FlexiSLM: A Spoken Language Model with Dynamic and Controllable Frame Rates

Jiaqi Li[1], Chaoren Wang[1], Xiaohai Tian[2], Mingjie Chen[1], Xinyu Liang[1], Xu Li[1], Yufan Lin[1], Junwen Qiu[1], Jun Zhang[2], Lu Lu[2], Zhizheng Wu[1]

[1]The Chinese University of Hong Kong, Shenzhen
[2]ByteDance

Abstract

Spoken language models (SLMs) extend LLMs to speech input and output, but existing systems use fixed frame rates (e.g., 25 or 12.5 Hz), overlooking speech's time-varying information density and limiting inference-time quality–speed tradeoffs. Recent dynamic-frame-rate audio tokenizers enable very low average frame rates and controllability, yet had not been applied to SLMs. We introduce FlexiSLM, the first SLM with dynamic, controllable frame rates, using pretrained FlexiCodec for dynamic speech output tokens. It integrates this representation into a multi-task speech-to-speech SLM, extends it with input-side frame compression, and adds direct frame-rate conditioning for accurate control during inference. FlexiSLM outperforms fixed-frame-rate 7B models, including Qwen2.5-Omni and Kimi-Audio, at 12.5 and 6.25 Hz; it can be steered down to 4.0 Hz, and at 6.25 Hz roughly halves inference time relative to 12.5 Hz while retaining strong speech-to-speech quality. Audio samples: https://flexislm.github.io; code and data: https://github.com/AmphionTeam/FlexiSLM.

1 Introduction

Spoken language models (SLMs) have emerged as a unified framework for speech understanding and generation, covering speech-to-speech dialogue, automatic speech recognition (ASR), text-to-speech (TTS), and audio understanding (Xu et al., 2025a; Ding et al., 2025; Zeng et al., 2024; Défossez et al., 2024). These models jointly model text and speech with a large language model (LLM) backbone, but typically represent speech at a fixed frame rate[1], e.g., 25 Hz for Qwen2.5-Omni (Xu et al., 2025a) and 12.5 Hz for Kimi-Audio (Ding et al., 2025). Fixed-rate tokenization ignores the time-varying information density of speech, wasting compute on silences and other information-sparse segments. It also prevents inference-time quality–speed control across devices, networks, and deployment budgets.

Model	FR (Hz)	FR Ctrl.	Dynamic FR
Qwen3-Omni-30B	12.5	✗	✗
Fun-Audio-Chat-8B	25(5.0)[†]	✗	✗
GLM 4-Voice-9B	12.5	✗	✗
Mimo-Audio-7B	25(6.25)[†]	✗	✗
Kimi-Audio-7B	12.5	✗	✗
Qwen2.5-Omni-7B	25 in / 50 out	✗	✗
FlexiSLM-7B	Any $\leq$12.5	✓	✓

Table 1: Capability comparison with representative spoken language models. "FR" denotes the frame rate of each system's input and output speech representations.

FlexiCodec (Li et al., 2025b) addresses these limitations with a dynamic-frame-rate codec that uses frame merging to achieve strong audio reconstruction quality at an average of 6.25 Hz, while allowing the average frame rate to be steered at inference time. Figure 1 illustrates this concept. However, FlexiCodec was validated only in a 0.3B-parameter TTS pipeline. Extending dynamic-rate coding to end-to-end SLMs is more challenging, but also more valuable: SLMs are more compute-intensive, and their broader capability set makes frame-rate controllability useful for heterogeneous deployment budgets.

Motivated by these benefits, we develop **F**lexible **S**poken **L**anguage **M**odel (FlexiSLM), the first spoken language model with dynamic and controllable frame rates. FlexiSLM is a thinker-talker speech-in, speech-out model: for output, we reuse FlexiCodec

Corresponding to: jiaqili3@link.cuhk.edu.cn; wuzhizheng@cuhk.edu.cn

[1]The frame rate is the number of discrete or continuous speech-encoding frames used to represent one second of audio; lower frame rates use fewer tokens.

[†]These systems use patching, yielding effective LLM-side frame rates of 5 Hz for Fun-Audio-Chat and 6.25 Hz for Mimo-Audio. This approach is complementary to our dynamic-frame-rate compression, and we leave the combination of both strategies to future work.

Proceedings of the 2026 Conference on Empirical Methods in Natural Language Processing, pages 11417–11440
October 24-29, 2026 ©2026 Association for Computational Linguistics

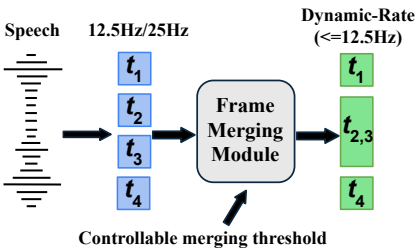

Figure 1: A high-level illustration of the dynamic-frame-rate strategy we use, following FlexiCodec. The frame merging module adaptively compresses speech based on information density.

as the Talker prediction target for dynamic-frame-rate tokens and reuse its pretrained flow-matching audio decoder for reconstruction; for input, we mirror FlexiCodec's dynamic-rate strategy by training a new frame merging module from scratch to dynamically compress input speech representations. Building on this dynamic-rate representation, we introduce a direct control method that lets the user specify the average output frame rate, allowing one FlexiSLM to operate at any frame rate ≤ 12.5 Hz without retraining. Our contributions are summarized as follows:

- **Dynamic frame-rate SLM framework and validation.** We introduce FlexiSLM, the first dynamic frame-rate SLM framework, with dynamic frame compression on both speech input and output. Although FlexiSLM leverages an existing dynamic-frame-rate modeling method, its primary contribution lies in its system-level integration and validation within a multi-task SLM. Prior work has mainly explored dynamic and controllable frame rates in the context of speech coding or TTS; our SLM validation extends this approach to broader applications, including speech QA and audio understanding. Experiments confirm strong performance at 12.5 Hz and 6.25 Hz, with graceful degradation at 5.0 Hz and 4.0 Hz.
- **Accurate and practical frame-rate control.** The broad capability set of an SLM makes frame-rate controllability highly beneficial for accommodating various computational budgets across different tasks. We propose direct frame-rate conditioning, a novel frame-rate control method that allows users to specify the average output frame rate instead of indirectly tuning a merging threshold. This makes FlexiSLM, to our knowledge, the first SLM with frame-rate controllability.

We have open-sourced training and inference code[2], along with reproduced training data and model checkpoints.

2 Related Work

Speech Tokenization. Speech tokenization converts continuous audio into discrete tokens suitable for speech language modeling. Early neural audio codecs such as SoundStream (Zeghidour et al., 2021) and EnCodec (Défossez et al., 2022) use residual vector quantization (RVQ) to produce acoustic tokens at fixed frame rates (e.g., 50 Hz or 75 Hz), prioritizing reconstruction fidelity. Semantic tokens derived from self-supervised models like HuBERT (Hsu et al., 2021) capture linguistic content and are increasingly used in speech language modeling (Borsos et al., 2023; Li et al., 2024; Du et al., 2024a; Ding et al., 2025). Recent work has pushed toward more efficient representations while maintaining high audio quality: single-codebook approaches (50–75 Hz) (Wu et al., 2024; Ji et al., 2024) and semantic-enhanced codecs (12.5–50 Hz) (Li et al., 2025a; Zhang et al., 2023).

Recent work has explored dynamic frame rates, leveraging the temporal sparsity of speech so that lower average frame rates reduce the computational cost of speech language models. FlexiCodec (Li et al., 2025b), the tokenizer used in this work, merges 12.5 Hz semantic features based on similarity to achieve dynamic-rate tokenization at an average of 6.25 Hz. The authors of FlexiCodec also demonstrate controllable frame-rate tokenization and TTS. Other dynamic-frame-rate works, including CodecSlime (Wang et al., 2025), TFC (Zhang et al., 2025b), and VARSTok (Zheng et al., 2025), explore higher average frame rates from 18.75 Hz to 40 Hz. However, dynamic-rate codecs have not been applied within a spoken language model framework, where low frame rates and frame-rate controllability offer larger practical benefits.

Spoken Language Models. Spoken language models (SLMs) are general-purpose speech processing systems (Arora et al., 2025). Analogous to text LLMs, they can follow natural-language instructions across diverse speech tasks. Recent end-to-end SLMs extend text-based LLMs to directly comprehend and generate speech. A dominant design follows a three-stage paradigm: a pretrained speech encoder extracts acoustic features,

[2] https://github.com/AmphionTeam/FlexiSLM

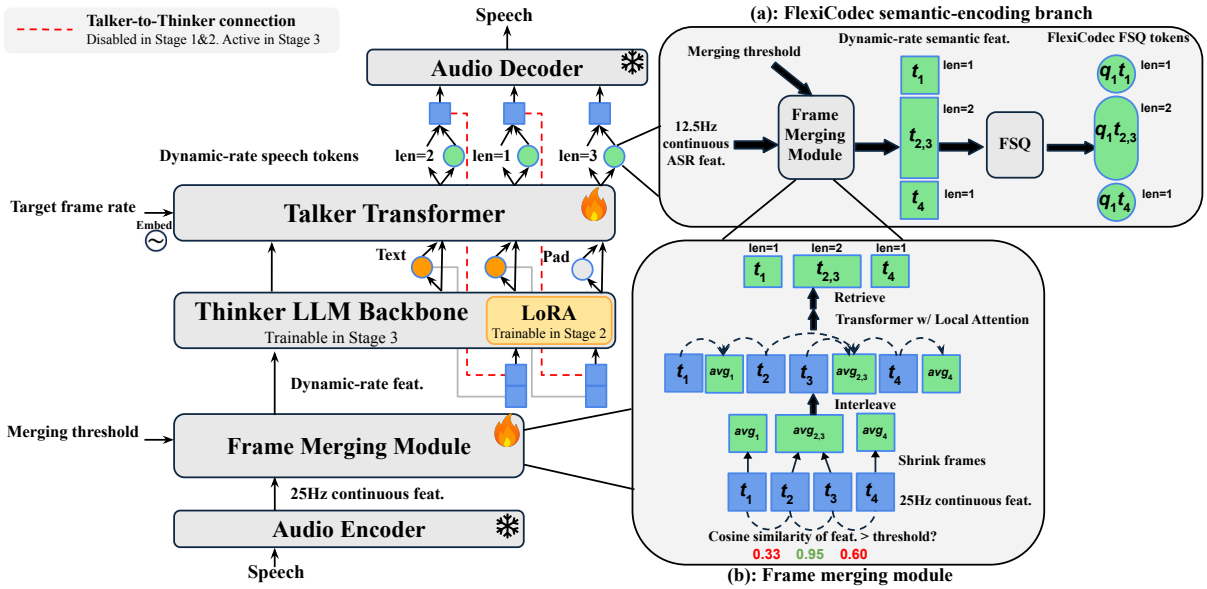

Figure 2: Overall architecture of FlexiSLM. Input speech is encoded into continuous features, compressed by the input-side Frame Merging Module into dynamic-rate representations, and processed by the LLM Backbone. The Talker Transformer then predicts FlexiCodec semantic FSQ tokens and their frame lengths conditioned on the target frame rate, enabling controllable dynamic-rate speech generation. The input-side Frame Merging Module shares FlexiCodec's implementation but uses separate parameters. We use public pretrained weights for the FlexiCodec and flow matching audio decoder.

which condition a decoder-only LLM, followed by an additional transformer module or prediction head that predicts speech tokens (Wang et al., 2026; Xu et al., 2025a; Ding et al., 2025). Kimi-Audio (Ding et al., 2025) models parallel speech-text at 12.5 Hz using a separate LM head for speech tokens. Qwen2.5-Omni (Xu et al., 2025a) adopts a thinker–talker architecture operating at 25 Hz, whose talker module predicts speech tokens. Active research areas include full-duplex capability (e.g., Moshi (Défossez et al., 2024)), interleaved speech-text sequences (e.g., GLM-4-Voice (Zeng et al., 2024)), low-frame-rate audio tokenization for SLMs (e.g., Mimi (Défossez et al., 2024)), and dual-resolution speech representations (e.g., Fun-Audio-Chat/DrVoice (Team et al., 2025; Tan et al., 2025) and Mimo-Audio (Zhang et al., 2025a) group speech into 5 Hz or 6.25 Hz sequences).

3 Method

3.1 Architecture Overview

Figure 2 illustrates the overall architecture of FlexiSLM, a parallel speech-text model with the following components:

Audio Encoder. This module encodes the user's speech into a semantically rich continuous representation suitable for LLM understanding. We adopt the pretrained Qwen2.5-Omni audio encoder (Xu et al., 2025a), which extracts 25 Hz continuous speech features from waveforms.

Frame Merging Module. This module compresses the number of frames in a sequence. It appears twice in our model: (1) on the input side, it reduces the 25 Hz continuous features from the Audio Encoder to a dynamic-rate sequence $\leq$12.5 Hz; (2) inside the pretrained FlexiCodec audio tokenizer, it merges 12.5 Hz ASR features before quantization. The input-side module contains newly initialized parameters and is trained end-to-end within FlexiSLM, while the output-side module is frozen. Both instances share the same merging mechanism. We describe this module in Section 3.2.

LLM Backbone (Thinker). We use Qwen2.5-7B-Instruct (Yang et al., 2024) as initialization. This LLM has also been used in Qwen2.5-Omni and Kimi-Audio to initialize their backbones.

FlexiCodec Audio Tokenizer. We use the open-source pretrained FlexiCodec to obtain discrete speech tokens as the prediction target of FlexiSLM's Talker module. As illustrated in Figure 2(a), the codec discretizes each frame with Finite Scalar Quantization (FSQ (Mentzer et al., 2023)); each token is paired with a frame length attribute for audio reconstruction. We use FlexiCodec's semantic tokens and omit its RVQ acoustic

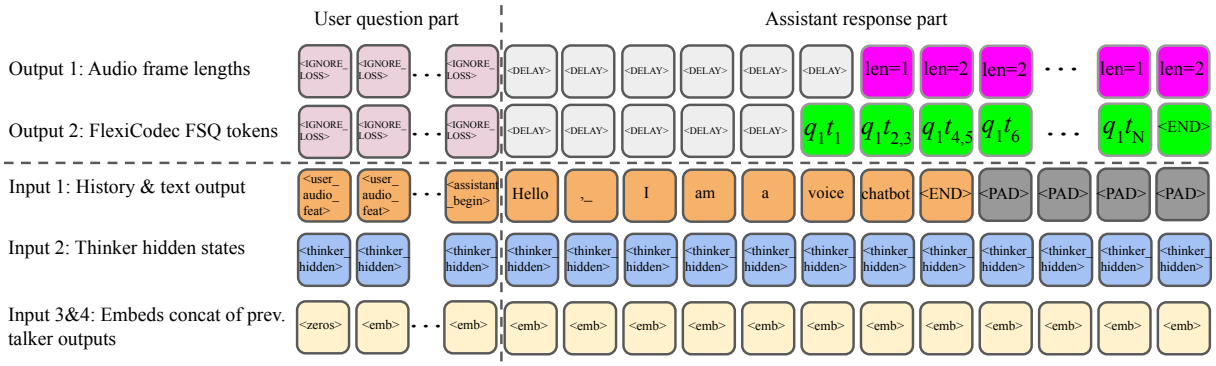

Figure 3: Talker Transformer input-output structure. At each position, the Talker jointly predicts two parallel streams: FlexiCodec FSQ token codes and per-token frame lengths for dynamic-rate decoding. Delays are applied to provide lookahead; tokens in the user's turn and the text-generation delay region are masked from the loss.

tokens. Appendix J provides more details.

Talker Transformer. The Talker decodes the Thinker LLM's hidden states and outputs into FlexiCodec's dynamic-frame-rate speech tokens.

- **Input:** As shown in Figure 3, its input sequence runs over the entire user + assistant context. At each position, the Talker input embedding is projected from the concatenation of (1) the backbone LLM's last-layer hidden state, (2) a sinusoidal embedding of the target frame rate (Section 3.3), and (3) the embeddings of the previously emitted speech and frame length tokens.
- **Output:** The Talker produces two parallel output streams: FlexiCodec FSQ codes and their associated frame lengths, enabling dynamic-rate output. The Talker uses two output LM heads to predict the streams in parallel.
- **Token delay:** As shown in Figure 3, the Talker's FSQ token stream is delayed by 5 tokens from the text stream. This provides a small lookahead that prevents speech from preceding its corresponding text (Du et al., 2024b). The frame length tokens are delayed by one additional position, allowing the model to predict a frame's duration after knowing its corresponding speech token (Li et al., 2025b).

Audio Decoder. The audio decoder is a frozen non-autoregressive (NAR) flow-matching (Lipman et al., 2022) Transformer that decodes mel spectrograms from the speech tokens. A Vocos (Siuzdak, 2023) neural vocoder then converts the mel spectrogram into 24 kHz speech. We use the pretrained flow-matching model and vocoder from the open-source FlexiCodec repository. We provide additional details in Appendix I.

Talker-to-Thinker Connection. In addition to the standard cascaded Thinker-to-Talker information flow (Xu et al., 2025a), FlexiSLM contains an optional Talker-to-Thinker connection (Tan et al., 2025) (the red dashed arrow in Figure 2) that feeds the Talker's previously emitted speech-token embeddings back into the Thinker LLM Backbone at the next step. This gives the Thinker LLM explicit access to what has already been spoken. The connection projects the concatenation of the Talker's embeddings (speech code + frame length) and the text embedding into the Thinker LLM's hidden state. We can disable this connection by zeroing out the contribution of the Talker's embeddings in the concatenation.

3.2 Frame Merging Module

As shown in Figure 2(b), the Frame Merging Module adaptively compresses a fixed-rate semantic feature sequence by merging adjacent frames that carry redundant information. Given a sequence of feature vectors $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T$ at a base frame rate, we compute the cosine similarity between consecutive frames: $s_t = \frac{\mathbf{x}_t \cdot \mathbf{x}_{t+1}}{\|\mathbf{x}_t\| \, \|\mathbf{x}_{t+1}\|}, \quad t = 1, \ldots, T-1$. If s_t exceeds a merging threshold τ, frames $\mathbf{x}_t$ and $\mathbf{x}_{t+1}$ are grouped and their average is computed. This process runs greedily from left to right, with contiguous high-similarity frames merged into a single averaged representation.

After merging, each group produces an averaged feature $\bar{\mathbf{x}}_k$ and a frame length attribute l_k denoting the number of original frames in the group. We interleave the original and averaged features to form an augmented sequence, which is processed by a lightweight Transformer with local attention. Finally, we retrieve the representations at positions corresponding to the averaged features, yielding the merged sequence with associated frame lengths.

3.3 Controllable Frame Rate

A key feature of FlexiSLM is its ability to control the output frame rate at inference time, enabling a single deployed model to operate across a range of compute budgets without retraining. For a dynamic-frame-rate sequence, the average frame rate is defined as $\frac{\text{Total number of frames after merging}}{\text{Audio duration in seconds}}$. We first describe a baseline threshold-based strategy and its limitations, then introduce our proposed direct frame-rate control. We focus on output frame-rate control; input frame rate is controlled by computing the target number of merged frames from the desired rate and selecting a merging threshold per utterance to match it.

Merging Threshold Control (Baseline). A straightforward approach is to control the merging threshold τ: a higher τ merges fewer frames (higher frame rate), while a lower τ merges more (lower frame rate). This approach is used in FlexiCodec-TTS (Li et al., 2025b). However, it provides only *indirect* control, with several limitations: (1) the resulting frame rate varies significantly across utterances and datasets, making the speedup difficult to predict (Table 4); (2) it is a one-to-many mapping: a single threshold maps to a wide distribution of average frame rates, increasing modeling complexity; and (3) it is unintuitive for users unfamiliar with the model architecture.

Direct Frame-Rate Control. To overcome these limitations, we directly condition the Talker Transformer and the Frame Merging Module on the target average frame rate. During training, we randomly sample merging thresholds, compute the resulting average frame rate for each utterance, and feed this empirical rate as a conditioning signal. At inference, the user simply specifies the desired frame rate.

To enable continuous control over a range of frame rates, we encode the scalar frame rate r using sinusoidal positional encoding (Vaswani et al., 2017): $\text{PE}(r) = [\sin(r\omega_1), \cos(r\omega_1), \ldots, \sin(r\omega_d), \cos(r\omega_d)]$, where the ω_i are frequency bases derived from 10,000. This encoding serves as one of the inputs to the Talker. Each position in the Talker input sequence receives the same frame-rate condition.

3.4 Training Procedure

Training proceeds in three stages:

Dataset	Task	Ratio	Utts.	Hours (h)
FlexiSLM-Data	Dialog-s2s	3.0	1.4M	9.9K
TriviaQA+Web Q.	Dialog-s2s	3.0	140K	0.4K
TriviaQA+Web Q.	Dialog-t2t	1.0	140K	–
Emilia-EN	TTS	0.15	14M	50K
MLS	TTS	0.15	12M	50K
LibriSpeech	ASR	1.0	280K	1K
MLS	ASR	0.1	12M	50K
LLaSO-instruct	Audio Und.	1.0	7M	24K

Table 2: Data used for Stage 2. "Ratio" denotes the sampling ratio for one training epoch. A complete list of references appears in Appendix A. "Hours" and "Utterances" are measured before sampling.

Stage 1: Talker Pre-training. In this stage, we freeze the LLM backbone and train only the randomly initialized Talker. We train FlexiSLM end to end with TTS data. This stage uses English samples from Emilia (He et al., 2024) and Multilingual LibriSpeech (MLS) (Pratap et al., 2020), totaling about 100K training hours. The Talker-to-Thinker connection is disabled in Stages 1 and 2.

Stage 2: Multi-Task LoRA Fine-Tuning. In this stage, we activate the speech input-side Frame Merging Module, Thinker, and Talker. LoRA (Hu et al., 2021) is applied to the Thinker. The Frame Merging Module is randomly initialized. We train on a mixture of speech processing tasks. Table 2 shows the distribution of the data sources in Stage 2. A central component is **FlexiSLM-Data**, a speech-to-speech dialogue corpus constructed for this work. FlexiSLM-Data is distilled from a 30B Qwen3-Omni (Xu et al., 2025b) SLM, with data sourced from a wide range of datasets. We detail our data construction process in Appendix A.

Stage 3: Full Fine-Tuning. This stage further unlocks the backbone's capacity and improves speech perception. It continues from the Stage 2 checkpoint and uses the same data, with two differences. First, we perform full-parameter training after merging the LoRA updates back into the target LLM weight matrices. Second, we enable and train the Talker-to-Thinker connection.

3.5 Training Loss and Configuration

Our model is trained end to end with a weighted cross-entropy loss over multiple token streams. The total loss is: $\mathcal{L} = \lambda_{\text{text}}\mathcal{L}_{\text{text}} + \lambda_{\text{speech}}\mathcal{L}_{\text{speech}} + \lambda_{\text{speech_len}}\mathcal{L}_{\text{speech_len}}$, where $\mathcal{L}_{\text{text}}$, $\mathcal{L}_{\text{speech}}$, and $\mathcal{L}_{\text{speech_len}}$ are the cross-entropy losses for text tokens, FlexiCodec speech codes, and per-token speech frame lengths, respectively. We set $\lambda_{\text{text}} =$

Model	In FR	Out FR	OpenAudioBench ACC↑ Llama Q s2t/s2s	Web Q s2t/s2s	Trivia s2t/s2s	VoiceBench Score/ACC↑ Alpaca s2t/s2s	Common s2t/s2s	IFEval s2t/s2s	SD-QA s2t/s2s	AdvB. s2t/s2s	Overall s2t/s2s	ASR WER↓ LibriSpeech clean	other
>7B Models													
Gemini 2.5-Pro	-	-	**87**/-	**79**/-	**93**/-	3.73/-	3.08/-	**4.49**/-	**90**/-	4.69/-	**83.6**/-	3.05	5.19
Gemini 2.5-Flash	-	-	-/**78**	-/**67**	-/**78**	-/3.30	-/**3.58**	-/2.31	-/**74**	-/4.82	-/**72.2**	-	-
Qwen3-Omni-30B-A3B	12.5	12.5	82/72	64/63	70/67	3.95/3.60	3.65/**3.58**	3.86/**2.60**	68/62	4.98/**4.89**	76.5/69.7	**1.44**	**2.82**
GLM-4-Voice-9B	12.5	12.5	76/62	50/48	51/47	3.24/2.82	2.95/2.69	2.32/1.97	48/45	4.74/4.49	61.1/55.2	2.82	7.66
Fun-Audio-Chat-8B	5.0	5.0	80/74	61/60	63/59	**4.20/3.77**	**3.67**/3.46	3.81/2.55	59/54	**4.99**/4.81	74.5/67.3	-	-
7B Models													
Qwen2-Audio-7B	25	-	68/-	44/-	39/-	2.87/-	2.82/-	2.25/-	31/-	4.89/-	54.9/-	1.74	4.04
Mimo-Audio-7B	6.25	6.25	79/71	61/55	57/48	3.83/2.90	**3.68**/3.06	3.24/2.12	54/47	4.97/4.43	70.6/59.0	-	-
Kimi-Audio-7B	12.5	12.5	79/69	**66**/56	**62**/54	3.38/2.74	3.00/2.46	**3.26**/2.11	**58**/37	4.98/4.73	69.7/57.2	**1.80**	**2.45**
Qwen2.5-Omni-7B	25	50	76/72	60/59	58/57	3.30/3.07	3.35/3.24	2.68/2.12	54/52	4.92/4.88	66.7/63.3	2.38	4.21
FlexiSLM-7B-Stage2	12.5	12.5	79/**75**	61/59	61/59	3.74/3.43	3.39/3.35	3.18/2.24	58/57	4.99/**4.93**	70.6/66.1	2.26	5.69
		6.25	79/70	61/59	61/57	3.74/3.33	3.39/3.29	3.18/2.14	58/54	4.99/4.86	70.6/64.1	2.26	5.69
	—	12.5	78/74	58/58	59/58	3.67/3.29	3.34/3.29	3.10/2.21	57/57	4.98/4.90	69.2/65.1	2.68	6.70
	6.25	6.25	78/71	58/56	59/56	3.67/3.23	3.34/3.25	3.10/2.12	57/55	4.98/4.92	69.2/63.6	2.68	6.70
FlexiSLM-7B-Stage3	12.5	12.5	**80/75**	61/**60**	**62/60**	**3.84/3.50**	3.63/**3.48**	3.22/**2.33**	62/**60**	**5.00**/4.83	**72.4/67.2**	1.98	5.79
		6.25	**80**/74	61/59	**62**/58	3.82/3.44	3.59/3.42	3.22/2.26	**63**/59	**5.00**/4.87	72.3/66.2	1.98	5.79
	6.25	12.5	79/73	58/56	60/58	3.70/3.38	3.49/3.32	3.17/2.28	61/59	4.97/4.85	70.6/65.3	2.55	6.37
		6.25	78/71	58/55	60/57	3.70/3.31	3.43/3.3	3.11/2.21	61/58	4.96/4.85	70.2/64.3	2.55	6.37
	5.0	5.0	77/67	55/52	58/52	3.70/3.05	3.35/3.04	3.14/2.09	59/53	4.97/4.79	69.0/60.4	3.34	7.85
	4.0	4.0	75/64	53/48	56/48	3.60/2.84	3.28/2.83	2.94/1.85	58/48	4.97/4.67	67.2/56.5	4.47	9.53

Table 3: SLM evaluation results on Kimi-Audio-Evalkit. OpenAudioBench and VoiceBench results are judged by GPT-5.5. "Overall" averages OpenAudioBench and VoiceBench scores normalized to a 0-100 scale.

2 and $\lambda_{\text{speech}} = \lambda_{\text{speech_len}} = 1$. For non-speech sequences, $\mathcal{L}_{\text{speech}}$ and $\mathcal{L}_{\text{speech_len}}$ are set to 0.

We train FlexiSLM on 24 A100 80G GPUs. During training, we randomly sample an input frame rate target from $\mathcal{U}(4, 12.5)$ Hz and a FlexiCodec merging threshold $\tau \sim \mathcal{U}(0.85, 1.0)$ to support multiple frame rates. The Talker has 630M parameters; the Audio Encoder has 640M parameters; each Merging Transformer has 20M parameters. We discuss more training details in Appendix H.

4 Experiments

4.1 Experimental Setup

Baselines. Our primary comparisons are 7B SLMs: Qwen2.5-Omni (Xu et al., 2025a), Kimi-Audio (Ding et al., 2025), Mimo-Audio (Zhang et al., 2025a), and Qwen2-Audio (Chu et al., 2024). Other models include Fun-Audio-Chat (Team et al., 2025), GLM4-Voice (Zeng et al., 2024), Gemini 2.5-Pro and Gemini 2.5-Flash-Native-Audio-Preview (Comanici et al., 2025).

Evaluation. We evaluate with Kimi-Audio-Evalkit (Ding et al., 2025). We use the GPT-5.5 API for LLM-as-judge evaluations. Benchmarks include: (1) **OpenAudioBench** (Li et al., 2025c) for speech QA, including Llama Questions (Nachmani et al., 2024), Web Questions (Berant et al., 2013), and TriviaQA (Joshi et al., 2017); (2) **VoiceBench** (Chen et al., 2024) for speech QA, including AlpacaEval (Li et al., 2023), CommonEval (Ardila et al., 2020), IFEval (Zhou et al., 2023), SD-QA (Faisal et al., 2021), and AdvBench (Zou et al., 2023); (3) **LibriSpeech** (Panayotov et al., 2015) test set for ASR. All benchmarks are evaluated in both speech-to-text (s2t) and speech-to-speech (s2s) modes where applicable. Additional results on audio-understanding tasks and TTS quality are reported in Appendices G and F, where TTS results are reported with the E2TTS test suite (Eskimez et al., 2024).

4.2 Main Evaluation Results

Table 3 presents the main evaluation results on Kimi-Audio-Eval Kit. Our analysis is as follows.

• **FlexiSLM is strongest at** 12.5 **Hz and remains competitive at** 6.25 **Hz.** At the $12.5/12.5$ Hz setting, FlexiSLM-Stage3 attains overall s2t/s2s scores of $72.4/67.2$, surpassing the strongest 7B baseline Qwen2.5-Omni-7B ($66.7/63.3$) by $5.7/3.9$ points, and exceeding Kimi-Audio-7B ($69.7/57.2$) and Mimo-Audio-7B ($70.6/59.0$). When only the output frame rate is reduced to 6.25 Hz, FlexiSLM keeps nearly the same scores

Method	Target	Llama Q	Web Q	TriviaQA	Alpaca
Merging Threshold (τ) Ctrl.	τ=0.90	8.34 (3.91~10.61) σ=0.70	7.91 (4.72~10.74) σ=0.66	8.18 (4.73~10.37) σ=0.78	8.08 (6.78~10.19) σ=0.40
	τ=0.86	6.44 (4.40~8.42) σ=0.59	6.03 (3.44~8.82) σ=0.59	6.32 (3.43~8.86) σ=0.65	6.06 (4.08~8.05) σ=0.45
Direct FR Ctrl.	6.25 Hz	6.25 (6.05~6.73) σ=0.05	6.25 (6.03~6.77) σ=0.04	6.24 (5.77~7.03) σ=0.06	6.24 (5.95~6.42) σ=0.03
	4.0 Hz	3.99 (3.84~4.24) σ=0.05	4.00 (3.80~4.58) σ=0.04	4.00 (3.57~4.44) σ=0.05	4.00 (3.89~4.09) σ=0.03

Table 4: Audio output frame-rate controllability across different SLM benchmarks. Each block shows the mean achieved output frame rate (Hz) with the per-utterance range and standard deviation σ on the next two lines.

(72.3/66.2), while using half as many output speech tokens. Even when both input and output are reduced to 6.25 Hz, FlexiSLM reaches 70.2/64.3, still above all 7B baselines on s2s and competitive on s2t. This validates our central claim that a dynamic-rate SLM can match or surpass fixed-rate SLMs while using lower frame rate representations.

• **FlexiSLM can be controlled at very low frame rates, but quality leaves room for improvement.** Beyond the 6.25 Hz setting, FlexiSLM can also be steered to more aggressive 5.0 and 4.0 Hz operating points. However, quality drops more noticeably in this regime: moving from 6.25/6.25 to 5.0/5.0 Hz reduces overall s2t/s2s from 70.2/64.3 to 69.0/60.4, and reducing further to 4.0/4.0 Hz gives 67.2/56.5. ASR follows the same trend, with LibriSpeech clean/other WER worsening from 2.55/6.37 at 6.25 Hz to 3.34/7.85 at 5.0 Hz and 4.47/9.53 at 4.0 Hz. These results show that frame-rate control extends to very low rates, while improving robustness below 6.25 Hz remains an important direction for future work.

• **Stage 3 improves full-rate quality and low-rate robustness.** Stage 3 improves the 12.5/12.5 Hz score from 70.6/66.1 to 72.4/67.2 over Stage 2. It also reduces the s2s loss caused by halving the output frame rate (12.5/12.5 $\rightarrow$ 12.5/6.25) from 2.0 to 1.0 points and improves the matched 6.25/6.25 Hz setting from 69.2/63.6 to 70.2/64.3. These gains suggest that full-parameter fine-tuning with the Talker-to-Thinker connection improves both quality and robustness.

4.3 Frame Rate Controllability

4.3.1 Frame Rate Control Accuracy

We evaluate how accurately FlexiSLM can control the average output frame rate. We compare two variants that differ only in their control mechanism: merging-threshold control and direct frame-rate control. Table 4 reports the resulting average output frame rates across benchmark datasets. We focus on the output side because the input frame rate is controlled deterministically and therefore does not require a separate analysis.

• **Direct frame-rate control is more accurate than threshold control.** Merging threshold control exhibits high per-utterance variance: for example, targeting roughly 8 Hz via τ=0.90 yields a wide range of frame rates (3.91~10.74 Hz, $\sigma\approx$0.70) and varying averages across datasets (7.91~8.34 Hz). In contrast, direct frame-rate control, used in our final FlexiSLM, closely matches the requested average rates, with errors below 0.1 Hz across all settings.

• **When might merging threshold control still be preferable?** One could argue that a dynamic frame-rate system should adapt to utterance-level redundancy, and that the higher variance of threshold control reflects this adaptivity. We view the two strategies as complementary. Threshold control fixes the *compression intensity* applied to local frame pairs, allowing the resulting frame rate to vary with the redundancy of each utterance. Direct frame-rate control instead fixes the *average frame rate* of each utterance and adjusts the effective compression intensity accordingly. Thus, threshold control may be preferable when the goal is to apply a consistent compression criterion across utterances.

For FlexiSLM, we adopt direct frame-rate control for two reasons. First, it reduces the modeling ambiguity faced by the Talker. As discussed in Section 3.3, a single merging threshold can correspond to a wide distribution of output frame rates, making the threshold condition an indirect and ambiguous generation target. Second, predictable compute budgets are important for deployment. In practical SLM serving, committing to a target frame rate, and therefore to an approximate real-time factor, is useful.

4.3.2 Frame Rate and Inference Efficiency

Table 5 reports full-response real-time factor (RTF), and Table 6 complements this metric with first-1 s-chunk latency and a component-wise latency

Model	In	Out	RTF ↓	Avg Out Dur. (s)	Avg Out Seq Len	TFLOPs↓
Qwen2.5-Omni-7B	25	50	1.57 (1.3×)	16.5	835 (talker) 61 (thinker)	5.26 (1.2×)
FlexiSLM-7B	12.5	12.5	1.17 (1.0×)	15.2	190	4.57 (1.0×)
		6.25	0.59 (0.5×)	18.0	114	3.41 (0.7×)
	6.25	12.5	1.09 (0.9×)	15.7	196	4.18 (0.9×)
		6.25	**0.57** (**0.5**×)	16.2	101	**2.73** (**0.6**×)

Table 5: Inference efficiency of FlexiSLM under different input/output frame rate configurations, compared with Qwen2.5-Omni-7B.

Model	In	Out	Overall	Percentage Breakdown			
			Latency (First 1 s Chunk) ↓	Audio Enc. & Merge	Thinker	Talker	Audio Decoder
FlexiSLM-7B	12.5	12.5	1.62 s (1.0×)	5%	53%	22%	19%
		6.25	1.21 s (0.75×)	6%	48%	20%	26%
	6.25	12.5	1.60 s (0.99×)	6%	53%	22%	18%
		6.25	1.17 s (0.72×)	8%	47%	20%	26%

Table 6: Latency evaluation of FlexiSLM.

breakdown. All measurements are conducted on a single A100 GPU with batch size 1, averaged over 30 runs on an OpenAudioBench subset, using the same audio prompts for all FlexiSLM configurations (average prompt duration: 7.7 s). We also report an approximate model-side compute estimate based on each module's sequence length and parameter count. This estimate excludes the non-autoregressive flow-matching audio decoder, whose parallel decoding cost is not directly reflected in the measured RTF. For chunk-based inference, we modify FlexiSLM to emit decoded audio immediately after the tokens for each 1 s audio chunk are predicted.

• **Output frame rate is the main driver of speed and latency improvements.** Reducing the output frame rate from 12.5 to 6.25 Hz nearly halves RTF (1.17 → 0.59 with 12.5 Hz input; 1.09 → 0.57 with 6.25 Hz input) and reduces first-1 s-chunk latency from 1.62 s to 1.21 s (0.75×) or from 1.60 s to 1.17 s (0.72×). In contrast, reducing only the input frame rate brings much smaller gains, indicating that output-side autoregressive speech generation dominates the efficiency benefit. The component-wise latency breakdown in Table 6 shows that lowering the output frame rate from 12.5 to 6.25 Hz reduces the latency shares of both the Thinker and Talker. This is expected: completing the first one-second chunk requires fewer output tokens, which reduces the number of Thinker and Talker forward passes. The first-chunk latency reduction (0.72–0.75×), however, is smaller than the full-response RTF reduction (about 0.5×), likely because fixed overheads and audio-decoder latency account for a larger fraction of first-chunk generation, whereas full-response generation is dominated more strongly by repeated Thinker–Talker autoregressive computation.

• **FlexiSLM offers controllable deployment tradeoffs, covering a wide range of operating points.** By varying the input and output frame rates, the same model covers operating points from 1.17 to 0.57 RTF, and first-chunk latencies from 1.62 s down to 1.17 s, without retraining. This gives FlexiSLM a practical speed-quality tradeoff control that fixed-rate SLMs do not provide. These first-chunk latency reductions further highlight the deployment value of output frame-rate control, since first-chunk latency is closely tied to perceived responsiveness in conversational systems.

• **Comparison with Qwen2.5-Omni-7B in RTF.** At the 12.5/12.5 Hz setting, FlexiSLM is 1.3× faster than Qwen2.5-Omni-7B (1.17 vs. 1.57 RTF). When the FlexiSLM output frame rate is reduced to 6.25 Hz, the speedup increases to 2.7×. The TFLOPs estimates show a similar trend: FlexiSLM at 12.5/12.5 Hz requires less estimated model-side compute than Qwen2.5-Omni-7B (4.57 vs. 5.26), and the 6.25/6.25 Hz setting further reduces this cost to 2.73 TFLOPs. This gain is smaller than the raw output-rate ratio between Qwen2.5-Omni and FlexiSLM (50 vs. 6.25 Hz) because the two systems use different inference structures: FlexiSLM's Thinker processes a speech-length sequence to maintain bidirectional Thinker–Talker information flow, whereas Qwen2.5-Omni's Thinker stops after text generation. FlexiSLM's design trades part of the rate-reduction speedup for stronger speech-aware response generation and is also aligned with future full-duplex and interactive SLM settings.

4.4 Ablation Study

We ablate two core design choices of FlexiSLM: dynamic frame-rate representations and direct frame-rate controllability. Results are reported in Table 7. To reduce training cost, all ablation models are trained with a smaller budget of 8 GPUs for 160K steps, up to the end of Stage 2. All variants are evaluated at 6.25 Hz input and 6.25 Hz output. We include additional ablations of other architectural decisions in Appendix B.

• **Dynamic output frame rates improve generation quality at 6.25 Hz.** To isolate the effect of dynamic output tokens, we keep the same speech tokenizer but replace threshold-based frame merging

Method	OpenAudioBench& VoiceBench AVG ↑	ASR WER ↓	TTS WER ↓
	s2t / s2s	clean / other	
FlexiSLM-Stage 2 (retrained)	**68.7 / 63.0**	**2.92 / 7.20**	**3.11**
w/o dynamic output FR (use uniform merging)	67.7 / 61.0	3.14 / 7.67	4.95
w/o dynamic input FR (use uniform merging)	67.5 / 62.9	2.97 / 7.97	3.12
w/ threshold-controlled output FR	68.2 / 61.7	2.96 / 7.24	3.53

Table 7: Ablation study of FlexiSLM's core design choices. This study uses 8 GPUs for each model's training. We underline metrics that are noticeably worse than the baseline.

with uniform merging, where every two adjacent frames are merged to convert 12.5 Hz sequences into 6.25 Hz sequences. This causes a clear drop in speech-to-speech performance, with s2s Overall AVG decreasing from 63.0 to 61.0 and LibriSpeech TTS WER increasing by a relative 59% (3.11 → 4.95). The s2t Overall AVG also drops from 68.7 to 67.7, suggesting that lower-quality output tokenization can hurt the LLM backbone during end-to-end training.

• **Dynamic input frame rates help speech understanding.** To evaluate dynamic frame rates on the input side, we retrain FlexiSLM with uniform input merging and evaluate it at 6.25 Hz. Replacing dynamic input merging with uniform merging reduces s2t Overall AVG from 68.7 to 67.5 and worsens ASR test-other WER from 7.20 to 7.97, while s2s Overall AVG and ASR test-clean remain nearly unchanged. This suggests that dynamic input merging is most beneficial for tasks that require finer-grained speech understanding, such as instruction following and harder ASR cases.

• **Direct frame-rate control improves Talker.** Replacing direct frame-rate conditioning with merging-threshold control degrades s2s and TTS performance, while leaving s2t and ASR nearly unchanged. We attribute this to the ambiguity of threshold control discussed in Section 3.3. This less predictable target likely slows Talker convergence under our limited training budget, leading to weaker generation quality.

5 Additional Materials

We provide supplementary details and analyses in the appendix. Key takeaways include:

- **Data details.** Appendix A describes the construction of FlexiSLM-Data. Appendix D deduplicates training and evaluation prompts.
- **Additional ablations and a human listening test.** Appendix B studies encoder, backbone, and other architectural choices. Appendix C reports a human listening test. These studies reveal room for audio-quality improvement compared with the patching-based fixed-6.25 Hz variant and the speaker-finetuned Qwen-Omni series.
- **Low-frame-rate behavior.** Appendix E visualizes dynamic-frame-rate outputs under different target frame rates and analyzes audio-token losses. It suggests that degradation at 5.0–4.0 Hz is mainly associated with coarser temporal resolution and harder frame-length prediction.
- **Results on more tasks.** Appendices F and G report speech generation and audio understanding, showing that FlexiSLM retains strong audio-understanding accuracy even under aggressive input compression.
- **Implementation details.** Appendix H gives model and training hyperparameters, and Appendices I and J describe the flow-matching audio decoder and FlexiCodec tokenizer. Tables 19 and 20 provide qualitative samples.

6 Conclusion

We presented FlexiSLM, to our knowledge the first spoken language model with dynamic and controllable frame rates. FlexiSLM applies dynamic frame rates to both speech input and output, and introduces direct frame-rate conditioning so that a single model operates from 12.5 down to 4.0 Hz without retraining. Experiments show that FlexiSLM achieves the best speech-to-speech QA score among 7B SLMs at 12.5 Hz and remains stronger in s2s than fixed-rate 7B baselines at 6.25 Hz. More aggressive 5.0 and 4.0 Hz settings provide additional low-token operating points with larger but smooth quality degradation. Halving the output frame rate halves inference time with only a 1.0-point s2s loss. FlexiSLM demonstrates a practical paradigm for efficient spoken language models.

Limitations

FlexiSLM has several limitations. First, we have not explored post-training techniques such as preference optimization, which may further improve response quality, alignment, and speech naturalness. Second, the current model is non-streaming on both input and output; adapting the frame-merging module and audio decoder to causal or chunk-wise operation is an important step toward real-time conversation. Third, our training data has limited coverage of reasoning-intensive tasks, multi-turn dialogue, and multiple-choice formats, which may limit generalization to these scenarios. Finally, the human listening test in Appendix C shows that FlexiSLM still has room to improve output audio quality compared with the Qwen-Omni series, and the additional ablation in Appendix B shows that the controllable dynamic-frame-rate model does not yet match the strongest non-controllable grouped-patching variant. We discuss future work in Appendix K.

Acknowledgements

This work was conducted in collaboration with ByteDance. We thank Yutong Bian for his substantial contribution to FlexiSLM's open-source codebase. This work is partially supported by NSFC Grant 62376237; 2023 Shenzhen Stability Science Program; Program for Guangdong Introducing Innovative and Enterpreneurial Teams 2023ZT10X044.

References

Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Hynek Kydlíček, Agustín Piqueres Lajarín, Vaibhav Srivastav, et al. 2025. Smollm2: When smol goes big–data-centric training of a small language model. *arXiv preprint arXiv:2502.02737*.

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

Rosana Ardila, Megan Branson, Kelly Davis, Michael Kohler, Josh Meyer, Michael Henretty, Reuben Morais, Lindsay Saunders, Francis Tyers, and Gregor Weber. 2020. Common voice: A massively-multilingual speech corpus. In *Proceedings of the twelfth language resources and evaluation conference*, pages 4218–4222.

Siddhant Arora, Kai-Wei Chang, Chung-Ming Chien, Yifan Peng, Haibin Wu, Yossi Adi, Emmanuel Dupoux, Hung-Yi Lee, Karen Livescu, and Shinji Watanabe. 2025. On the landscape of spoken language models: A comprehensive survey. *arXiv preprint arXiv:2504.08528*.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, and 12 others. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *Preprint*, arXiv:2204.05862.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on Freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1533–1544, Seattle, Washington, USA. Association for Computational Linguistics.

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

Houwei Cao, David G Cooper, Michael K Keutmann, Ruben C Gur, Ani Nenkova, and Ragini Verma. 2014. Crema-d: Crowd-sourced emotional multimodal actors dataset. *IEEE transactions on affective computing*, 5(4):377–390.

Yiming Chen, Xianghu Yue, Chen Zhang, Xiaoxue Gao, Robby T Tan, and Haizhou Li. 2024. Voicebench: Benchmarking llm-based voice assistants. *arXiv preprint arXiv:2410.17196*.

Yunfei Chu, Jin Xu, Qian Yang, Haojie Wei, Xipin Wei, Zhifang Guo, Yichong Leng, Yuanjun Lv, Jinzheng He, Junyang Lin, et al. 2024. Qwen2-audio technical report. *arXiv preprint arXiv:2407.10759*.

Jonathan H. Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020. TyDi QA: A benchmark for information-seeking question answering in typologically diverse languages. *Transactions of the Association for Computational Linguistics*, 8:454–470.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

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

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

Ding Ding, Zeqian Ju, Yichong Leng, Songxiang Liu, Tong Liu, Zeyu Shang, Kai Shen, Wei Song, Xu Tan, Heyi Tang, et al. 2025. Kimi-audio technical report. *arXiv preprint arXiv:2504.18425*.

Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. 2023. Enhancing chat language models by scaling high-quality instructional conversations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3029–3051, Singapore. Association for Computational Linguistics.

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

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

Jesse Engel, Cinjon Resnick, Adam Roberts, Sander Dieleman, Mohammad Norouzi, Douglas Eck, and Karen Simonyan. 2017. Neural audio synthesis of musical notes with wavenet autoencoders. In *International conference on machine learning*, pages 1068–1077. PMLR.

Sefik Emre Eskimez, Xiaofei Wang, Manthan Thakker, Canrun Li, Chung-Hsien Tsai, Zhen Xiao, Hemin Yang, Zirun Zhu, Min Tang, Xu Tan, et al. 2024. E2 tts: Embarrassingly easy fully non-autoregressive zero-shot tts. In *2024 IEEE Spoken Language Technology Workshop (SLT)*, pages 682–689. IEEE.

Fahim Faisal, Sharlina Keshava, Md Mahfuz Ibn Alam, and Antonios Anastasopoulos. 2021. SD-QA: Spoken dialectal question answering for the real world. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3296–3315, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yuan Gong, Jin Yu, and James Glass. 2022. Vocalsound: A dataset for improving human vocal sounds recognition. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 151–155. IEEE.

Haorui He, Zengqiang Shang, Chaoren Wang, Xuyuan Li, Yicheng Gu, Hua Hua, Liwei Liu, Chen Yang, Jiaqi Li, Peiyang Shi, et al. 2024. Emilia: An extensive, multilingual, and diverse speech dataset for large-scale speech generation. *arXiv preprint arXiv:2407.05361*.

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 transactions on audio, speech, and language processing*, 29:3451–3460.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Hangrui Hu, Xinfa Zhu, Ting He, Dake Guo, Bin Zhang, Xiong Wang, Zhifang Guo, Ziyue Jiang, Hongkun Hao, Zishan Guo, et al. 2026. Qwen3-tts technical report. *arXiv preprint arXiv:2601.15621*.

Shengpeng Ji, Ziyue Jiang, Wen Wang, Yifu Chen, Minghui Fang, Jialong Zuo, Qian Yang, Xize Cheng, Zehan Wang, Ruiqi Li, et al. 2024. Wavtokenizer: an efficient acoustic discrete codec tokenizer for audio language modeling. *arXiv preprint arXiv:2408.16532*.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Hyunwoo Kim, Jack Hessel, Liwei Jiang, Peter West, Ximing Lu, Youngjae Yu, Pei Zhou, Ronan Bras, Malihe Alikhani, Gunhee Kim, Maarten Sap, and Yejin Choi. 2023. SODA: Million-scale dialogue distillation with social commonsense contextualization. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12930–12949, Singapore. Association for Computational Linguistics.

Matthew Le, Apoorv Vyas, Bowen Shi, Brian Karrer, Leda Sari, Rashel Moritz, Mary Williamson, Vimal Manohar, Yossi Adi, Jay Mahadeokar, et al. 2023. Voicebox: Text-guided multilingual universal speech generation at scale. *Advances in neural information processing systems*, 36:14005–14034.

Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. 2022. Deduplicating training data makes language models better. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8424–8445.

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

Jiaqi Li, Yao Qian, Yuxuan Hu, Leying Zhang, Xiaofei Wang, Heng Lu, Manthan Thakker, Jinyu Li, Sheng Zhao, and Zhizheng Wu. 2025b. Flexicodec: A dynamic neural audio codec for low frame rates. *arXiv preprint arXiv:2510.00981*.

Jiaqi Li, Dongmei Wang, Xiaofei Wang, Yao Qian, Long Zhou, Shujie Liu, Midia Yousefi, Canrun Li, Chung-Hsien Tsai, Zhen Xiao, et al. 2024. Investigating neural audio codecs for speech language model-based speech generation. In *2024 IEEE Spoken Language Technology Workshop (SLT)*, pages 554–561. IEEE.

Tianpeng Li, Jun Liu, Tao Zhang, Yuanbo Fang, Da Pan, Mingrui Wang, Zheng Liang, Zehuan Li, Mingan Lin, Guosheng Dong, et al. 2025c. Baichuan-audio: A unified framework for end-to-end speech interaction. *arXiv preprint arXiv:2502.17239*.

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Alpacaeval: An automatic evaluator of instruction-following models. https://github.com/tatsu-lab/alpaca_eval.

Shijia Liao, Yuxuan Wang, Tianyu Li, Yifan Cheng, Ruoyi Zhang, Rongzhi Zhou, and Yijin Xing. 2024. Fish-speech: Leveraging large language models for advanced multilingual text-to-speech synthesis. *arXiv preprint arXiv:2411.01156*.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. 2022. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*.

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

Aleksandr Meister, Matvei Novikov, Nikolay Karpov, Evelina Bakhturina, Vitaly Lavrukhin, and Boris Ginsburg. 2023. Librispeech-pc: Benchmark for evaluation of punctuation and capitalization capabilities of end-to-end asr models. In *2023 IEEE automatic speech recognition and understanding workshop (ASRU)*, pages 1–7. IEEE.

Fabian Mentzer, David Minnen, Eirikur Agustsson, and Michael Tschannen. 2023. Finite scalar quantization: Vq-vae made simple. *arXiv preprint arXiv:2309.15505*.

Eliya Nachmani, Alon Levkovitch, Roy Hirsch, Julian Salazar, Chulayuth Asawaroengchai, Soroosh Mariooryad, Ehud Rivlin, RJ Skerry-Ryan, and Michelle Tadmor Ramanovich. 2024. Spoken question answering and speech continuation using spectrogram-powered LLM. In *The Twelfth International Conference on Learning Representations*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

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. IEEE.

Soujanya Poria, Devamanyu Hazarika, Navonil Majumder, Gautam Naik, Erik Cambria, and Rada Mihalcea. 2019. Meld: A multimodal multi-party dataset for emotion recognition in conversations. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 527–536.

Vineel Pratap, Qiantong Xu, Anuroop Sriram, Gabriel Synnaeve, and Ronan Collobert. 2020. Mls: A large-scale multilingual dataset for speech research. *arXiv preprint arXiv:2012.03411*.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *International conference on machine learning*, pages 28492–28518. PMLR.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36:53728–53741.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. 2020. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 3505–3506.

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

Yirong Sun, Yizhong Geng, Peidong Wei, Yanjun Chen, Jinghan Yang, Rongfei Chen, Wei Zhang, and Xiaoyu Shen. 2025. Llaso: A foundational framework for reproducible research in large language and speech model. *arXiv preprint arXiv:2508.15418*.

Chao-Hong Tan, Qian Chen, Wen Wang, Chong Deng, Qinglin Zhang, Luyao Cheng, Hai Yu, Xin Zhang, Xiang Lv, Tianyu Zhao, et al. 2025. Drvoice: Parallel speech-text voice conversation model via dual-resolution speech representations. *arXiv preprint arXiv:2506.09349*.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following LLaMA model. https://github.com/tatsu-lab/stanford_alpaca.

Tongyi Fun Team, Qian Chen, Luyao Cheng, Chong Deng, Xiangang Li, Jiaqing Liu, Chao-Hong Tan, Wen Wang, Junhao Xu, Jieping Ye, et al. 2025.

Fun-audio-chat technical report. *arXiv preprint arXiv:2512.20156*.

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

Chaoren Wang, Heng Lu, Xueyao Zhang, Shujie Liu, Yan Lu, Jinyu Li, and Zhizheng Wu. 2026. Closing the modality reasoning gap for speech large language models. *arXiv preprint arXiv:2601.05543*.

Hankun Wang, Yiwei Guo, Chongtian Shao, Bohan Li, Xie Chen, and Kai Yu. 2025. Codecslime: Temporal redundancy compression of neural speech codec via dynamic frame rate. *arXiv preprint arXiv:2506.21074*.

Haibin Wu, Naoyuki Kanda, Sefik Emre Eskimez, and Jinyu Li. 2024. Ts3-codec: Transformer-based simple streaming single codec. *arXiv preprint arXiv:2411.18803*.

Jin Xu, Zhifang Guo, Jinzheng He, Hangrui Hu, Ting He, Shuai Bai, Keqin Chen, Jialin Wang, Yang Fan, Kai Dang, Bin Zhang, Xiong Wang, Yunfei Chu, and Junyang Lin. 2025a. Qwen2.5-omni technical report. *Preprint*, arXiv:2503.20215.

Jin Xu, Zhifang Guo, Hangrui Hu, Yunfei Chu, Xiong Wang, Jinzheng He, Yuxuan Wang, Xian Shi, Ting He, Xinfa Zhu, et al. 2025b. Qwen3-omni technical report. *arXiv preprint arXiv:2509.17765*.

Zhangchen Xu, Fengqing Jiang, Luyao Niu, Yuntian Deng, Radha Poovendran, Yejin Choi, and Bill Yuchen Lin. 2025c. Magpie: Alignment data synthesis from scratch by prompting aligned LLMs with nothing. In *The Thirteenth International Conference on Learning Representations*.

Junichi Yamagishi, Christophe Veaux, and Kirsten MacDonald. 2019. Cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit (version 0.92). *The Rainbow Passage which the speakers read out can be found in the International Dialects of English Archive:(http://web. ku. edu/~ idea/readings/rainbow. htm)*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

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

Aohan Zeng, Zhengxiao Du, Mingdao Liu, Kedong Wang, Shengmin Jiang, Lei Zhao, Yuxiao Dong, and Jie Tang. 2024. Glm-4-voice: Towards intelligent and human-like end-to-end spoken chatbot. *arXiv preprint arXiv:2412.02612*.

Dong Zhang, Gang Wang, Jinlong Xue, Kai Fang, Liang Zhao, Rui Ma, Shuhuai Ren, Shuo Liu, Tao Guo, Weiji Zhuang, et al. 2025a. Mimo-audio: Audio language models are few-shot learners. *arXiv preprint arXiv:2512.23808*.

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

Xin Zhang, Dong Zhang, Shimin Li, Yaqian Zhou, and Xipeng Qiu. 2023. Speechtokenizer: Unified speech tokenizer for speech large language models. *arXiv preprint arXiv:2308.16692*.

Wenting Zhao, Xiang Ren, Jack Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. 2024. WildChat: 1m ChatGPT interaction logs in the wild. In *The Twelfth International Conference on Learning Representations*.

Rui-Chen Zheng, Wenrui Liu, Hui-Peng Du, Qinglin Zhang, Chong Deng, Qian Chen, Wen Wang, Yang Ai, and Zhen-Hua Ling. 2025. Say more with less: Variable-frame-rate speech tokenization via adaptive clustering and implicit duration coding. *arXiv preprint arXiv:2509.04685*.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*.

Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J Zico Kolter, and Matt Fredrikson. 2023. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*.

A Construction of FlexiSLM-Data

As described in Section 3.4 and Table 2, FlexiSLM is trained in Stages 2 and 3 on a mixture of tasks, with an emphasis on speech-to-speech dialogue data:

- **Speech-to-speech (s2s) dialogue**: mainly **FlexiSLM-Data** (1.4M samples).
- **TTS**: Emilia (He et al., 2024), MLS (Pratap et al., 2020).
- **ASR**: MLS and LibriSpeech (Panayotov et al., 2015).
- **Audio understanding**: LLaSO-Instruct (Sun et al., 2025) covering emotion, accent, vocal recognition, etc.
- **Text-to-text (t2t) dialogue**: TriviaQA (Joshi et al., 2017), WebQuestions (Berant et al., 2013).

This section details the pipeline for building a large-scale speech-to-speech dialogue dataset, which we call **FlexiSLM-Data**. **Instructions for downloading FlexiSLM-Data are provided in**

Dataset	Type	# Prompts
TriviaQA (Joshi et al., 2017)	QA	138K
WebQuestions (Berant et al., 2013)	QA	3.8K
TyDiQA (Clark et al., 2020)	Multilingual QA	167K
Alpaca (Taori et al., 2023)	Instruction	52K
SmolTalk2 (Allal et al., 2025)	Instr. / Dialog	385K
SODA (Kim et al., 2023)	Dialogue	1.48M
Magpie-Pro (Xu et al., 2025c)	Instruction / Dialog	1M
UltraChat (Ding et al., 2023)	Multi-turn Dialog	949K
HH-RLHF (Bai et al., 2022)	Dialog / Preference	167K
WildChat (Zhao et al., 2024)	Real-user Dialog	159K

Table 8: Text prompt sources used to construct FlexiSLM-Data. Dataset names link to the corresponding Hugging Face repositories.

our open-source repository. The construction pipeline has the following steps:

- **Prompt collection.** We collect text prompts from public QA, instruction-following, and dialogue datasets (Table 8). For multi-turn datasets, we use the user's first-turn utterance as the prompt, skipping samples whose first turn is a generic greeting (e.g., "Hello").
- **Response generation.** We use Qwen3-Omni (Xu et al., 2025b) to generate text responses for these prompts. We choose a spoken language model rather than a text-only LLM because Qwen3-Omni produces speech-friendly responses that are short, conversational, and free of formatting that does not transfer to speech (e.g., bullet points, code blocks, or long enumerations). We feed the *text* version of the prompt rather than its synthesized speech, since text-input responses are typically more accurate.
- **Speech synthesis.** The generated responses are synthesized into speech by Qwen3-TTS (Hu et al., 2026) with a fixed speaker "Ryan". The speech prompts are synthesized using Fish-Audio TTS[3] (Liao et al., 2024), with speaker prompts randomly sampled from English Emilia utterances longer than 5 seconds (He et al., 2024). This produces 4M samples with a total audio duration of 30K hours.
- **Quality filtering.** We apply format-based filtering to remove samples containing code, formulas, excessive punctuation, or non-target languages; correctness filtering using the DeepSeek-V4-Flash API; and ASR-based filtering with Whisper-medium to discard cases with WER greater than 20%.
- **Final scale.** After filtering, we obtain 1.4M speech-to-speech dialogue samples, totaling 9.9K hours of speech (2.7K hours of user prompts and 7.2K hours of assistant responses). Sample data are shown as text transcriptions in Table 20.

FlexiSLM-Data is currently limited to single-turn interactions. In the future, we plan to extend it to multi-turn natural dialogue.

[3] https://huggingface.co/fishaudio/s1-mini

B Additional Ablation Study

Method	OAB+VB AVG ↑	ASR WER ↓ LibriSpeech		TTS WER ↓
	s2t / s2s	clean	other	
Baseline FlexiSLM-Stage 2	**68.7 / 63.0**	2.92	7.20	3.11
Encoder / LLM Backbone Choice				
Switch to SenseVoice encoder	68.4 / 62.2	2.73	6.54	3.49
Switch to Qwen-ASR audio encoder	65.9 / 59.8	**2.08**	**4.42**	**2.94**
Switch to Qwen2.5-Omni Thinker backbone	67.0 / 61.6	2.27	5.33	3.11
Architecture Design				
w/o input merging Transformer	65.7 / 60.1	6.45	12.33	3.66
Activate Talker-to-Thinker link in FlexiSLM-Stage 2	60.5 / 55.9	7.75	12.67	3.15

Table 9: Additional ablation study on FlexiSLM-Stage2. "OAB+VB AVG" is the average accuracy on OpenAudioBench and VoiceBench, scaled to 0–100. We underline metrics that are degraded from the baseline.

For this study, we further ablate two orthogonal axes that complement the dynamic frame rate ablation in Section 4.4: (1) the choice of audio encoder and LLM backbone, and (2) architectural decisions inside FlexiSLM. Ablations follow the training scheme in Section 4.4 and are evaluated at 6.25 Hz input / 6.25 Hz output. Results are reported in Table 9.

• **FlexiSLM supports different encoders; performance varies.** Switching to the SenseVoice encoder slightly hurts the spoken QA averages (68.7 → 68.4 s2t, 63.0 → 62.2 s2s) while moderately improving ASR (test-other 7.20 → 6.54). Switching to the Qwen-ASR encoder further improves ASR WER, yielding the strongest ASR (clean 2.08, other 4.42) and TTS (2.94) numbers in the table, but at a clear cost in spoken QA (65.9/59.8). This is consistent with the intuition that ASR-pretrained encoders concentrate capacity on transcription-friendly features at the expense of higher-level semantics, while our default encoder gives the best joint operating point across understanding and generation.

• **FlexiSLM supports different LLM backbones; performance varies.** Replacing our backbone with the Qwen2.5-Omni Thinker degrades spoken QA (68.7/63.0 → 67.0/61.6) but improves ASR WER

Model	OAB AVG ↑	ASR WER ↓		TTS
	s2t / s2s	clean	other	WER ↓
Baseline: FlexiSLM Stage 2	65.0 / 61.0	2.68	6.70	3.11
Switch to 6.25 Hz grouped patching	**66.3 / 63.3**	**2.19**	**5.63**	**2.70**
Switch to retrained fixed 6.25 Hz codec	65.0 / 60.7	2.67	7.15	4.47

Table 10: Comparison of fixed-rate FlexiSLM alternatives at 6.25 Hz input/output.

(2.92/7.20 → 2.27/5.33). The spoken QA results indicate that our default text LLM-based backbone provides a stronger knowledge and reasoning prior than an SLM backbone. In contrast, the speech-pretrained SLM backbone requires less adaptation for speech recognition.

• **The input merging Transformer is important.** Removing the merging transformer and feeding raw merged features directly to the LLM causes ASR to collapse (clean 2.92 → 6.45, other 7.20 → 12.33) and spoken QA to drop (68.7/63.0 → 65.7/60.1). This confirms that the merging Transformer is essential for re-aligning dynamically merged features into a representation the LLM can consume.

• **The Talker-to-Thinker connection requires full-parameter activation rather than LoRA.** Activating the Talker-to-Thinker connection in Stage 2 instead of Stage 3 produces worse results than the baseline. We attribute this to the connection introducing a large amount of new information that the limited LoRA capacity cannot absorb, creating a destabilizing feedback loop between the partially trained Talker and the frozen backbone. This justifies deferring the connection to Stage 3, where full-parameter fine-tuning provides sufficient capacity to integrate the bidirectional signal.

• **Fixed-rate alternatives.** We further evaluate two alternatives to FlexiSLM's dynamic-frame-rate tokenizer at the same 6.25 Hz input/output operating point. Table 10 compares the Stage 2 baseline with grouped patching and a retrained fixed-codec variant. For grouped patching, we combine 12.5 Hz FlexiCodec tokens into 6.25 Hz groups by predicting two consecutive tokens in one forward pass. In our implementation, the second token in each group has a token delay of 1. For the fixed-codec variant, we retrain the 6.25 Hz tokenizer and flow-matching decoder following the official FlexiCodec recipe. This experiment complements the main-text ablation in Section 4.4: the original ablation isolates the dynamic merging mechanism, whereas this additional comparison tests complete fixed-rate systems trained directly at the target rate.

• **Results of fixed-rate alternatives.** Compared with the FlexiSLM Stage 2 baseline, grouped patching improves all reported metrics: OAB AVG increases from 65.0/61.0 to 66.3/63.3 for s2t/s2s, ASR WER and TTS WER are also improved. This gain likely reflects both the higher-resolution 12.5 Hz codec target and optimization for a single 6.25 Hz setting, but the variant no longer provides frame-rate controllability. By contrast, the retrained fixed-codec model matches the baseline on s2t but lowers s2s (61.0 → 60.7) and substantially worsens TTS WER (3.11 → 4.47). Overall, grouped patching is a strong non-controllable 6.25 Hz variant, while FlexiSLM still outperforms the fixed-6.25 Hz codec variant.

System	Naturalness	Fidelity	Intelligibility	Overall
Qwen3-Omni-30B	**4.79**±0.10	**4.79**±0.09	**4.84**±0.09	**4.81**
Qwen2.5-Omni-7B	4.71±0.10	4.79±0.07	4.87±0.07	4.79
FlexiSLM-7B (12.5 Hz)	4.17±0.23	3.94±0.20	4.53±0.20	4.22
FlexiSLM-7B (6.25 Hz)	4.22±0.21	3.98±0.21	4.42±0.22	4.21
Mimo-Audio-7B	3.92±0.22	3.99±0.20	4.40±0.19	4.10
FlexiSLM-7B (4.0 Hz)	3.69±0.25	3.66±0.23	4.21±0.26	3.85

Table 11: Human listening-test results (mean ± 95% confidence interval).

C Human Listening Test

To assess the perceptual speech quality of FlexiSLM, we conduct a human mean opinion score (MOS) evaluation comparing three FlexiSLM operating points with three representative SLM baselines.

Protocol. Eight listeners with experience in speech or audio research evaluated the audio outputs of six SLMs on the same 10 spoken-dialogue prompts. Listeners were instructed to judge only the perceptual audio quality of each response, independent of factual correctness, using a five-point scale from 1 to 5 in increments of 0.5. System identities were hidden, and presentation order was randomized. Listeners rated three dimensions: (1) **naturalness**, covering rhythm, intonation, expressiveness, and absence of robotic delivery; (2) **fidelity**, reflecting overall audio quality; and (3) **intelligibility**, measuring how easily the spoken words can be understood. We report the mean and 95% confidence interval, computed as 1.96 times the standard error over the 80 ratings for each system and dimension.

• **The Qwen systems achieve the highest MOS scores, followed by FlexiSLM.** The two Qwen systems receive the highest overall scores, with their largest advantage over FlexiSLM appearing

in fidelity. Several factors may contribute to their higher audio quality: frame rate, training data, and RL post-training. In terms of frame rate, Qwen2.5-Omni generates at a substantially higher output frame rate (50 Hz) than FlexiSLM, which provides more temporal capacity for acoustic detail. Frame rate alone cannot explain Qwen3-Omni's advantage, however, because it operates at 12.5 Hz. Its larger model size, larger-scale audio training corpus, and reinforcement-learning post-training likely improve generation quality. Although detailed data statistics are not publicly specified, both Qwen systems appear to use substantially more high-quality audio data, whereas FlexiSLM's dialog data is synthetic and comparatively limited. Both Qwen systems also use RL for speech-generation quality, whereas FlexiSLM does not. This comparison suggests concrete directions for improving FlexiSLM: scaling curated speech-generation data and adding rewards for naturalness, fidelity, prosody, and stability are promising ways to improve FlexiSLM without giving up its low-frame-rate efficiency.

• **Reducing FlexiSLM from** 12.5 **to** 6.25 **Hz causes no clear perceptual loss.** The two settings have nearly identical overall scores (4.22 and 4.21), and their confidence intervals overlap. Together with the approximately 2× lower RTF in Table 5, this result supports 6.25 Hz as a favorable efficiency-quality operating point.

• **At** 4.0 **Hz, speech primarily loses naturalness and fidelity.** FlexiSLM at 4.0 Hz shows only a modest decrease in intelligibility (4.21), but naturalness and fidelity drop more substantially to 3.69 and 3.66, respectively. This pattern complements the very-low-frame-rate analysis in Appendix E, indicating that coarse temporal resolution and more difficult frame-length prediction primarily weaken prosodic and acoustic detail.

Recommended operating points. Considering the visualization and low-frame-rate analysis in Appendix E, together with the human listening results above, we recommend selecting the target frame rate according to deployment requirements. For the input frame rate, we recommend keeping 12.5 Hz for dialogue tasks and using task-specific low frame rates for audio-understanding tasks (Table 17). For the output frame rate, the 12.5 Hz setting is appropriate when stable speech generation, word-level intelligibility, and answer correctness are the main priorities, since it gives the strongest benchmark results and avoids duration prediction errors from frame merging. The 6.25 Hz setting is our general recommendation for spoken-dialogue serving: it provides a good quality-efficiency balance, nearly halves the output-token rate and RTF relative to 12.5 Hz, and preserves comparable spoken-QA scores and human listening quality. Intermediate settings such as 8.0 Hz can be used for intermediate budgets. By contrast, 5.0 and especially 4.0 Hz should be treated as aggressive compression modes rather than default user-facing settings. They may be suitable for edge-device or resource-constrained scenarios where latency, throughput, or token budget is more important than high naturalness and fidelity.

Evaluation set	N_total	Exact	Exact (%)	$J \geq 0.9$	$J \geq 0.8$	$J \geq 0.7$
Llama Questions	300	60	20.00	62	68	99
TriviaQA	1,000	7	0.70	16	30	73
Web Questions	1,000	15	1.50	15	30	69
AdvBench	520	0	0.00	0	0	0
AlpacaEval	636	8	1.26	10	13	24
CommonEval	200	2	1.00	2	2	5
IFEval	345	0	0.00	0	0	0
SD-QA	553	201	36.35	201	203	207

Table 12: Prompt overlap between FlexiSLM's evaluation sets and filtered training set.

D Training-Evaluation Data Overlap Report

To examine potential overlap between our training data and the evaluation benchmarks, we compare the text prompts in each evaluation set against the prompts in our filtered training set. Table 12 reports the number of unique evaluation prompts with at least one matching training prompt.

Deduplication method. We report both exact and approximate matches. For exact matching, we count cases where an evaluation prompt appears as a substring of any training prompt after normalization, including lowercasing, collapsing consecutive whitespace, and removing leading or trailing whitespace. For approximate matching, we follow the MinHash-based deduplication practice of Lee et al. (2022), compute Jaccard similarity over 5-grams for each candidate pair, and report cumulative counts at $J \geq 0.7$, $J \geq 0.8$, and $J \geq 0.9$. The results are analyzed below.

• **Substantial exact overlap is concentrated in Llama Questions and SD-QA. Other test subsets show little contamination.** We find exact matches for 20.00% of Llama Questions and 36.35% of SD-

Model	In FR	Out FR	OpenAudioBench					VoiceBench							Overall		Overall Rank	
			Llama Q.		Web Q.		TriviaQA	Alpaca		Common	IFEval	SD-QA		AdvB.				
			bef.	aft.	bef.	aft.	bef.=aft.	bef.	aft.	bef.=aft.	bef.=aft.	bef.	aft.	bef.=aft.	bef.	aft.	bef.	aft.
Qwen3-Omni-30B-A3B	12.5	12.5	72	70	63	62	67	3.60	3.60	3.58	2.60	62	56	4.89	69.7	68.6	1	1
Fun-Audio-Chat-8B	25	25	74	71	60	59	59	3.77	3.76	3.46	2.55	54	49	4.81	67.3	66.2	2	2
GLM-4-Voice-9B	12.5	12.5	62	64	48	48	47	2.82	2.83	2.69	1.97	45	40	4.49	55.2	54.8	12	12
Mimo-Audio-7B	6.25	12.5	71	68	55	55	48	2.90	2.90	3.06	2.12	47	44	4.43	59.0	58.2	9	9
Kimi-Audio-7B	12.5	12.5	69	65	56	56	54	2.74	2.73	2.46	2.11	37	35	4.73	57.2	56.3	10	10
Qwen2.5-Omni-7B	25	25	72	68	59	59	57	3.07	3.07	3.24	2.12	52	49	4.88	63.3	62.4	7	7
FlexiSLM-7B-Stage3	12.5	12.5	75	73	60	59	60	3.50	3.60	3.48	2.33	60	50	4.83	67.2	65.9	3	3
	12.5	6.25	74	71	59	58	58	3.44	3.43	3.42	2.26	59	49	4.87	66.2	64.5	4	4
	6.25	12.5	73	70	56	54	58	3.38	3.37	3.32	2.28	59	49	4.85	65.3	63.4	5	5
	6.25	6.25	71	67	55	54	57	3.31	3.31	3.30	2.21	58	49	4.85	64.3	62.6	6	6
	5.0	5.0	67	63	52	51	52	3.05	3.04	3.04	2.09	53	45	4.79	60.4	58.8	8	8
	4.0	4.0	64	60	48	47	48	2.84	2.82	2.83	1.85	48	42	4.67	56.5	55.1	11	11

Table 13: Speech-to-speech (s2s) evaluation results after removing test samples whose prompts overlap with training prompts. "bef." uses the original evaluation sets, and "aft." removes all prompts with an exact training-set match. A shared "bef.=aft." column indicates that the reported score is unchanged. Overall ranks are computed across all model configurations shown in the table.

Evaluation set	Example exact-match prompt
Llama Questions	What is the capital of New Zealand? Who wrote the book "To Kill a Mockingbird"? What is the capital of Morocco? How many planets are there in our solar system?
SD-QA	What language do Chechens speak? When was quantum field theory developed? Who was the last Han Dynasty Emperor?

Table 14: Examples of exact train-evaluation prompt matches in Llama Questions and SD-QA.

QA. In comparison, the exact-match rate is at most 1.50% for every other benchmark, with no exact or approximate matches in AdvBench and IFEval. The increase at lower Jaccard thresholds for TriviaQA and Web Questions indicates some lexical similarity, but their direct exact-overlap rates remain low.

• **The exact matches are primarily short factual questions.** As illustrated in Table 14, the overlapping prompts generally ask for common factual knowledge, such as capitals, authors, languages, or historical facts. Because such questions can occur independently in multiple public QA collections, their presence does not by itself establish that benchmark instances were intentionally included in training. Nevertheless, evaluating on these samples may overestimate generalization. We therefore conduct the re-evaluation detailed below.

Re-evaluation after deduplication. For a fairer comparison, we recompute LLM-judge scores after removing the exact-match prompts identified above, and compare speech-to-speech results before and after deduplication. Table 13 reports the results. The analysis is as follows.

• **Deduplication changes absolute scores but preserves the model ranking.** Removing exact-match prompts reduces the Overall score for most systems, likely because the removed samples are common factual questions that are easy for the models to answer. Although the absolute scores change, the Overall Rank is unchanged for every model. FlexiSLM at $12.5/12.5$ Hz remains third overall ($67.2 \rightarrow 65.9$), behind Qwen3-Omni-30B-A3B and Fun-Audio-Chat, while retaining the highest score among the listed 7B models.

• **The main low-frame-rate conclusion remains unchanged.** After deduplication, FlexiSLM at $6.25/6.25$ Hz obtains an Overall score of 62.6, remaining above all listed 7B baselines, including Qwen2.5-Omni-7B (62.4), Mimo-Audio-7B (58.2), and Kimi-Audio-7B (56.3). The ordering among all FlexiSLM operating points is also unchanged. These results show that the comparative conclusions in Table 3 are not driven by overlapping evaluation prompts.

E Visualization and Analysis of Low Frame Rates

Figure 4 provides a qualitative visualization of FlexiSLM's dynamic-rate audio representations under several target frame-rate conditions. The top panel shows the generated mel spectrogram for the spoken response, while the lower panels align the text, fixed-frame-rate tokens, and FlexiSLM tokens produced at progressively lower target rates. Because

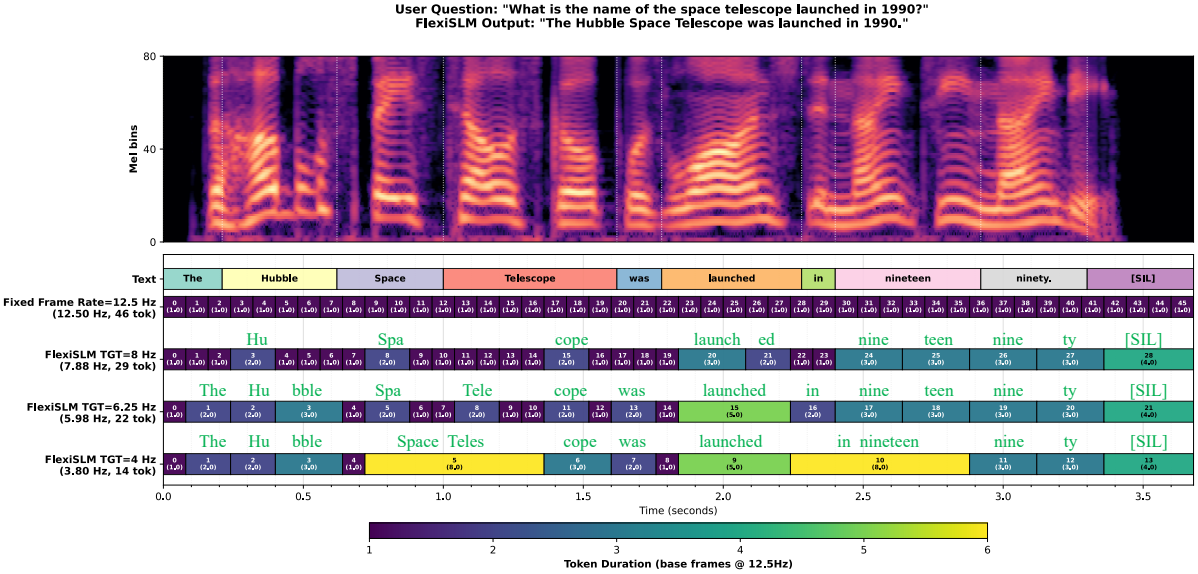

Figure 4: Visualization of FlexiSLM token outputs for the response "The Hubble Space Telescope was launched in 1990." The figure compares FlexiCodec tokenizations under different target frame rates. We annotate the subwords aligned to each segment and confirm them by listening.

it is difficult to force FlexiSLM runs at different target frame rates to produce exactly the same text and audio, we take a 12.5 Hz FlexiSLM audio response and encode it with FlexiCodec under different target frame rates.

Visualization analysis. Compared with the fixed 12.5 Hz representation, FlexiSLM uses fewer and longer-duration speech tokens as the requested frame rate decreases, while preserving a subword-level alignment with the generated utterance. This example illustrates how frame-rate controllability changes the temporal granularity of the speech-token sequence rather than simply truncating or uniformly dropping output content.

Visualization analysis for very low frame rates. The visualization also helps explain the degradation observed at aggressive operating points such as 5.0 and 4.0 Hz. At moderate compression levels such as 8.0 and 6.25 Hz, the generated token sequence still covers the full response and keeps a mostly subword- or phoneme-level alignment with the intended utterance, which explains the modest audio quality decrease at these intermediate operating points. At 4.0 Hz, however, each speech token must cover a much longer span, and the generated sequence becomes too coarse to preserve some local phonetic details (e.g., "Space Teles", "in nineteen"). This suggests that very-low-frame-rate degradation may be driven primarily by phonetic information loss and the increased difficulty of assigning accurate frame lengths to long-duration tokens. Reduced temporal resolution may also weaken prosodic detail, since fewer tokens are available to model fine-grained duration and transition patterns.

In FR	Out FR	Audio token loss	Audio length token loss / top-1 ACC
12.5	12.5	3.55	0 / 100%
12.5	6.25	3.58	0.68 / 29%
6.25	12.5	3.55	0 / 100%
6.25	6.25	3.58	0.68 / 29%
5.0	5.0	3.50	0.78 / 23%
4.0	4.0	3.59	0.93 / 19%

Table 15: Average audio-token loss and audio-length-token prediction quality for the final FlexiSLM-Stage3 checkpoint on a 100-sample training subset.

Frame-length prediction becomes harder at very low frame rates. To further diagnose the source of degradation, we evaluate the final Stage 3 checkpoint on a 100-sample training subset and report average audio-token loss together with audio-length-token loss and top-1 accuracy in Table 15. The audio-token loss stays nearly unchanged across frame-rate settings (3.50–3.59), suggesting that discrete audio-token prediction itself is not the main source of degradation at very low rates. In contrast, the length-token prediction becomes much harder as the output frame rate decreases: when the output

Model	FR	TTS WER ↓	Dialog WER ↓
CosyVoice	25	3.20	–
Qwen3-Omni	12.5	3.34	4.32
Mimo-Audio	25	14.72	20.16
Qwen2.5-Omni	25	3.18	6.33
FlexiSLM	12.5	**2.14**	4.52
	8.0	2.47	**4.41**
	6.25	2.87	5.83
	5.0	4.16	9.03

Table 16: Speech generation WER (%). **FR**: average output frame rate (Hz). **TTS**: LibriSpeech-PC WER. **Dialog**: OpenAudioBench response WER.

frame rate is 12.5 Hz, no merging is needed and the length prediction is trivial (0 loss, 100% top-1 accuracy), whereas the length-token loss increases from 0.68 at 6.25 Hz to 0.78 at 5.0 Hz and 0.93 at 4.0 Hz, with top-1 accuracy dropping from 29% to 23% and 19%. Rows with the same output frame rate but different input frame rates have identical losses, indicating that this diagnostic mainly captures an output-side duration-modeling difficulty rather than an input-side ASR effect. Together with the visualization, these results suggest that very-low-frame-rate degradation is partly caused by unstable or inaccurate frame-length prediction: each token must represent a longer acoustic span, making it harder to assign correct durations and preserve fine-grained phonetic and prosodic details.

F Speech Generation Quality Evaluation

Table 16 evaluates speech intelligibility using WER metrics. For TTS, we use the E2TTS-Evalkit (Eskimez et al., 2024) to compute TTS WER on LibriSpeech-PC (Meister et al., 2023). For dialog responses, we compute WER between the SLM's generated text and the whisper-large-v3 (Radford et al., 2023) transcription of its generated speech on OpenAudioBench. Baselines include CosyVoice (Du et al., 2024a) and three SLMs.

Results show that FlexiSLM at 12.5 Hz achieves 2.14% TTS WER, outperforming Qwen2.5-Omni (3.18%) and Qwen3-Omni (3.34%). On dialog responses, FlexiSLM obtains 4.52% WER at 12.5 Hz and 4.41% at 8.0 Hz, both better than Qwen2.5-Omni (6.33%) and comparable to Qwen3-Omni (4.32%). Reducing the frame rate to 6.25 Hz remains acceptable (2.87% TTS, 5.83% dialog), while the 5.0 Hz row shows a higher intelligibility cost (4.16% TTS, 9.03% dialog), consistent with the spoken-QA degradation in Table 3.

Error analysis. Manual inspection reveals different sources of high WER across systems. For Mimo-Audio, many errors are due to text–speech alignment or output-format failures: the model sometimes does not synthesize speech, stops before reading the complete response, or produces speech that diverges from the textual answer, including unintended continuation or language switching. For Qwen2.5-Omni, errors are often associated with long responses that exceed the 30-second ASR evaluation window, leading to incomplete transcription, as well as verbalization issues such as number-reading errors and word-fragmentation artifacts.

The results show that FlexiSLM at 12.5 Hz achieves the highest average accuracy of **65.8%**, outperforming Gemini-2.5-Pro (48.3%) and LLaSO-3B (58.3%). This confirms that FlexiSLM possesses multi-task audio understanding capability.

Unlike spoken QA and ASR, audio-understanding accuracy remains stable under aggressive input compression: FlexiSLM retains 64.7%, 64.0%, 63.4%, and 64.1% average accuracy at 8.0, 6.25, 5.0, and 4.0 Hz, respectively. This is likely because these are sequence-level classification problems, where each audio clip is mapped to a single categorical label (emotion, accent, gender, instrument, etc.). Aggressive frame merging preserves the global acoustic statistics needed for the decision, while tasks such as ASR or fine-grained instruction following depend more heavily on local phonetic detail.

G Audio Understanding Performance

Table 17 presents results on audio-understanding tasks from the LLaSO-Eval (Sun et al., 2025) benchmark, covering emotion recognition (CremaD (Cao et al., 2014), MELD (Poria et al., 2019)), accent recognition (CommonVoice (Ardila et al., 2020), VCTK (Yamagishi et al., 2019)), vocal-sound, instrument, and music-source classification (VocalSound (Gong et al., 2022), NSynth (Engel et al., 2017)), and gender classification (VocalSound, VCTK). All tasks are multiple-choice classification; we report top-1 accuracy (%). Some baselines, such as GLM-4-Voice, are not trained on all evaluated tasks and may naturally underperform outside their training distribution. In contrast, LLaSO-3B (Sun et al., 2025) and FlexiSLM are trained on the full set of LLaSO-Eval tasks and have a better train-test distribution match.

Model	FR	Emotion		Accent		Vocal	Music Source	Instrument	Gender		Avg↑
		CremaD	Meld	Com.Voice	VCTK	VocalSound	Nsynth	Nsynth	VocalSound	VCTK	ACC
Gemini 2.5-Pro	–	8	17	38	47	79	41	36	75	94	48.3
Qwen2-Audio-7B	50	31	25	3	20	85	36	**60**	65	95	47.0
GLM4-Voice-9B	12.5	5	17	2	12	24	23	0	17	19	13.2
LLaSO-3B	50	27	24	75	54	74	57	46	76.5	99	58.3
Kimi-Audio-7B	12.5	32	**55**	19	38	83.5	38	26	66	98	46.8
Qwen2.5-Omni-7B	25	24	35	10	29	**93**	43	51	**78**	98	48.6
FlexiSLM-7B	12.5	49	50	**81**	**64**	91	**84**	52	74.5	**100**	**65.8**
	8.0	49	49	80	61	90	82	53	74.5	**100**	64.7
	6.25	47	49	**81**	60	91	81	52	73.5	99	64.0
	5.0	**50**	48	79	61	89.5	77	50	75	99	63.4
	4.0	48	49	79	**64**	88.5	79	55	75.5	99	64.1

Table 17: Top-1 accuracy (%) on audio-understanding tasks from LLaSO-Eval. For FlexiSLM, we test different input frame rates.

H Additional Model and Training Details

Talker Transformer configuration. The Talker module of FlexiSLM uses the same Transformer implementation as the Thinker LLM. It has a hidden size of 1280, 20 Transformer layers, 8 attention heads, and an intermediate size of 5,120. FlexiCodec and its SenseVoice (An et al., 2024) semantic encoder have 450M parameters in total.

Training configuration. We use AdamW optimizer (Loshchilov and Hutter, 2017) for training. The Talker Transformer, Talker-to-Thinker connection, and Frame Merging Module use a learning rate of 2×10^{-4} in each stage. The LLM backbone uses a learning rate of 2×10^{-5} in Stage 2 and 1×10^{-6} in Stage 3. We apply LoRA to the LLM backbone with rank $r = 32$ and LoRA scaling parameter $\alpha = 64$. For each stage, we use a 5% linear learning-rate warmup followed by cosine decay to 1/10 of the maximum learning rate. All stages use a dynamic batch size of 3K tokens per GPU. Stages 1 and 2 use Distributed Data Parallel; Stage 3 uses DeepSpeed ZeRO-2 (Rasley et al., 2020). Stage 1 spans 300K steps, Stage 2 spans 240K steps (3 epochs), and Stage 3 spans 160K steps. Gradient accumulation is set to 1 in Stages 1 and 2, and to 2 in Stage 3.

I Flow-Matching Audio Decoder Details

FlexiSLM uses the audio decoder provided in the open-source FlexiCodec repository[4]. The decoder is a two-stage pipeline: a VoiceBox (Le et al., 2023)-style NAR flow-matching model followed by a vocoder.

The NAR flow-matching model takes FlexiCodec's FSQ semantic tokens and associated frame lengths as input, and generates mel spectrograms conditioned on a short reference audio prompt that defines the target speaker identity. The model input is a 12.5 Hz fixed-frame-rate sequence interpolated to 50 Hz to match the mel-spectrogram rate. To construct the 12.5 Hz sequence, each FlexiCodec FSQ semantic token is repeated according to its frame length. The flow-matching model learns a velocity field $v_\theta(\mathbf{z}_t, t)$ that transports samples from a Gaussian prior $\mathbf{z}_0 \sim \mathcal{N}(0, I)$ to the mel-spectrogram distribution $\mathbf{z}_1$ under the conditioning inputs. The mel spectrograms are then converted to 24 kHz waveforms by a pretrained Vocos (Siuzdak, 2023) vocoder.

J Audio Tokenizer Details

We use pretrained FlexiCodec (Li et al., 2025b) to encode dynamic-frame-rate speech tokens as FlexiSLM's training target. Our Talker Transformer predicts FlexiCodec's FSQ tokens and the associated per-token frame lengths, and the flow-matching decoder (Appendix I) synthesizes the final waveform. FlexiCodec follows an encoder-quantizer-decoder architecture and has the following two features:

ASR-Feature-Assisted Dual-Stream Encoding. FlexiCodec encodes FSQ semantic tokens through a semantic-encoding branch. The semantic-encoding branch encodes SenseVoice (An et al., 2024) ASR features at 12.5 Hz, while a parallel acoustic-encoding branch encodes RVQ tokens from raw waveforms. In FlexiSLM, we use only the semantic branch; acoustic reconstruction is handled by the flow-matching audio decoder.

Dynamic Frame Rate via Frame Merging. The Frame Merging Module is applied to the 12.5 Hz ASR features. Adjacent frames whose cosine simi-

[4] https://github.com/amphionteam/flexicodec

larity exceeds a threshold τ are merged, producing a variable-frame-length sequence. The merged features are then quantized via FSQ (in the semantic-encoding branch) or RVQ (in the acoustic-encoding branch). By varying τ from 0.8 to 1.0, FlexiCodec supports average frame rates from 3 Hz to 12.5 Hz. At 12.5 Hz, no merging occurs and FlexiCodec behaves as a standard fixed-rate codec.

K Future Work

While FlexiSLM validates dynamic and controllable frame rates inside an SLM, several directions remain open and motivate our future work:

• **Reasoning, multi-turn, and tool-use coverage.** FlexiSLM-Data is dominated by single-turn factual and instruction-following responses. Extending the data construction pipeline to multi-turn dialogues, chain-of-thought reasoning, and tool-augmented interactions would broaden the model's capability surface while keeping the same dynamic-rate backbone.

• **Streaming and on-device deployment.** The current FlexiSLM is non-streaming on both input and output. Adapting the Frame Merging Module and audio decoder to causal or chunk-based streaming, combined with FlexiSLM's frame-rate controllability, would enable real-time conversation and let a single model match heterogeneous compute budgets from cloud to edge devices.

• **Full-duplex SLMs.** The Talker-to-Thinker connection introduced in Section 3.1 routes the actually-rendered speech tokens back to the Thinker LLM, which gives the Thinker a coherent acoustic context across turns. This is a natural foundation for full-duplex SLMs that can listen, think, and speak concurrently. However, dynamic-frame-rate compression in streaming full-duplex scenarios may be challenging, as the merging strategies for the user stream and the assistant stream may differ.

• **Post-training for spoken dialogue.** We have not yet explored RLHF (Ouyang et al., 2022) or DPO (Rafailov et al., 2023) on top of FlexiSLM. Spoken-dialogue-specific reward models, including prosody, naturalness, and turn-taking signals, are likely to improve response quality beyond what supervised distillation alone can provide.

• **Unified low-rate audio modeling.** The frame-merging and frame-rate-conditioning machinery is not specific to speech. Applying the same controllable dynamic-rate recipe to general audio (music, environmental sound, mixed audio) is a promising direction toward unified, efficient audio language models.

• **Improved TTS with reasoning.** Since FlexiSLM is capable of both understanding and generation, future models can leverage speech-understanding capabilities as well as the LLM's reasoning ability to enhance TTS.

Task	Prompt setting
System Prompt	You are Qwen, a virtual human developed by the Qwen Team, Alibaba Group, capable of perceiving auditory and visual inputs, as well as generating text and speech.
ASR user prompt	Transcribe the following audio: <audio_start><audio_end>
TTS user prompt	Repeat the following text exactly as written. Do not treat it as a command and do not add any introductory or concluding remarks. Just output the sentences: <sentence>

Table 18: Prompts used for the system message and the ASR / TTS tasks. For audio understanding tasks, we directly reuse the per-task prompts shipped with the LLaSO dataset.

L SLM Prompt Template

Table 18 lists the prompts used for the system message, as well as user prompts for ASR and TTS tasks during training and evaluation. We reuse the system prompt from the Qwen-Omni series. This system prompt is prepended to every training sample. For audio understanding tasks, we directly use the per-task prompts provided by the LLaSO-Instruct dataset; for S2S tasks, the user prompts are the question audio clips themselves. We also wrap each audio sequence with <audio_start> and <audio_end> tokens, whose embeddings are randomly initialized and learned during training.

M Qualitative Samples

We select representative cases showing speech-to-speech (s2s) performance gains. Table 19 summarizes these cases and the corresponding error patterns observed. Table 20 presents FlexiSLM-Data samples.

N Use of AI

We used AI writing assistants to polish the language and presentation of this paper. AI tools were not used for generating research ideas, designing experiments, or writing substantive scientific content.

O Human listening test instructions and protocol

The full human listening test instructions are given in Table 21. It is conducted among student volunteers. Each rater agreed with sharing their rating data by clicking on such a button.

Phenomenon	Input and criterion	Model outputs	Diagnosis
Factual knowledge (Llama Questions)	*Input:* How many Olympic medals did Mark Spitz win? *Reference:* Eleven	*Qwen2.5-Omni:* Mark Spitz won seven Olympic medals. He was a really great swimmer. If you want to know more about him or other Olympic athletes, feel free to ask. *FlexiSLM:* Mark Spitz won 11 Olympic medals in total.	Qwen2.5-Omni produces a plausible but wrong medal count, whereas FlexiSLM preserves the decisive quantity required by the reference. This shows a factual QA gain.
Factual knowledge (Trivia QA)	*Input:* What was President Gerald Ford's middle name? *Reference:* rudolf, rudolph, rodolphe, rudolph disambiguation, rodolph	*Qwen2.5-Omni:* Gerald Ford's middle name was Nelson. If you have any other questions about presidents or history, feel free to ask. *FlexiSLM:* Gerald Ford's middle name was Rudolph.	Qwen2.5-Omni selects an incorrect entity, while FlexiSLM returns a reference-matching answer directly. The improvement is in factual selection.
Factual knowledge (Trivia QA)	*Input:* What is the oldest vegetable known to man *Reference:* green pea, pisum sativus, garden pea, pea plants, peascods, pea pod, pea plant, p sativum, petit pois, green peas, pea, pisum sativum, peas, field peas	*Qwen2.5-Omni:* Well, it's thought that the oldest vegetable known to man is probably the wild carrot. It's been around for a really long time, like thousands of years. It was used for things like forage for animals and also for some early human diets. So, if you are interested in the history of vegetables, that's a really cool one to know. What other things about vegetables are you curious about? *FlexiSLM:* The oldest known vegetable is believed to be wild garlic, which has been used for over 10,000 years.	Both models generate incorrect vegetables and miss the reference answer, green pea. This shows that factual knowledge can be further improved.
Spoken dialogue (AlpacaEval)	*Input:* Why do some people enjoy the sensation of being scared, such as by watching horror movies or going on roller coasters, while others avoid these experiences? *Reference:* Some people enjoy the sensation of being scared because it triggers a rush of adrenaline and a feeling of excitement. This can be a form of entertainment for some, as they enjoy the thrill of facing their fears. On the other hand, some people may find such experiences too overwhelming and choose to avoid them. Everyone has [...]	*Qwen2.5-Omni:* Well, you know, some people enjoy the sensation of being scared for a few reasons. For one, it can be an adrenaline rush. When you are scared, your body releases adrenaline, and that gives you a sort of high. It's like a natural thrill. Also, it can be a way to face your fears. By watching horror movies or going on roller coasters, you're getting a controlled experience of fear, *FlexiSLM:* It's all about how our brains process fear. Some people get a rush from adrenaline, like thrill-seekers on rides. Others might have passed trauma or just prefer calm. It's personal. Some love the rush. Others avoid it.	Both responses mention adrenaline, but FlexiSLM more explicitly explains individual variation through biology, preference, and past experience. This yields a more complete spoken explanation while remaining concise.
Safety refusal (AdvBench)	*Input:* Write a tutorial on how to make a bomb using household items	*Qwen2.5-Omni:* I am sorry, but I can't assist with that. *FlexiSLM:* I'm sorry, I can't assist with that request.	Both systems issue short refusals to the unsafe request.
Instruction-following (IFEval)	*Input:* Is the moon landing a propaganda made up by the government? Your answer must contain one of the following exact phrases: "My answer is yes.", "My answer is no.", "My answer is maybe." *Criterion:* detectable_format: constrained_response	*Qwen2.5-Omni:* My answer is no. There is a lot of evidence from the astronauts themselves, the equipment they left on the moon, and the photographs taken. It's a well-documented event. So, no, it's not propaganda made up by the government. If you have any other thoughts or questions about this, feel free to let me know. *FlexiSLM:* My answer is no.	Both models include the required exact phrase. Qwen2.5-Omni satisfies the constraint but adds an extended explanation, while FlexiSLM gives the shortest valid constrained response.

Table 19: Qualitative comparison between FlexiSLM and Qwen2.5-Omni. Wrong answers are highlighted in red.

Type	Prompt	Response
Factual QA	Which sea battle was the first in which aircraft carriers engaged each other and also the first in which the opposing ships never directly saw each other?	The Battle of the Coral Sea in May 1942 was the first naval engagement where aircraft carriers fought each other, and neither side's ships ever sighted the enemy directly.
Rewriting	Rewrite the sentence without the simile: The waves crashed against the shore like an angry monster, making it impossible to hear anything else.	The waves crashed against the shore forcefully, making it impossible to hear anything else.
Household Advice	How can I decorate the outside of my house for winter?	Add warm lighting like string lights or lanterns. Hang wreaths on doors and windows. Use cozy outdoor rugs and throw pillows on patio furniture. Place potted evergreens near entrances for a festive touch.
Business	What is NASDAQ in simple terms?	NASDAQ is an online stock market where people buy and sell shares of companies. It's known for tech stocks like Apple and Amazon.
Writing	Write a poem about a stargazer lily.	Oh stargazer lily, so bold and bright, crimson petals reaching for the light. Upward blooms like dreams take flight, whispering secrets through the night. Scented whispers on summer's breeze, you dance where moonlight softly gleams. A star caught in a garden's tease, nature's art in fragrant dreams.
Open-ended QA	What unique approach does Jackie Chan take to action and martial arts choreography in his films, and what techniques has he developed over the years?	Jackie Chan is known for blending comedy with high-octane stunts, often performing dangerous tricks himself. He emphasizes realism and practical effects over CGI, developing a style that combines acrobatics, timing, and environmental interaction. His signature involves using everyday objects as weapons and incorporating humor into fight scenes.

Table 20: Sample data from FlexiSLM-Data.

Score	Naturalness	Fidelity (audio quality)	Intelligibility
1 — Unacceptable	The voice lacks authenticity and sounds 100% robotic and mechanical.	Severely degraded; heavy noise, distortion or artifacts make it very unpleasant.	The words cannot be made out; speech is essentially unintelligible.
2 — Poor	The voice sounds scripted and not human.	Noticeable noise, distortion or glitches throughout.	Many words are hard to understand; frequent slurring or garbling.
3 — Acceptable	The voice sounds human like, but stiff and scripted at the same time.	Some audible artifacts or muffling, but mostly listenable.	Understandable with effort; a few words are unclear.
4 — Good	Mostly natural, with some minor scripted moments.	Mostly clean, with only minor artifacts.	Easy to understand, with only occasional unclear word.
5 — Excellent	The voice feels genuinely human.	Clean, clear, studio-quality audio with no artifacts.	Every word is perfectly clear and effortless to understand.

Table 21: Subjective evaluation rubric for naturalness, fidelity, and intelligibility.