gogo}}$ to the greatest extent, while deferring the challenging and fluctuating details to the later fine tokens. Since later tokens are rarely preserved and receive fewer gradient updates, we further introduce a re-weighting mechanism for compensation. Concretely, the gradient of the j-th speech token is scaled by $w_j = 0.5/(1-(j-1)/n_q)$, assigning larger weights to tokens with fewer updates and vice versa. Details of the re-weighting implementation are provided in Appendix A.

Loss balancer is utilized to adjust the loss coefficients λ_{CFM} and λ_{ASR}, further ensuring that the learned speech tokens are organized in a coarse-to-fine manner. Specifically, when n_k is small, the model should emphasize $\mathcal{L}_{\text{ASR}}$ so that coarse tokens encode richer linguistic content. Conversely, when n_k is large, $\mathcal{L}_{\text{CFM}}$ should dominate to ensure that fine tokens capture more acoustic details. Let λ_{max} and λ_{min} denote the maximum and minimum weighting coefficients, respectively. The loss balancer adaptively adjusts λ_{CFM} and λ_{ASR} as follows:

$$\lambda_{\text{CFM}} = \lambda_{\text{min}} + \frac{(n_k - 1)(\lambda_{\text{max}} - \lambda_{\text{min}})}{n_q - 1}, \quad \lambda_{\text{ASR}} = \lambda_{\text{max}} - \frac{(n_k - 1)(\lambda_{\text{max}} - \lambda_{\text{min}})}{n_q - 1}. \tag{5}$$

2.2 GogoSpeech

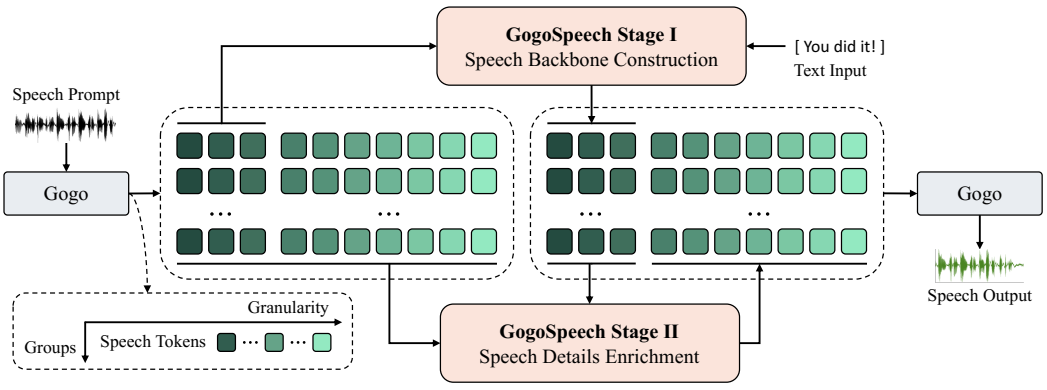

Figure 3: Architecture of GogoSpeech. Gogo encodes the speech prompt into speech tokens, which serve as input to GogoSpeech for generating the target speech tokens. The target speech tokens are transformed into waveform by Gogo. For visualization, the number of speech queries n_q in each group is set to 10 and the speech backbone is defined as the first 3 speech tokens of each group.

2.2.1 STAGE I: SPEECH BACKBONE CONSTRUCTION

Stacking all s^i yields the speech token matrix $\mathbf{S} \in \mathbb{R}^{n_g \times n_q}$. The speech backbone is defined as the first b tokens of each group, *i.e.*, $\mathbf{S}_{:,1:b} \in \mathbb{R}^{n_g \times b}$, which contains the high-level cues of the speech signal. As shown in Figure 3, given the input text $y = (y_1, \ldots, y_L)$ and the input backbone $\mathbf{S}_{:,1:b}$, the autoregressive model in Stage I generates the backbone of target speech $\tilde{\mathbf{S}}_{:,1:b} \in \mathbb{R}^{\tilde{n}_g \times b}$, where $\tilde{n}_g$ denotes the number of groups in the target speech. Let $\Gamma(\cdot)$ denote the flatten operation. The objective of Stage I is to minimize the negative log-likelihood over the target speech backbone:

$$\mathcal{L}_{\text{stage1}} = -\sum_{i=1}^{\tilde{n}_g} \sum_{t=1}^{b} \log P(\tilde{\mathbf{S}}_{i,t} \mid y, \Gamma(\mathbf{S}_{:,1:b}), \Gamma(\tilde{\mathbf{S}}_{1:i-1,1:b}), \tilde{\mathbf{S}}_{i,1:t-1}) \tag{6}$$

2.2.2 STAGE II: SPEECH DETAILS ENRICHMENT

In Stage II, GogoSpeech enriches the speech backbone predicted by Stage I via adding fine-grained acoustic details. For the i-th group of the target speech, the autoregressive model in Stage II generates the fine tokens $\tilde{\mathbf{S}}_{i,b+1:n_q}$ conditioned on all tokens of the input speech $\mathbf{S}$, all tokens of the previously generated groups $\tilde{\mathbf{S}}_{1:i-1,:}$, and the speech backbone of the current group $\tilde{\mathbf{S}}_{i,1:b}$. The training objective for Stage II is given by:

$$\mathcal{L}_{\text{stage2}} = -\sum_{i=1}^{\tilde{n}_g} \sum_{t=b+1}^{n_q} \log P(\tilde{\mathbf{S}}_{i,t} \mid \Gamma(\mathbf{S}), \Gamma(\tilde{\mathbf{S}}_{1:i-1,:}), \tilde{\mathbf{S}}_{i,1:t-1}) \tag{7}$$

2.3 TOKEN ALLOCATOR

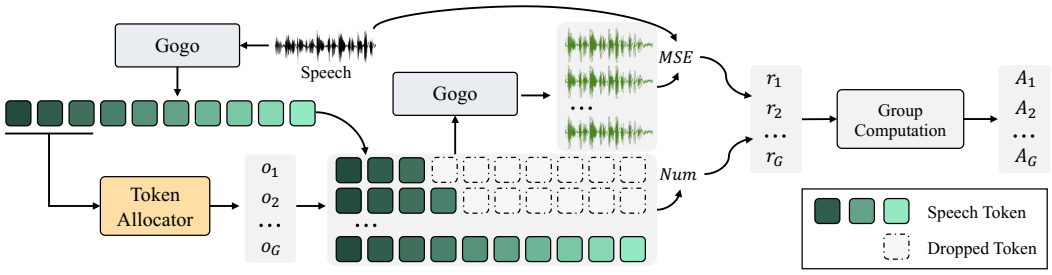

Figure 4: GRPO-trained token allocator. Throughout the GRPO training, the Gogo is kept frozen.

To improve the efficiency of speech generation, we want to allocate more tokens to acoustically complex groups while assigning fewer tokens to simpler ones, such as silence. To enable such adaptive allocation, we design a token allocator that receives the backbone of each group, $\tilde{\mathbf{S}}_{i,1:b}$, as input and outputs a budget $\xi_i \in \{0, 1, \ldots, n_q - b\}$, indicating the number of fine tokens to be generated for that group in GogoSpeech Stage II. The skipped unnecessary fine tokens are replaced with masking tokens m. The cooperation between the token allocator π_ω and GogoSpeech is formulated as:

$$\xi_i = \pi_\omega(\tilde{\mathbf{S}}_{i,1:b}) \in \{0, 1, \ldots, n_q - b\}, \tag{8}$$

$$\tilde{\mathbf{S}}_{i,b+1:b+\xi_i} = \arg\max_{u} \prod_{j=b+1}^{b+\xi_i} p\left(u_j \mid y, \Gamma(\mathbf{S}), \Gamma(\tilde{\mathbf{S}}_{1:i-1,:}), \tilde{\mathbf{S}}_{i,1:j-1}\right), \quad \tilde{\mathbf{S}}_{i,b+\xi_i+1:n_q} = m. \tag{9}$$

As shown in Figure 4, the token allocator is trained from scratch using a slightly modified GRPO technique (Shao et al., 2024). Given that the output space of the allocator is relatively small, comprising $(n_q - b + 1)$ discrete allocation choices, we enumerate all possible outputs $(o_1, o_2, \cdots, o_G)$, corresponding to using from b to n_q tokens for reconstructing the input speech. The resulting $(n_q - b + 1)$ reconstructed samples are employed to compute the group scores.

We adopt two reward metrics, including $\mathcal{R}_n$ which penalizes the number of tokens utilized for reconstruction, and $\mathcal{R}_d$ which penalizes the distance between the input and reconstructed speech. This

joint reward encourage the allocator to learn allocation strategies that achieve high reconstruction fidelity while minimizing the number of tokens consumed. The rewards $\mathcal{R}_n$ and $\mathcal{R}_d$, and the entire reward $\mathcal{R}$ are defined as follows:

$$\mathcal{R}_n = -\text{Num}(\bar{x}), \;\; \mathcal{R}_d = -\mathbb{E}\left[\|\text{Mel}(w) - \text{Mel}(\bar{w})\|_2^2\right], \tag{10}$$

$$\mathcal{R} = \lambda_n \mathcal{R}_n + \lambda_d \mathcal{R}_d, \tag{11}$$

where $\text{Num}(\bar{x})$ denotes the number of speech tokens used to generate $\bar{w}$, λ_n and λ_d are coefficients used to balance the two reward terms. After obtaining the reward for each allocation choice, the advantage is calculated using group relative advantage estimation (Shao et al., 2024):

$$\mathcal{A}_j = \frac{\mathcal{R}_j - \text{mean}(\mathcal{R})}{\text{std}(\mathcal{R})}. \tag{12}$$

Since the token allocator is initialized from scratch, we omit the KL penalty term used in the original GRPO framework. The allocator π_ω is then optimized by maximizing the following objective:

$$\mathcal{J}_{\text{GRPO}} = \mathbb{E}_{o_j \sim \pi_\omega(o|\tilde{\mathbf{S}}_{i,1:b})} \left[\frac{1}{G} \sum_{j=1}^{G} \pi_\omega(o_j|\tilde{\mathbf{S}}_{i,1:b}) \mathcal{A}_j \right]. \tag{13}$$

3 Experimental Setup

3.1 Implementation Details

We extract 100-dimensional log mel-filterbank features from audio resampled to 24 kHz, using a hop length of 256 and a window size of 1024, resulting in a feature rate of approximately 94 Hz. We set the group size to $g = 20$, and allocate $n_q = 10$ speech queries per group. Therefore, the token rate of Gogo is computed as $n_q \times (94/g) = 47$ Hz. The speech backbone is defined as the first $b = 3$ tokens of each group. Thus the token rate of backbone is about 14 Hz. Both Stage I and Stage II of GogoSpeech are initialized from the LLaMA (Grattafiori et al., 2024), with the vocabulary expanded to include speech tokens. A pretrained Vocos (Siuzdak, 2023) is employed to convert the generated spectrograms into waveforms. Please refer to Appendix C for more detailed model configuration.

During training, we empirically set the weight of loss $\mathcal{L}_{\text{AR}}$ to $\lambda_{\text{AR}} = 0.06$ and amplify the gradient of the AR prior by a factor of 50 to ensure effective updates. The coefficients for the losses $\mathcal{L}_{\text{ASR}}$ and $\mathcal{L}_{\text{CFM}}$ are dynamically adjusted using the loss balancer defined in Eq. 5, with $\lambda_{\min} = 0.2$ and $\lambda_{\max} = 1.8$. For training the token allocator via GRPO, we set the reward coefficients $\lambda_n = 0.2$ and $\lambda_d = 1.0$ to balance the trade-off between token efficiency and reconstruction quality. Additional hyperparameters, training schedules, and inference configurations are provided in Appendix D.

3.2 Datasets

We train both Gogo, GogoSpeech, and the token allocator on the Emilia dataset (He et al., 2024), a large-scale and diverse in-the-wild speech corpus designed for multilingual speech generation. In this work, we use its English subset, which contains approximately 50K hours of transcribed speech covering a diverse set of speakers, acoustic characteristics, and background conditions. For evaluating the reconstruction quality of Gogo, we adopt the LibriTTS test-clean set (Zen et al., 2019) with 4,837 samples in total. To assess the zero-shot speech generation capability of GogoSpeech, we use the Seed-TTS test-en set (Anastassiou et al., 2024), which consists of 1,000 samples drawn from Common Voice dataset (Ardila et al., 2019). All speech samples are resampled to 24 kHz.

3.3 Baselines and Evaluation Metrics

We compare Gogo against multiple codec baselines, including EnCodec (Défossez et al., 2022), DAC (Kumar et al., 2023), SpeechTokenizer (Zhang et al., 2024), Mimi (Défossez et al., 2024), SNAC (Siuzdak et al., 2024), WavTokenizer (Ji et al., 2025), MagiCodec (Song et al., 2025), X-codec2 (Ye et al., 2025b), TAAE (Parker et al., 2025), and DualCodec (Li et al., 2025). All baseline results are obtained using their official checkpoints. Details of compared codecs see Appendix E.

Table 1: Comparison between different codec models on the LibriTTS test-clean set. Bold values indicate the best for each token rate. TPS and FPS denote the number of tokens and frames per second, respectively. #CB denotes the number of codebook employed in each model.

Model	TPS	FPS	#CB	UT MOS	DNS MOS	STOI	PESQ WB	PESQ NB	SIM	WER
Ground Truth	-	-	-	4.13	3.83	1.00	4.64	4.55	1.00	5.86
DAC	600	75	8	**3.78**	**3.75**	**0.99**	**3.52**	**3.85**	**0.98**	**6.10**
EnCodec	600	75	8	3.13	3.56	0.94	2.74	3.36	0.97	6.24
DAC	150	75	2	1.94	3.27	0.85	1.53	1.95	0.90	10.81
EnCodec	150	75	2	1.57	3.20	0.85	1.54	1.92	0.91	8.98
SpeechTokenizer	150	50	3	3.10	3.56	0.85	1.47	1.86	0.90	7.23
Mimi	150	12.5	12	**3.88**	**3.77**	**0.94**	**2.67**	**3.22**	**0.95**	**6.54**
SNAC	82	47	3	3.80	**3.84**	0.91	2.23	2.75	**0.91**	**7.47**
DAC	75	75	1	1.33	2.97	0.76	1.18	1.45	0.81	30.06
EnCodec	75	75	1	1.24	2.69	0.78	1.21	1.45	0.77	33.15
WavTokenizer	75	75	1	**4.11**	3.65	**0.92**	**2.43**	**2.96**	0.90	8.34
Mimi	75	12.5	6	3.54	3.69	0.90	2.01	2.53	0.90	7.65
SpeechTokenizer	50	50	1	1.31	3.09	0.68	1.11	1.28	0.67	9.18
MagiCodec	50	50	1	4.21	3.96	**0.93**	2.55	3.18	0.86	7.45
X-codec2	50	50	1	4.17	3.90	0.92	2.45	3.07	0.83	6.40
TAAE	50	25	2	**4.27**	3.89	0.91	2.14	2.82	0.87	8.18
DualCodec	50	25	2	4.05	3.80	0.89	2.02	2.58	0.89	6.54
Mimi	50	12.5	4	3.16	3.62	0.86	1.64	2.10	0.87	9.24
Gogo	47	47	1	4.19	**3.99**	0.92	**2.59**	**3.26**	**0.91**	**6.35**

We compare GogoSpeech with TTS baselines, including FireRedTTS-1S (Guo et al., 2025a), F5-TTS Chen et al. (2025), XTTS-v2 (Casanova et al., 2024), Llasa (Ye et al., 2025b), CosyVoice 2 (Du et al., 2024b), and VoiceCraft (Peng et al., 2024). More details can be found in Appendix F.

For evaluation, we employ both objective and subjective metrics. For objective assessment, we adopt UT-MOS (Saeki et al., 2022), DNS-MOS (Reddy et al., 2022), and Perceptual Evaluation of Speech Quality (PESQ) (Rix et al., 2001) to quantify perceptual quality and speech distortion. Speech intelligibility is measured using Short-Time Objective Intelligibility (STOI) (Taal et al., 2010) and Word Error Rate (WER). In addition, speaker similarity (SIM) is calculated to evaluate the accuracy of speaker identity preservation. For subjective evaluation, we employ the Similarity Mean Opinion Score (SMOS) and Comparative MOS (CMOS) to assess speaker similarity and relative naturalness, respectively. Detailed definitions for each metric are provided in Appendix G.

4 Experimental Results

4.1 Codec Comparison

We compare our Gogo with a range of existing codecs, and the results are summarized in Table 1. Despite operating at a relatively low token rate of 47 tokens per second, Gogo achieves superior performance across multiple metrics compared to codecs operating at 50 tokens per second. DAC and EnCodec achieve the best overall performance when operating at a high token rate of 600, with the exception of UT-MOS and DNS-MOS. However, their performance degrades significantly as the token rate is reduced. Notably, Gogo attains UT-MOS and DNS-MOS scores that even surpass the ground-truth recordings, which we attribute to the generative nature of Gogo's flow-matching decoder, enabling enhanced perceptual quality and improved noise robustness.

4.2 Effectiveness of Group-wise Quantization for Autoregressive Modeling

To evaluate whether group-wise quantization better supports downstream AR modeling, we train a naive autoregressive model to perform AR prediction over speech token sequences, which are con-

Table 2: Perplexity of autoregressive modeling on speech tokens produced by different quantization schemes. Column headers specify the source of speech tokens, denoting their positions within each group for group-wise quantization or their corresponding RVQ layer for frame-wise quantization. The boldface denotes the best result. † indicates frame-wise quantization with single-layer VQ.

Scheme	1	2	3	4	5	6	7	8	9	10
Frame-wise †	247.8	-	-	-	-	-	-	-	-	-
Frame-wise	2.3	25.9	84.4	114.9	189.2	261.2	441.8	442.0	727.8	691.4
Group-wise	**0.9**	**8.5**	**42.0**	**96.9**	**169.5**	**201.6**	**204.1**	**221.9**	**229.6**	**228.3**

structed by collecting the j-th token from each group in Gogo, where $j \in [1, n_q]$. For the frame-wise baseline, we remove the grouping operation in Gogo and replace FSQ with a 10-level RVQ, following the standard frame-wise quantization scheme. Each RVQ layer generates a frame-level token sequence that encodes the residual information left by all preceding layers. Each sequence is modeled autoregressively, and perplexity is reported as an indicator of modeling difficulty. Additional experimental details and perplexity computation are provided in Appendix I.

The perplexity results are summarized in Table 2. Group-wise quantization consistently yields lower perplexity across all granularities than frame-wise quantization, suggesting that the group-wise tokens produced by Gogo are more autoregressive-friendly and capture temporal dependencies more effectively. We can also see that coarse tokens yield substantially lower perplexity than fine tokens in both quantization schemes, confirming that fine-grained acoustic details are more challenging for AR models to predict. These findings further motivate the two-stage design of GogoSpeech, where a high-level speech backbone is generated first, followed by fine-grained detail enrichment.

4.3 What Do Granularity-ordered Tokens Encode?

To gain deeper insight into the behavior of group-wise granularity-ordered quantization, we conduct probing experiments on Gogo's tokens at different granularities to examine the type of information each token encodes. We probe these tokens using a diverse set of acoustic, prosodic, and linguistic features. The probing task is formulated as a regression problem, where the mean squared error reported by the probing model is used as an indicator of each token's representational capacity. More details of the probing setup are provided in Appendix H.

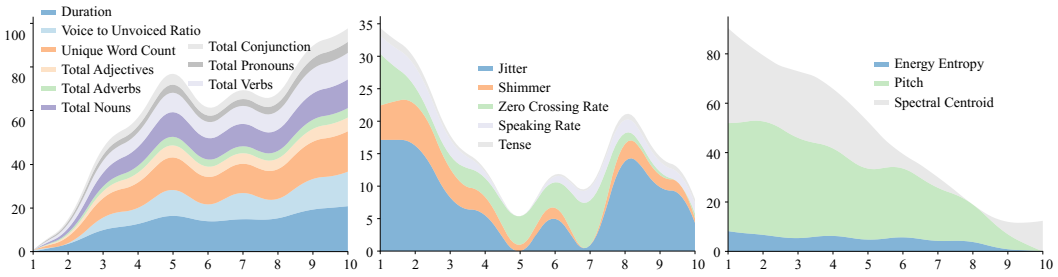

Figure 5: Performance of granularity-ordered tokens across multiple feature prediction tasks. Results are visualized as stacked area charts, where the x-axis denotes token positions within each group and the y-axis indicates the normalized prediction loss relative to the maximum loss for each feature. A higher value corresponds to greater loss and thus lower predictive performance.

The probing results, presented in Figure 5, reveal a clear progression of information across token granularities. Specifically, the first three tokens primarily capture global and high-level information, such as total duration, voiced-to-unvoiced ratio, word count, and linguistic content. Tokens in the middle range predominantly encode prosodic attributes, including speaking rate, jitter, and shimmer. Finally, the last three tokens are responsible for capturing detailed acoustic information, such as pitch, energy, and spectral centroid. These findings confirm that Gogo's group-wise quantization

Table 3: Comparison between different TTS models on the Seed-TTS test-en set. The boldface denotes the best result, the underline denotes the second best.

Model	Objective				Subjective	
	SIM	WER	SIM†	WER†	SMOS	CMOS
Ground Truth	0.734	2.143	0.809	2.037	4.752	0.000
F5-TTS (Chen et al., 2025)	0.647	**1.830**	0.716	<u>1.812</u>	4.173	<u>+1.730</u>
XTTS-v2 (Casanova et al., 2024)	0.463	3.248	0.490	2.292	2.426	−0.961
Llasa-8B-250k (Ye et al., 2025b)	0.574	2.970	0.629	5.947	3.297	+0.882
CosyVoice 2 (Du et al., 2024b)	0.654	2.380	0.701	2.324	<u>4.331</u>	+1.638
FireRedTTS-1S (Guo et al., 2025a)	0.660	<u>2.170</u>	0.705	2.129	4.247	+1.634
VoiceCraft (Peng et al., 2024)	0.470	7.556	0.360	10.25	2.965	−0.751
GogoSpeech (47 Hz)	**0.667**	2.394	**0.725**	**1.788**	**4.381**	**+1.832**
w/ Token Allocator (47 Hz → 36 Hz)	<u>0.662</u>	2.469	<u>0.717</u>	1.845	4.253	+1.587

† All target speeches corresponding to the same prompt speech are concatenated, and only constructed samples longer than 10s are retained to evaluate the stability of systems in long speech generation.

organizes tokens in a coarse-to-fine manner, allowing high-level linguistic and prosodic cues to be modeled with fewer tokens while reserving fine tokens for detailed acoustic reconstruction.

4.4 Zero-Shot TTS Comparison

We compare GogoSpeech with several state-of-the-art TTS baselines, and the results are summarized in Table 3. In objective evaluations, GogoSpeech achieves the highest SIM and competitive WER compared to the leading systems. For long-form speech generation, it further attains the best SIM and WER, demonstrating the effectiveness of the two-stage design in enhancing generation stability. In subjective evaluations, GogoSpeech achieves the best SMOS and CMOS scores, confirming its ability to preserve speaker identity while maintaining strong intelligibility and overall quality.

4.5 Effectiveness of Token Allocator

We further evaluate the effectiveness of the proposed token allocator by comparing GogoSpeech with and without adaptive token allocation. The results are presented in the last two rows of Table 3. With the token allocator, GogoSpeech generates on average only 36 tokens per second of speech, compared to 47 tokens without adaptive allocation. The token allocator significantly reduces the computational cost of speech generation and incurs only a marginal performance degradation in both objective and subjective scores. These results demonstrate that the token allocator achieves a favorable trade-off between generation efficiency and speech quality. Please refer to Appendix K for visualization of adaptive allocation.

5 Related Work

Our related work is put in Appendix L.

6 Conclusion

In this paper, we present Gogo, a group-wise granularity-ordered codec, and GogoSpeech, a two-stage speech language model. Specifically, Gogo produces autoregressive-friendly tokens for each speech group, arranged in order from coarse to fine. Built upon Gogo, GogoSpeech performs speech generation by first constructing a high-level speech backbone, then enriching it with fine-grained details. Furthermore, we proposed a GRPO-trained token allocator that adaptively allocates token budgets based on group-wise complexity, significantly reducing the number of tokens required for synthesis without sacrificing perceptual quality. Extensive experiments on speech reconstruction and zero-shot text-to-speech demonstrate that Gogo achieves superior performance compared to state-of-the-art codecs, and GogoSpeech delivers high-quality, stable, and efficient speech generation.

REFERENCES

Keyu An, Qian Chen, Chong Deng, Zhihao Du, Changfeng Gao, Zhifu Gao, Yue Gu, Ting He, Hangrui Hu, Kai Hu, et al. Funaudiollm: Voice understanding and generation foundation models for natural interaction between humans and llms. *arXiv preprint arXiv:2407.04051*, 2024.

Philip Anastassiou, Jiawei Chen, Jitong Chen, Yuanzhe Chen, Zhuo Chen, Ziyi Chen, Jian Cong, Lelai Deng, Chuang Ding, Lu Gao, et al. Seed-tts: A family of high-quality versatile speech generation models. *arXiv preprint arXiv:2406.02430*, 2024.

Rosana Ardila, Megan Branson, Kelly Davis, Michael Henretty, Michael Kohler, Josh Meyer, Reuben Morais, Lindsay Saunders, Francis M Tyers, and Gregor Weber. Common voice: A massively-multilingual speech corpus. *arXiv preprint arXiv:1912.06670*, 2019.

Kushal Arora, Layla El Asri, Hareesh Bahuleyan, and Jackie Chi Kit Cheung. Why exposure bias matters: An imitation learning perspective of error accumulation in language generation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 700–710, 2022.

James Betker. Better speech synthesis through scaling. *arXiv preprint arXiv:2305.07243*, 2023.

Zalán Borsos, Raphaël Marinier, Damien Vincent, Eugene Kharitonov, Olivier Pietquin, Matt Sharifi, Dominik Roblek, Olivier Teboul, David Grangier, Marco Tagliasacchi, et al. Audiolm: a language modeling approach to audio generation. *IEEE/ACM transactions on audio, speech, and language processing*, 31:2523–2533, 2023.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Edresson Casanova, Kelly Davis, Eren Gölge, Görkem Göknar, Iulian Gulea, Logan Hart, Aya Aljafari, Joshua Meyer, Reuben Morais, Samuel Olayemi, and Julian Weber. Xtts: a massively multilingual zero-shot text-to-speech model. *Interspeech*, pp. 4978–4982, 2024.

Yunkee Chae, Woosung Choi, Yuhta Takida, Junghyun Koo, Yukara Ikemiya, Zhi Zhong, Kin Wai Cheuk, Marco A Martínez-Ramírez, Kyogu Lee, Wei-Hsiang Liao, et al. Variable bitrate residual vector quantization for audio coding. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1–5, 2025.

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, et al. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1505–1518, 2022.

Yushen Chen, Zhikang Niu, Ziyang Ma, Keqi Deng, Chunhui Wang, Jian Zhao, Kai Yu, and Xie Chen. F5-tts: A fairytaler that fakes fluent and faithful speech with flow matching. *arXiv preprint arXiv:2410.06885*, 2025.

Yu-An Chung, Yu Zhang, Wei Han, Chung-Cheng Chiu, James Qin, Ruoming Pang, and Yonghui Wu. W2v-bert: Combining contrastive learning and masked language modeling for self-supervised speech pre-training. In *IEEE Automatic Speech Recognition and Understanding Workshop*, pp. 244–250, 2021.

Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. High fidelity neural audio compression. *arXiv preprint arXiv:2210.13438*, 2022.

Alexandre Défossez, Laurent Mazaré, Manu Orsini, Amélie Royer, Patrick Pérez, Hervé Jégou, Edouard Grave, and Neil Zeghidour. Moshi: a speech-text foundation model for real-time dialogue. *arXiv preprint arXiv:2410.00037*, 2024.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies*, volume 1, pp. 4171–4186, 2019.

Sander Dieleman, Charlie Nash, Jesse Engel, and Karen Simonyan. Variable-rate discrete representation learning. *arXiv preprint arXiv:2103.06089*, 2021.

Zhihao Du, Qian Chen, Shiliang Zhang, Kai Hu, Heng Lu, Yexin Yang, Hangrui Hu, Siqi Zheng, Yue Gu, Ziyang Ma, et al. Cosyvoice: A scalable multilingual zero-shot text-to-speech synthesizer based on supervised semantic tokens. *arXiv preprint arXiv:2407.05407*, 2024a.

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

Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12873–12883, 2021.

Qingkai Fang, Shoutao Guo, Yan Zhou, Zhengrui Ma, Shaolei Zhang, and Yang Feng. LLaMA-omni: Seamless speech interaction with large language models. In *The Thirteenth International Conference on Learning Representations*, 2025.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Hao-Han Guo, Yao Hu, Fei-Yu Shen, Xu Tang, Yi-Chen Wu, Feng-Long Xie, and Kun Xie. Fireredtts-1s: An upgraded streamable foundation text-to-speech system. *arXiv preprint arXiv:2503.20499*, 2025a.

Yiwei Guo, Zhihan Li, Hankun Wang, Bohan Li, Chongtian Shao, Hanglei Zhang, Chenpeng Du, Xie Chen, Shujie Liu, and Kai Yu. Recent advances in discrete speech tokens: A review. *arXiv preprint arXiv:2502.06490*, 2025b.

Haorui He, Zengqiang Shang, Chaoren Wang, Xuyuan Li, Yicheng Gu, Hua Hua, Liwei Liu, Chen Yang, Jiaqi Li, Peiyang Shi, et al. Emilia: An extensive, multilingual, and diverse speech dataset for large-scale speech generation. In *IEEE Spoken Language Technology Workshop*, pp. 885–890, 2024.

Tianxing He, Jingzhao Zhang, Zhiming Zhou, and James Glass. Exposure bias versus self-recovery: Are distortions really incremental for autoregressive text generation? In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 5087–5102, 2021.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

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

Shengpeng Ji, Ziyue Jiang, Wen Wang, Yifu Chen, Minghui Fang, Jialong Zuo, Qian Yang, Xize Cheng, Zehan Wang, Ruiqi Li, Ziang Zhang, Xiaoda Yang, Rongjie Huang, Yidi Jiang, Qian Chen, Siqi Zheng, and Zhou Zhao. Wavtokenizer: an efficient acoustic discrete codec tokenizer for audio language modeling. In *The Thirteenth International Conference on Learning Representations*, 2025.

Daejin Jo, Jeeyoung Yun, Byungseok Roh, and Sungwoong Kim. Lm-spt: Lm-aligned semantic distillation for speech tokenization. *arXiv preprint arXiv:2506.16738*, 2025.

Tero Karras, Miika Aittala, Jaakko Lehtinen, Janne Hellsten, Timo Aila, and Samuli Laine. Analyzing and improving the training dynamics of diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24174–24184, 2024.

Eugene Kharitonov, Damien Vincent, Zalán Borsos, Raphaël Marinier, Sertan Girgin, Olivier Pietquin, Matt Sharifi, Marco Tagliasacchi, and Neil Zeghidour. Speak, read and prompt: High-fidelity text-to-speech with minimal supervision. *Transactions of the Association for Computational Linguistics*, 11:1703–1718, 2023.

W Bastiaan Kleijn, Andrew Storus, Michael Chinen, Tom Denton, Felicia SC Lim, Alejandro Luebs, Jan Skoglund, and Hengchin Yeh. Generative speech coding with predictive variance regularization. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 6478–6482, 2021.

Rithesh Kumar, Prem Seetharaman, Alejandro Luebs, Ishaan Kumar, and Kundan Kumar. High-fidelity audio compression with improved rvqgan. *Advances in Neural Information Processing Systems*, 36:27980–27993, 2023.

Kushal Lakhotia, Eugene Kharitonov, Wei-Ning Hsu, Yossi Adi, Adam Polyak, Benjamin Bolte, Tu-Anh Nguyen, Jade Copet, Alexei Baevski, Abdelrahman Mohamed, et al. On generative spoken language modeling from raw audio. *Transactions of the Association for Computational Linguistics*, 9:1336–1354, 2021.

Jiaqi Li, Xiaolong Lin, Zhekai Li, Shixi Huang, Yuancheng Wang, Chaoren Wang, Zhenpeng Zhan, and Zhizheng Wu. Dualcodec: A low-frame-rate, semantically-enhanced neural audio codec for speech generation. *arXiv preprint arXiv:2505.13000*, 2025.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *11th International Conference on Learning Representations*, 2023.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017a.

Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2017b.

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

OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2024.

Julian D Parker, Anton Smirnov, Jordi Pons, CJ Carr, Zack Zukowski, Zach Evans, and Xubo Liu. Scaling transformers for low-bitrate high-quality speech coding. In *The Thirteenth International Conference on Learning Representations*, 2025.

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

Puyuan Peng, Po-Yao Huang, Shang-Wen Li, Abdelrahman Mohamed, and David Harwath. 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*, pp. 12442–12462, 2024.

Chandan KA Reddy, Vishak Gopal, and Ross Cutler. Dnsmos p. 835: A non-intrusive perceptual objective speech quality metric to evaluate noise suppressors. In *IEEE international conference on acoustics, speech and signal processing*, pp. 886–890, 2022.

Oren Rippel, Michael Gelbart, and Ryan Adams. Learning ordered representations with nested dropout. In *International Conference on Machine Learning*, pp. 1746–1754, 2014.

Antony W Rix, John G Beerends, Michael P Hollier, and Andries P Hekstra. Perceptual evaluation of speech quality (pesq)-a new method for speech quality assessment of telephone networks and codecs. In *IEEE international conference on acoustics, speech, and signal processing*, volume 2, pp. 749–752, 2001.

Paul K Rubenstein, Chulayuth Asawaroengchai, Duc Dung Nguyen, Ankur Bapna, Zalán Borsos, Félix de Chaumont Quitry, Peter Chen, Dalia El Badawy, Wei Han, Eugene Kharitonov, et al. Audiopalm: A large language model that can speak and listen. *arXiv preprint arXiv:2306.12925*, 2023.

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

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Feiyu Shen, Yiwei Guo, Chenpeng Du, Xie Chen, and Kai Yu. Acoustic bpe for speech generation with discrete tokens. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 11746–11750, 2024.

Maohao Shen, Shun Zhang, Jilong Wu, Zhiping Xiu, Ehab AlBadawy, Yiting Lu, Mike Seltzer, and Qing He. Get large language models ready to speak: A late-fusion approach for speech generation. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1–5, 2025.

Hubert Siuzdak. Vocos: Closing the gap between time-domain and fourier-based neural vocoders for high-quality audio synthesis. *arXiv preprint arXiv:2306.00814*, 2023.

Hubert Siuzdak, Florian Grötschla, and Luca A Lanzendörfer. Snac: Multi-scale neural audio codec. *arXiv preprint arXiv:2410.14411*, 2024.

Yakun Song, Jiawei Chen, Xiaobin Zhuang, Chenpeng Du, Ziyang Ma, Jian Wu, Jian Cong, Dongya Jia, Zhuo Chen, Yuping Wang, et al. Magicodec: Simple masked gaussian-injected codec for high-fidelity reconstruction and generation. *arXiv preprint arXiv:2506.00385*, 2025.

Cees H Taal, Richard C Hendriks, Richard Heusdens, and Jesper Jensen. A short-time objective intelligibility measure for time-frequency weighted noisy speech. In *IEEE international conference on acoustics, speech and signal processing*, pp. 4214–4217, 2010.

Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *Transactions on Machine Learning Research*, pp. 1–34, 2024.

Jean-Marc Valin, Koen Vos, and Timothy Terriberry. Definition of the opus audio codec. *IETF RFC 6716*, 2012.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Stephen D Voran. Why some audio signal short-time fourier transform coefficients have nonuniform phase distributions. In *IEEE International Conference on Multimedia and Expo*, pp. 1–6, 2024.

Chengyi Wang, Sanyuan Chen, Yu Wu, Ziqiang Zhang, Long Zhou, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, et al. Neural codec language models are zero-shot text to speech synthesizers. *arXiv preprint arXiv:2301.02111*, 2023.

Hanyu Wang, Saksham Suri, Yixuan Ren, Hao Chen, and Abhinav Shrivastava. LARP: Tokenizing videos with a learned autoregressive generative prior. In *The Thirteenth International Conference on Learning Representations*, 2025.

Sanghyun Woo, Shoubhik Debnath, Ronghang Hu, Xinlei Chen, Zhuang Liu, In So Kweon, and Saining Xie. Convnext v2: Co-designing and scaling convnets with masked autoencoders. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16133–16142, 2023.

Dongchao Yang, Songxiang Liu, Haohan Guo, Jiankun Zhao, Yuanyuan Wang, Helin Wang, Zeqian Ju, Xubo Liu, Xueyuan Chen, Xu Tan, et al. Almtokenizer: A low-bitrate and semantic-rich audio codec tokenizer for audio language modeling. *arXiv preprint arXiv:2504.10344*, 2025.

Zhen Ye, Peiwen Sun, Jiahe Lei, Hongzhan Lin, Xu Tan, Zheqi Dai, Qiuqiang Kong, Jianyi Chen, Jiahao Pan, Qifeng Liu, et al. Codec does matter: Exploring the semantic shortcoming of codec for audio language model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 25697–25705, 2025a.

Zhen Ye, Xinfa Zhu, Chi-Min Chan, Xinsheng Wang, Xu Tan, Jiahe Lei, Yi Peng, Haohe Liu, Yizhu Jin, Zheqi Dai, et al. Llasa: Scaling train-time and inference-time compute for llama-based speech synthesis. *arXiv preprint arXiv:2502.04128*, 2025b.

Jiahui Yu, Xin Li, Jing Yu Koh, Han Zhang, Ruoming Pang, James Qin, Alexander Ku, Yuanzhong Xu, Jason Baldridge, and Yonghui Wu. Vector-quantized image modeling with improved VQ-GAN. In *International Conference on Learning Representations*, 2022.

Neil Zeghidour, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi. Soundstream: An end-to-end neural audio codec. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:495–507, 2021.

Heiga Zen, Viet Dang, Rob Clark, Yu Zhang, Ron J Weiss, Ye Jia, Zhifeng Chen, and Yonghui Wu. Libritts: A corpus derived from librispeech for text-to-speech. *arXiv preprint arXiv:1904.02882*, 2019.

Aohan Zeng, Zhengxiao Du, Mingdao Liu, Lei Zhang, Yuxiao Dong, Jie Tang, et al. Scaling speech-text pre-training with synthetic interleaved data. In *The Thirteenth International Conference on Learning Representations*, 2025.

Dong Zhang, Shimin Li, Xin Zhang, Jun Zhan, Pengyu Wang, Yaqian Zhou, and Xipeng Qiu. Speechgpt: Empowering large language models with intrinsic cross-modal conversational abilities. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 15757–15773, 2023.

Hanglei Zhang, Yiwei Guo, Zhihan Li, Xiang Hao, Xie Chen, and Kai Yu. Unlocking temporal flexibility: Neural speech codec with variable frame rate. *arXiv preprint arXiv:2505.16845*, 2025.

Xin Zhang, Dong Zhang, Shimin Li, Yaqian Zhou, and Xipeng Qiu. Speechtokenizer: Unified speech tokenizer for speech language models. In *The Twelfth International Conference on Learning Representations*, 2024.

A Nested Droupuot

Due to the nature of nested dropout, coarse tokens are more likely to be retained during training, whereas fine tokens are more likely to be dropped, leading to fewer effective gradient updates for the latter. To compensate for this imbalance, we rescale the gradient of each token according to its retention probability p. Specifically, the coarsest token, *i.e.*, the first learnable token in each group, is always retained, while the finest token, *i.e.*, the last learnable token, is retained with probability $1/n_q$, since we uniformly sample the number of tokens to retain. More generally, the retention probability of the j-th token is defined as:

$$p_j = 1 - \frac{j-1}{n_q}, \text{ where } j \in [1, n_q]. \tag{14}$$

We fix the compensation weight for the first token to 0.5, and accordingly define the weight for each token as:

$$w_j = \frac{0.5}{p_j}, \text{ where } j \in [1, n_q]. \tag{15}$$

Let the quantized features of the i-th group be $\bar{q}^i \in \mathbb{R}^{n_q \times d_h}$. During the forward pass, we apply the reparameterization trick to scale the token representations as:

$$\bar{q}^i[j] \leftarrow \frac{0.5}{p_j}\bar{q}^i[j] + \left(\bar{q}^i[j] - \frac{0.5}{p_j}\bar{q}^i[j]\right).\text{detach}, \tag{16}$$

where $\bar{q}^i[j]$ denotes the j-th token in group i, and the detach operator indicates that the gradient flow is stopped for the detached variable. By applying Equation 16, the value of each embedding $\bar{q}^i[j]$ remains unchanged during the forward pass. However, in the backward pass, its gradient is scaled by the inverse of p_j to offset the imbalance in p_j. As a result, $\bar{q}^i[j]$ that receive fewer gradient updates are compensated with a larger gradient scale.

B Auxiliary Modules

The AR prior takes the quantized representations $\bar{q}^i$ as input and predicts the feature representation of the next speech token at every position. To enhance training stability, the AR prior is optimized with a mean squared error loss in the feature space. Let f denotes the AR prior with parameters η. The AR loss is defined as:

$$\mathcal{L}_{\text{AR}} = \mathbb{E}_i \left[\frac{1}{n_q - 1} \sum_{j=1}^{n_q - 1} \left\| f_\eta\left(\bar{q}^i[1:j]\right) - \bar{q}^i[j+1] \right\|_2^2 \right]. \tag{17}$$

Furthermore, we incorporate an ASR module into the Gogo training pipeline to facilitate the linguistic representation learning. The group-wise quantized representations $\bar{q}^i \in \mathbb{R}^{n_q \times d_h}$ are concatenated along the temporal dimension to form $\bar{x}_s \in \mathbb{R}^{n_q(n_f/g) \times d_h}$. The ASR module, denoted as h_ϕ, takes $\bar{x}_s$ as input and outputs a predicted token sequence $\hat{y} = (\hat{y}_1, \ldots, \hat{y}_L)$ corresponding to the ground-truth transcription $y = (y_1, \ldots, y_L)$. The ASR loss is formulated as the cross entropy loss:

$$\mathcal{L}_{\text{ASR}} = \mathbb{E}_{(\bar{x}_s, y)} \left[-\frac{1}{L} \sum_{t=1}^{L} \log h_\phi(y_t \mid y_{<t}, \bar{x}_s) \right]. \tag{18}$$

C Model Configuration Details

For Gogo, the Transformer encoder, ASR module, and AR prior are implemented with 12, 8, and 4 layers of standard Transformer blocks (Vaswani et al., 2017), respectively, following the architecture of LLaMA (Grattafiori et al., 2024). Each layer uses a hidden dimension of 512, 8 attention heads, a feed-forward dimension of 1536, Rotary Position Embeddings (RoPE), RMSNorm, and SwiGLU activation. We set the group size to $g = 20$, and allocate $n_q = 10$ speech queries per group. The feature quantization adopts FSQ with levels $[8, 8, 8, 5, 5]$, yielding an effective codebook size of 12,800. The flow-matching model in Gogo is implemented as a latent Diffusion Transformer (DiT) (Peebles & Xie, 2023), following the configuration of Chen et al. (2025). The only modification is that we set the number of layers to 12. Additionally, the input mel-spectrograms of Gogo are first processed by four ConvNeXt V2 (Woo et al., 2023) layers, followed by a two-layer MLP that projects the features to match the hidden dimension of Gogo.

For GogoSpeech, both Stage I and Stage II are built on top of Llama-3.2-1B-Instruct, with the vocabulary extended to include Gogo's codebook tokens, enabling the model to directly perform speech token generation. GogoSpeech is trained under the next-token prediction paradigm to jointly model text and speech tokens. The maximum sequence length is set to 256 and 1024 tokens for Stage I and Stage II, respectively. The speech backbone is defined as the first $b = 3$ tokens of each group, which are generated in Stage I. The remaining 7 tokens per group are recovered in Stage II.

For the token allocator, we adopt a lightweight Transformer with 2 layers and the same architectural configuration used in Gogo, followed by a linear classifier to predict the token budget for each group.

D Training and Inference Details

Gogo, GogoSpeech, and the token allocator are optimized separately using the AdamW optimizer (Loshchilov & Hutter, 2017a) on 8 NVIDIA H100 NVL 94G GPUs. The learning rate is decayed

based on a cosine annealing schedule (Loshchilov & Hutter, 2017b). Speech samples longer than 20 seconds or shorter than 1 seconds are discarded during training. More detailed training hyperparameters are shown in Table 4.

Table 4: Hyperparameters for training Gogo, GogoSpeech, and the token allocator.

Hyperparameters	Gogo	GogoSpeech Stage I / Stage II	Token Allocator
Training Epochs	-	10 / 5	1
Update Steps	400k	-	-
Warmup Steps	10k	5k / 10k	1k
Batch Size	1440 Seconds	1152 / 288 Samples	128 Samples
Learning Rate	2e-4	5e-4	1e-4
Optimizer	AdamW	AdamW	AdamW
Momentum	$\beta_1, \beta_2 = 0.9, 0.999$	$\beta_1, \beta_2 = 0.9, 0.95$	$\beta_1, \beta_2 = 0.9, 0.999$
Weight Decay	0.01	0.01	0.01
Learning Rate Schedule	Cosine Annealing	Cosine Annealing	Cosine Annealing

For Gogo inference, we begin from sampled Gaussian noise x_0 and integrate it toward the target distribution x_1 conditioned on the speech tokens s^i. Following Eq. 1 and Eq. 2, the tokens s^i are first converted into aligned representations $\bar{x}$. We then employ an Euler ordinary differential equation (ODE) solver to iteratively integrate $\partial x_t / \partial t = v_\theta(x_t, \bar{x}, t)$ from x_0 to x_1, where the flow step t is sampled using the Sway Sampling strategy (Chen et al., 2025) to improve generation performance and efficiency. The resulting mel-spectrogram x_1 is converted into waveform using a pretrained Vocos vocoder (Siuzdak, 2023). To balance fidelity and diversity, we apply Classifier-Free Guidance (CFG) (Ho & Salimans, 2022) with a guidance scale of 2. For further stability, Exponential Moving Averaged (EMA) weights (Karras et al., 2024) of the model parameters are used during inference.

For GogoSpeech inference, we employ the standard autoregressive decoding strategy widely used in large language models. To balance generation diversity and fidelity, we set the temperature to 0.8, apply a repetition penalty of 1.2 to mitigate degenerate loops, and use nucleus sampling with $p = 1.0$. In Stage II of GogoSpeech, decoding for each group employs early stopping, terminating as soon as the maximum token budget assigned by the token allocator is consumed.

For token allocator inference, the first b tokens of each group (three tokens in our case) produced by Stage I of GogoSpeech are used as input. The allocator then predicts a budget for the group, specifying how many fine tokens should be generated in Stage II.

E Codec Baselines

EnCodec (Défossez et al., 2022) builds upon the residual vector quantization (RVQ) framework and employs a single multi-scale STFT-based discriminator to effectively suppress artifacts and enhance perceptual quality. It supports variable bandwidths by selecting different numbers of codebooks during training, providing a flexible solution for speech compression and discretization in speech language models. For evaluation, we adopt the official implementation and pretrained checkpoint[1].

DAC (Kumar et al., 2023) adapts advances from the Improved VQGAN (Yu et al., 2022) image model to address the codebook collapse problem. It performs codebook lookup in a low-dimensional space and replaces Euclidean distance with cosine similarity, improving both stability and quality. DAC further employs a multi-period discriminator in the waveform domain and a multi-band, multi-scale STFT discriminator in the frequency domain, enabling high-fidelity audio generation. For evaluation, we use the official implementation and pretrained checkpoint[2].

SpeechTokenizer (Zhang et al., 2024) is specifically designed for speech language modeling, where different aspects of speech are disentangled hierarchically across RVQ layers. Specifically, it em-

[1] https://github.com/facebookresearch/encodec

[2] https://github.com/descriptinc/descript-audio-codec

ploys HuBERT as a semantic teacher to distill content information into the first layer of RVQ. For evaluation, we use the official implementation and pretrained checkpoint[3].

Mimi (Défossez et al., 2024) takes inspiration from previous work on SpeechTokenizer and uses distillation to transfer high-level semantic information in WavLM into the quantized tokens. Unlike SpeechTokenizer, it distills semantic information into a plain VQ and apply an RVQ with 7 levels in parallel, thereby removing the constraint that acoustic information must reside in the residual of the semantic quantizer. For evaluation, we use the official implementation and pretrained checkpoint[4].

SNAC (Siuzdak et al., 2024) extends RVQ by allowing quantizers to operate at different temporal resolutions. Through a hierarchy of quantizers running at variable frame rates, it adapts to audio structure across multiple timescales, thereby capturing both coarse and fine details more effectively. Each quantizer applies average pooling for downsampling and nearest-neighbor interpolation for upsampling, enabling efficient compression. For evaluation, we adopt the official implementation and pretrained checkpoint[5].

WavTokenizer (Ji et al., 2025) improves subjective quality using a single quantizer with an expanded codebook to reduce information loss. It further enhances semantic modeling by introducing attention modules with extended contextual windows in the decoder. Finally, the inverse Fourier transform is employed to reconstruct the final audio directly. For evaluation, we adopt the official implementation and pretrained checkpoint[6].

MagiCodec (Song et al., 2025) is a single-layer, streaming Transformer-based audio codec trained with a multistage pipeline to mitigate codebook collapse and improve token efficiency. It introduces Gaussian noise injection and latent regularization techniques to encourage learning low-frequency semantic representations while preventing overfitting to high-frequency noise. For evaluation, we adopt the official implementation and pretrained checkpoint[7].

X-codec2 (Ye et al., 2025b) integrates semantic and acoustic features into a unified codebook using a single-layer FSQ quantizer. A pretrained w2v-BERT serves as the semantic encoder, while an acoustic encoder based on residual convolutional blocks with Snake activations captures fine-grained acoustic details. The semantic and acoustic features are concatenated and served as input to the vector quantizer. For evaluation, we adopt the official implementation and pretrained checkpoint[8].

TAAE (Parker et al., 2025) introduces a Transformer-based codec architecture that scales into the 1B parameter range, enabling state-of-the-art speech quality at extremely low bitrates. Unlike CNN-based codecs that rely on convolutional inductive biases with high parameter efficiency, TAAE leverages a more general Transformer architecture for greater scalability and better modeling capacity. For evaluation, we adopt the official implementation and pretrained checkpoint[9].

DualCodec (Li et al., 2025) is a dual-stream codec that jointly models self-supervised and waveform representations within an end-to-end framework. The first RVQ layer directly encodes semantic-rich features from a pretrained w2v-BERT-2 model, while the remaining RVQ layers, along with the encoder–decoder design, follow the DAC framework. This integration enables DualCodec to better preserve linguistic content while maintaining high-fidelity reconstruction. For evaluation, we use the official implementation and pretrained checkpoint[10].

F TTS BASELINES

FireRedTTS-1S (Guo et al., 2025a) is a high-quality streamable TTS system that achieves real-time speech generation with low latency under 150ms through text-to-semantic decoding and semantic-to-acoustic decoding. For evaluation, we use the pretrained checkpoint[11].

[3] https://github.com/ZhangXInFD/SpeechTokenizer
[4] https://huggingface.co/kyutai/mimi
[5] https://github.com/hubertsiuzdak/snac
[6] https://github.com/jishengpeng/WavTokenizer
[7] https://github.com/Ereboas/MagiCodec
[8] https://huggingface.co/HKUSTAudio/xcodec2
[9] https://github.com/Stability-AI/stable-codec
[10] https://github.com/jiaqili3/DualCodec
[11] https://github.com/FireRedTeam/FireRedTTS

F5-TTS Chen et al. (2025) is a non-autoregressive TTS model built on flow matching. Instead of relying on complex alignment mechanisms, F5-TTS pads the text input with filler tokens to match the length of the target speech and directly performs denoising to generate speech. For evaluation, we use the pretrained checkpoint[12].

XTTS-v2 (Casanova et al., 2024) is a multilingual zero-shot multi-speaker TTS model built upon the Tortoise model (Betker, 2023). XTTS-v2 supports 16 languages and achieves state-of-the-art results in most of them. For evaluation, we use the pretrained checkpoint[13].

Llasa (Ye et al., 2025b) is a large-scale speech synthesis system that employs a single-layer vector quantizer codec and a unified Transformer architecture to fully align with standard LLMs such as Llama. For evaluation, we use the pretrained checkpoint[14].

CosyVoice 2 (Du et al., 2024b) is a multilingual speech synthesis framework that integrates a pre-trained language model for discrete speech token prediction with a chunk-aware flow-matching model for speech feature generation. For evaluation, we use the pretrained checkpoint[15].

VoiceCraft (Peng et al., 2024) is a token infilling neural codec language mode. It employs a token rearrangement strategy with causal masking and delayed stacking, enabling seamless speech editing and zero-shot text-to-speech generation. For evaluation, we use the pretrained checkpoint[16].

G Evaluation Metrics

We employ a comprehensive set of evaluation metrics to assess speech quality across multiple dimensions, including intelligibility, perceptual quality, content preservation, and speaker similarity.

Short-Time Objective Intelligibility (STOI) (Taal et al., 2010) is a widely adopted metric for evaluating speech intelligibility. It computes the correlation between temporal envelopes of reference and reconstructed signals in short-time segments. The score ranges from 0 to 1, with higher values indicating better intelligibility.

Perceptual Evaluation of Speech Quality (PESQ) (Rix et al., 2001) measures perceptual speech quality by comparing the reconstructed audio with the clean reference signal using a perceptual auditory model. We report results under both narrow-band (NB, 8 kHz) and wide-band (WB, 16 kHz) conditions.

UTokyo-SaruLab MOS (UT-MOS) (Saeki et al., 2022) is an automatic MOS predictor trained to approximate human judgments of overall speech naturalness and quality. It provides a scalable alternative to subjective MOS tests.

Deep Noise Suppression MOS (DNS-MOS) (Reddy et al., 2022) is a non-intrusive quality metric designed for real-world audio evaluation. It estimates perceptual quality directly from the signal without requiring reference audio and has been shown to correlate strongly with human ratings.

Word Error Rate (WER) is used to quantify content preservation and intelligibility at the linguistic level. By default, we adopt HuBERT as the ASR model[17] to transcribe the reconstructed speech, and compute WER by comparing against the ground-truth transcripts. For zero-shot TTS evaluation on the Seed-TTS test-en set, the WER metric is computed using the provided script[18].

Speaker Similarity (SIM) measures the degree to which the synthesized speech retains the identity of the original speaker. By default, we follow Wang et al. (2023) and employ a WavLM-Large-based speaker verification model[19] to extract speaker embeddings from both reconstructed and reference

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

[13] https://huggingface.co/coqui/XTTS-v2

[14] https://huggingface.co/HKUSTAudio/Llasa-8B

[15] https://huggingface.co/FunAudioLLM/CosyVoice2-0.5B

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

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

[18] https://github.com/BytedanceSpeech/seed-tts-eval

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

speech, and compute cosine similarity as the final metric. For zero-shot TTS evaluation on the Seed-TTS test-en set, the SIM metric is computed using the provided script[18].

Similarity Mean Opinion Score (SMOS) evaluates the speaker similarity between the prompt and the generated speech. Human raters judge the degree of resemblance by considering speaker characteristics, style, acoustic properties, and potential background artifacts. SMOS is scored on a five-point scale, with higher values indicating stronger similarity.

Comparative Mean Opinion Score (CMOS) measures the relative perceptual quality of a synthesized sample compared to a reference. Raters assign scores on a scale from -3 to 3, where negative values indicate the synthesized speech is worse than the reference, positive values indicate it is better, and 0 denotes parity.

H PROBING EXPERIMENTS

The probing model consists of three fully connected layers with ReLU activations and dropout in between. The hidden dimensions of each layer are 512, 128, and 1, respectively. As illustrated in Figure 6, the probing procedure is formulated as a regression task: the input is the average of tokens at a specific position across all groups, and the output is the predicted value of the target feature. The mean squared error (MSE) loss of the probing model is reported as an indicator of the representational capacity of tokens at each position, with lower loss implying that the token more readily encodes the probed feature.

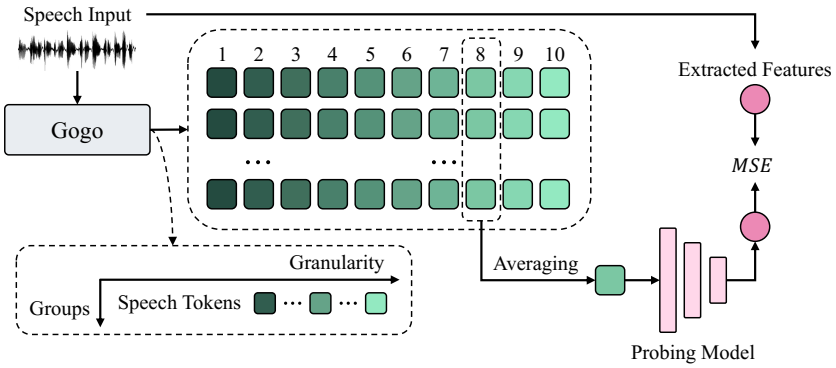

Figure 6: Procedure for probing the information encoded in tokens of different granularities. Throughout the probing model training, the Gogo codec is kept frozen. Here we probe the tokens at position 8 for demonstration.

We conduct probing experiments on Gogo's granularity-ordered tokens using a broad set of features spanning acoustic, prosodic, and linguistic dimensions. Acoustic features include zero-crossing rate, mean pitch, energy entropy, and spectral centroid, which characterize low-level signal properties. Prosodic features include jitter, shimmer, duration, voiced–unvoiced ratio, and speaking rate, capturing rhythm, intonation, and phonation stability. Linguistic features include lexical statistics such as unique word count and counts of adjectives, adverbs, nouns, verbs, pronouns, and conjunctions, reflecting higher-level syntactic content. The probing models are trained on the LibriTTS train-clean-100 subset for 20k steps using the AdamW optimizer and evaluated on the LibriTTS test-clean set. The detailed MSE losses for each feature are reported in Table 5.

We further investigate how reconstruction quality varies when only a subset of the granularity-ordered tokens is used. Specifically, we progressively retain the first n tokens in each group and reconstruct the input speech, where $n \in [1, n_q]$. The results are shown in Figure 7. We observe that the word error rate drops sharply when the first few tokens are included, indicating that most linguistic content is captured by the coarsest tokens in Gogo. However, when more than six tokens per group are retained, the improvement of WER becomes marginal, suggesting that the remaining fine-grained tokens primarily contribute to perceptual quality and acoustic details rather than linguistic content. In particular, both NB PESQ and WB PESQ scores show a marked improvement

Table 5: MSE losses for various probed features using Gogo's granularity-ordered tokens on the LibriTTS test-clean set. In our setting, each frame group is discretized into 10 tokens. Token position 1 corresponds to the coarsest token, whereas position 10 corresponds to the finest; ZRC: zero crossing rate, EE: energy entropy, SC: spectral centroid, J: jitter, S: Shimmer, Ratio: voiced to unvoiced ratio, SR: speaking rate, Count: unique word count, Tadj: total adjectives, Tadv: total adverbs, Tn: total nouns, Tv: total verbs, Tpron: total pronouns, Tconj: total conjunction.

Features	1	2	3	4	5	6	7	8	9	10
Duration	**14.44**	14.95	15.87	16.28	16.81	16.45	16.59	16.65	17.23	17.45
ZRC (e-3)	1.113	1.058	1.051	1.060	1.076	1.071	1.101	1.047	**1.036**	1.038
EE (e-3)	8.409	8.290	8.189	8.257	8.142	8.212	8.101	8.066	7.839	**7.771**
SC (e-3)	1.150	1.053	1.054	1.031	0.988	0.879	0.866	**0.830**	0.872	0.933
Pitch	1455	1478	1424	1371	1304	1295	1230	1166	1074	**1012**
J (e-5)	4.000	3.969	3.696	3.603	**3.415**	3.584	3.445	3.887	3.751	3.563
S (e-4)	3.463	3.497	3.435	3.382	3.321	3.345	**3.290**	3.377	3.350	3.312
Ratio	0.298	**0.297**	0.313	0.318	0.332	0.319	0.332	0.323	0.338	0.343
SR	0.414	0.415	0.413	0.406	**0.403**	0.406	0.411	0.411	0.409	0.409
Count	**35.81**	36.90	39.01	40.00	41.18	40.34	40.63	40.70	41.96	42.47
Tense	0.240	0.239	0.238	0.237	**0.236**	0.237	0.237	0.239	0.239	0.238
Tadj	**2.229**	2.250	2.299	2.329	2.352	2.334	2.337	2.338	2.359	2.370
Tadv	**1.474**	1.486	1.507	1.521	1.532	1.523	1.521	1.526	1.537	1.538
Tn	**9.716**	9.907	10.32	10.55	10.81	10.67	10.71	10.66	10.88	10.99
Tv	**4.708**	4.804	4.969	5.023	5.128	5.057	5.097	5.116	5.240	5.282
Tpron	**2.766**	2.797	2.838	2.847	2.868	2.847	2.853	2.866	2.897	2.906
Tconj	**0.967**	0.977	0.998	1.005	1.015	1.003	1.006	1.010	1.024	1.027

once more than four tokens per group are retained. The other objective metrics exhibit a generally monotonic improvement as the number of retained tokens increases.

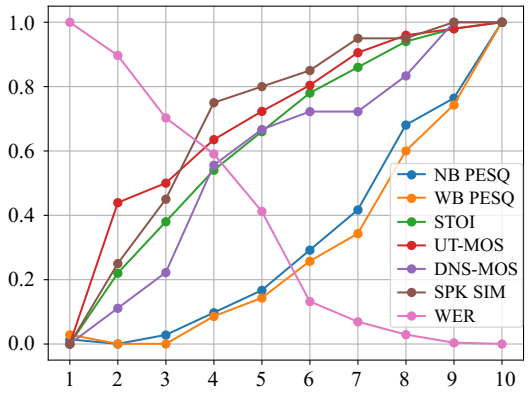

Figure 7: Normalized performance on the LibriTTS test-clean set with varying numbers of retained tokens per group.

I Perplexity Experiments

Perplexity (PPL) is a standard evaluation metric for language models that quantifies their uncertainty in predicting the next token in a sequence. Lower PPL indicates higher model confidence and prediction accuracy, whereas higher PPL reflects greater uncertainty and poorer predictive performance. Formally, perplexity is defined as the exponentiated average negative log-likelihood of the sequence. In our setting, given a speech token sequence $\Gamma(\mathbf{S}_{:,j})$, where $\mathbf{S} \in \mathbb{R}^{n_g \times n_q}$ is the token matrix produced by Gogo and $j \in [1, n_q]$ denotes the position of the speech token within each group, we compute PPL to assess the autoregressive modeling difficulty at different token granularities as

follow:

$$\text{PPL}\left(\Gamma(\mathbf{S}_{:,j})\right) = \exp\left\{-\frac{1}{n_g}\sum_{i}^{n_g}\log P(\mathbf{S}_{i,j}|\mathbf{S}_{<i,j})\right\}. \tag{19}$$

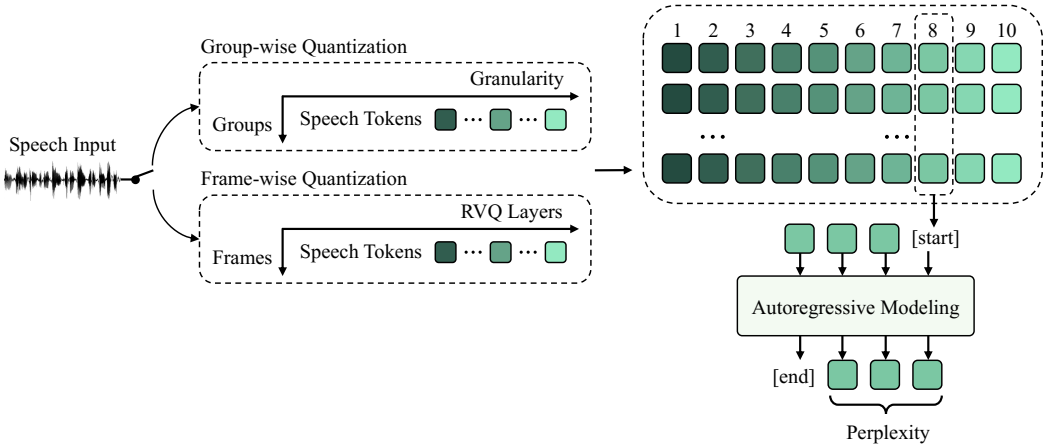

Figure 8: Procedure for evaluating the perplexity of autoregressive model on speech tokens generated by different quantization schemes. Here we show the evaluation using the 8th token of each group or tokens from the 8th RVQ layer for demonstration.

As illustrated in Figure 8, the autoregressive model used in the perplexity experiments is a 6-layer LLaMA-style Transformer with hidden dimension of 256. It is trained on the combined LibriTTS train-clean-100, train-clean-360, and train-other-500 subsets for 200k steps using the AdamW optimizer. Evaluation is conducted on the LibriTTS test-clean set.

J Ablation Experiments

Table 6: Ablation study on the auxiliary modules and granularity ordering. Gogo is evaluated on the LibriTTS test-clean set. GogoSpeech is evaluated on the Seed-TTS test-en set.

Model	Gogo							GogoSpeech	
	UT MOS	DNS MOS	STOI	PESQ WB	PESQ NB	SIM	WER	SIM	WER
Auxiliary Module									
Proposed Model	**4.19**	**3.99**	**0.92**	2.59	3.26	0.91	**6.35**	**0.667**	**2.394**
w/o ASR Module	3.97	3.91	0.87	**2.63**	**3.29**	**0.92**	8.66	0.640	4.372
w/o AR Prior	4.14	3.96	0.86	2.61	3.24	0.91	6.42	0.661	2.625
Granularity Ordering									
Proposed Model	**4.19**	**3.99**	0.92	**2.59**	**3.26**	**0.91**	**6.35**	**0.667**	**2.394**
w/o Asym. Masking	4.18	3.97	**0.94**	2.56	3.20	**0.91**	6.37	0.664	2.406
w/o Nested Dropout	4.10	3.89	0.89	2.39	3.16	0.89	6.41	0.657	3.969
w/o Loss Balancer	3.89	3.74	0.82	2.29	3.07	0.85	7.50	0.614	4.608

We perform an ablation study on all auxiliary modules and design choices in Gogo and further evaluate their impact on GogoSpeech. The results are summarized in Table 6. We can see that removing the ASR module leads to a slight improvement in PESQ and SIM for Gogo's reconstruction. However, the absence of ASR guidance significantly degrades the performance of GogoSpeech across all metrics. Eliminating the AR prior has little effect on Gogo's reconstruction metrics but results in a noticeable performance drop for GogoSpeech. This suggests that the AR prior primarily benefits the autoregressive modeling of GogoSpeech rather than signal reconstruction. Furthermore,

we ablate the granularity ordering mechanism. We can observe that removing asymmetric masking decreases all metrics except STOI and SIM for Gogo, and leads to a slight performance degradation for GogoSpeech. However, removing either nested dropout or the loss balancer causes a substantial decline in performance for both Gogo and GogoSpeech, highlighting their importance in learning informative coarse-to-fine token representations.

Table 7: Ablation study on the number of coarse tokens b used as the speech backbone on the Seed-TTS test-en set. Using all 10 tokens as the backbone degenerates into a single-stage model.

b	1	2	3	4	5	6	7	8	9	10
SIM	0.647	0.654	**0.667**	0.663	0.662	0.652	0.657	0.656	0.654	0.642
WER	2.754	2.531	2.394	**2.391**	2.397	2.441	2.346	2.451	2.460	3.121

We further perform an ablation study on the number of coarse tokens b used as the speech backbone, with results summarized in Table 7. We observe that increasing b from 1 to 3 yields substantial improvements in both SIM and WER, indicating that a slightly richer backbone provides more effective guidance for Stage II refinement. When b lies between 3 and 5, the performance of both metrics becomes relatively stable, with SIM reaching its peak at $b = 3$ and WER achieving its best value at $b = 4$. As b increases further from 6 to 9, all evaluation metrics exhibit a general downward trend, suggesting that overly detailed backbones may limit the capacity of Stage II to contribute fine-grained refinements. When $b = 10$, meaning that all tokens are generated entirely within Stage I of GogoSpeech, the performance drops to its lowest level across all metrics. These results provide strong empirical evidence for the effectiveness of our two-stage design, where a compact backbone combined with detail refinement yields the best overall performance.

K Visualization of Token Allocation

To better illustrate the behavior of the token allocator, we visualize the token allocation results in Figure 9 and Figure 10. Specifically, we present three aligned plots: (1) the original mel-spectrogram of the input speech, where vertical dashed lines indicate group boundaries; (2) the token budget assigned to each group by the allocator; and (3) the reconstructed mel-spectrogram obtained by Gogo using only the allocated tokens. The visualization clearly demonstrates that the allocator adaptively assigns more tokens to acoustically rich regions while reducing the allocation in silent or low-information segments, thereby achieving efficient yet high-quality reconstruction.

L Related Work

L.1 Neural Audio Codecs

Modern neural audio codecs are predominantly based on the VQ-GAN framework (Esser et al., 2021), which integrates an encoder, a vector quantizer, and a decoder into an end-to-end system trained with generative adversarial networks (GANs) (Goodfellow et al., 2020). Pioneering works such as SoundStream (Zeghidour et al., 2021) and EnCodec (Défossez et al., 2022) employ residual vector quantization (RVQ) together with carefully designed discriminators to improve perceptual fidelity. DAC (Kumar et al., 2023) further enhances codebook utilization by performing code lookup in a low-dimensional space using cosine similarity rather than Euclidean distance. To improve compatibility with SLMs, recent studies have explored injecting linguistic or semantic information into the quantization process. One line of work leverages self-supervised speech representations from models such as HuBERT (Hsu et al., 2021), WavLM (Chen et al., 2022), and w2v-BERT (Chung et al., 2021). For example, SpeechTokenizer (Zhang et al., 2024) distills semantic teacher representations into the first stage of RVQ, while Mimi (Défossez et al., 2024) transfers semantic information into a single-stage quantizer to decouple acoustic reconstruction from semantic coding. Another line of research integrates ASR-supervised features into codec training. Notably, $\mathcal{S}^3$ tokenizer (Du et al., 2024a) partitions the encoder of a pretrained SenseVoice ASR model (An et al., 2024) and inserts a quantization layer between its two halves. Despite recent advancements, the frame-wise quantization paradigm remains unchanged. Its inherent locality bias limits the codec's ability to learn high-level abstractions, which are essential for stable autoregressive modeling in SLMs.

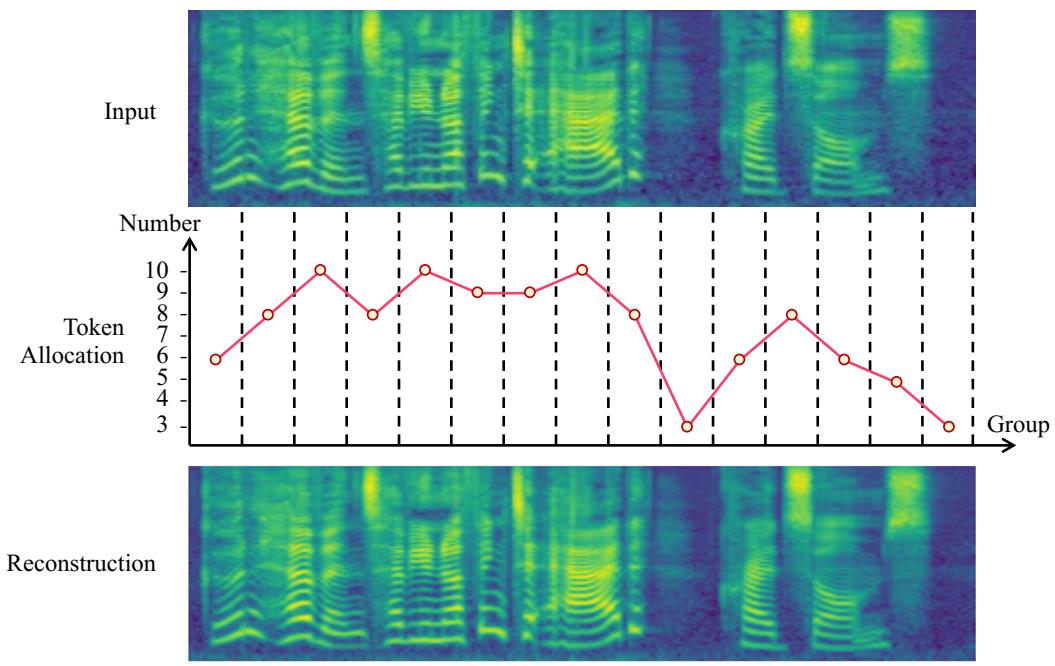

Figure 9: Visualization of sample 1580_141084_000085_000000 from the LibrtTTS test-clean set. The token allocator reduces the token rate from 47 Hz to 34.28 Hz.

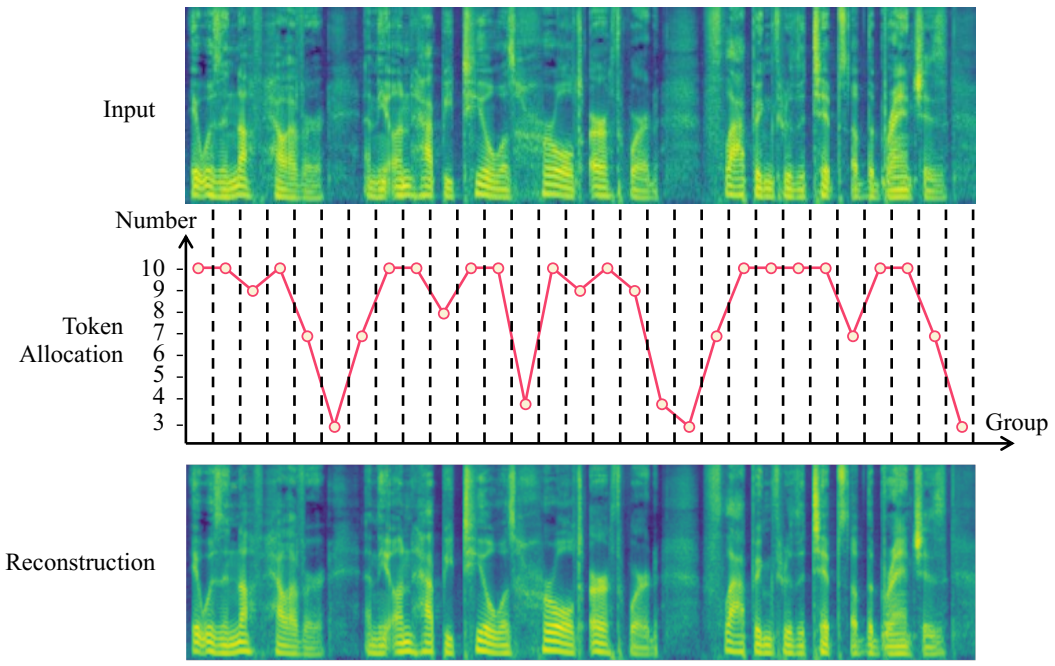

Figure 10: Visualization of sample 7176_88083_000011_000005 from the LibrtTTS test-clean set. The token allocator reduces the token rate from 47 Hz to 38.73 Hz.

L.2 Speech Language Models

Generative speech language modeling extends the LLM paradigm to the speech modality by modeling discrete speech tokens produced by neural audio codecs or clustering methods such as k-

means. Early works such as GSLM (Lakhotia et al., 2021) and SpeechGPT (Zhang et al., 2023) directly train language models on speech tokens, enabling the generation of natural-sounding speech. VALL-E (Wang et al., 2023) adopts an autoregressive model to generate the first token and a non-autoregressive model to predict residual tokens from EnCodec, conditioned on the linguistic content of text. To address the trade-off between high-fidelity reconstruction and effective autoregressive modeling, AudioLM (Borsos et al., 2023) introduces a hierarchical modeling framework that first generates semantic tokens and then refines them with acoustic tokens. In general, acoustic tokens are designed to encode speech at a low bitrate while preserving as much information as possible, whereas semantic tokens are learned from self-supervised speech models to capture phonetic or semantic representations that facilitate speech comprehension (Guo et al., 2025b). This hierarchical paradigm has inspired numerous follow-up works, including SPEAR-TTS (Kharitonov et al., 2023), AudioPaLM (Rubenstein et al., 2023), Moshi (Défossez et al., 2024), and TTS-Llama (Shen et al., 2025). While the hierarchical approach improves stability and controllability, the semantic modeling stage in existing methods typically operates at the same token rate as the acoustic modeling stage, which is substantially higher than the token rate used in the text modality.

L.3 Adaptive Bitrate in Neural Codecs

The non-uniform information density of speech signals makes constant-bitrate codecs inherently inefficient. SNAC (Siuzdak et al., 2024) extends RVQ to operate at multiple temporal resolutions, yet the bitrate remains fixed across different speech regions. Acoustic BPE (Shen et al., 2024) applies the byte-pair encoding (BPE) algorithm (Devlin et al., 2019) to speech tokens, reducing sequence length and increasing token correlation. More recently, VRVQ (Chae et al., 2025) introduces a variable-bitrate strategy into RVQ, allowing the number of quantizers per frame to be adaptively determined from a predicted importance map. Similarly, TFC (Zhang et al., 2025) dynamically allocates frame rates to different regions according to temporal entropy. However, these variable-bitrate codecs do not explicitly couple bitrate variation with reconstruction quality for joint optimization, and their effectiveness in generative tasks within the SLM framework remains underexplored.

M The Use of Large Language Models

Large language models were employed exclusively as auxiliary tools to edit and polish text written by the authors. Their usage was limited to improving clarity, grammar, and style of expression. No part of the research ideation, methodology, analysis, or results relied on LLMs.