BSLM: A Bi-Level Speech-Language Model for the Joint Modeling of Discrete and Continuous Tokens

Tianze Luo[1], **Zixin Wang**[2], **Kaizhi Qian**[3], **Yang Zhang**[3], **Chuang Gan**[2]

[1]Tsinghua University
[2]University of Massachusetts at Amherst
[3]MIT-IBM Watson AI Lab
ltz22@mails.tsinghua.edu.cn, zxwang@umass.edu, {kqian, yang.zhang2}@ibm.com, chuangg@cs.umass.edu

Abstract

Speech Language Models (SpeechLMs) are Large Language Models (LLMs) that can directly process both speech input and speech output, establishing a more natural framework for human-machine interaction. Traditional approaches employ speech encoders with vector-quantization modules to discretize continuous speech signals into tokens, allowing LLMs to unify the modeling of text and speech tokens. However, the inherent conflict between speech's continuous nature and text's discrete essence, coupled with speech data's substantially lower information density when compared to text data, poses significant challenges for these models. In this work, we propose a novel model and training methodology to enable joint generation of discrete and continuous tokens. Our autoregressive model features a bi-level whole-part architecture comprising a large transformer for long-range dependency modeling and a small diffusion transformer that generates continuous speech tokens using local information. Experimental results demonstrate that the proposed model achieves performance comparable to discrete token-based SpeechLMs while requiring fewer training tokens.

1 Introduction

The emergence of large language models (LLMs) such as GPT-4 (Achiam et al. 2023) has fundamentally transformed natural language processing through remarkable emergent capabilities, including instruction following, logical reasoning, and few-shot learning (Touvron et al. 2023; Yang et al. 2024a; Jiang et al. 2024; Jaech et al. 2024). While these text-based models demonstrate unprecedented language understanding, extending their capabilities to voice interactions remains challenging. Conventional systems typically employ a cascaded architecture where an automatic speech recognition (ASR) model first converts speech input to text, then a text-based LLM generates textual responses, and finally a text-to-speech (TTS) synthesis model converts the responses back to speech outputs. However, as noted by Défossez et al. (2024) and Zeng et al. (2024a), this modular approach suffers from inherent information loss during modality conversions, particularly in preserving prosodic features and emotional nuances. The sequential processing pipeline also introduces cumulative latency that degrades real-time interaction quality.

Recent advancements in end-to-end speech language models (SpeechLMs) aim to address these limitations through direct speech tokenization technologies. Early work by Lakhotia et al. (2021) proposed unsupervised learning on speech corpora using discrete tokens, with subsequent improvements achieved through textual LLM warm initialization (Hassid et al. 2023) and large-scale pretraining with interleaved speech-text data (Nguyen et al. 2025; Défossez et al. 2024; Zeng et al. 2024a). Contemporary SpeechLM architectures typically employ speech encoders with vector-quantization modules to discretize continuous speech signals into tokens, enabling joint generation of text and speech tokens through LLMs. These speech tokens generally fall into two categories: (1) acoustic tokens generated by neural codecs for high-fidelity audio reconstruction at low bitrates, and (2) semantic tokens extracted from features learned through self-supervised (Hsu et al. 2021; Chung et al. 2021) or supervised objectives (Du et al. 2024; Zeng et al. 2024a).

However, two critical challenges persist in these token-based approaches due to inherent modality discrepancies. First, due to speech's continuous nature, the vector quantization process not only introduces discretization errors, but also cuts off gradient information, which makes the speech encoder training more challenging than its continuous counterpart. Second, the substantially lower information density of speech signals compared to text leads to significantly more speech tokens required for equivalent semantic content. This token inflation not only reduces training efficiency but also adversely impacts model performance through extended sequence lengths.

The fundamental crux behind these challenges is the drastic difference between the sparse, continuous speech signal and the dense, discrete text. Therefore, our research question is: Can we design a speech language model that respects the continuous nature of speech, yet still fits into the mainstream discrete autoregressive text LLM architectures?

To address these challenges, we propose Bi-level Speech Language Model (BSLM), a novel autoregressive framework with unified text-speech generation capabilities. Our architecture employs a bi-level hierarchical structure: A primary LLM captures long-range cross-modal dependencies and generates text tokens, while a secondary diffusion transformer with fewer parameters produces continuous speech tokens using local LLM latents and adjacent speech context.

Additionally, our model supports grouped token input and generation mechanism, which enables efficient processing of closely-related speech tokens in clustered units. This approach significantly reduces the token count processed by the primary LLM while maintaining speech generation quality. Our method achieves simultaneous modeling of discrete text tokens and continuous speech representations, effectively bridging the information gap between speech and text modalities. Experimental results demonstrate that our model achieves competitive performance on speech language modeling tasks while requiring fewer training tokens compared to existing discrete token-based SpeechLMs, establishing a new method for unified speech-text language modeling.

2 Related Work

2.1 Multimodal Large Language Models

Recent advances in multimodal large language models (LLMs) primarily follow two paradigms. For multimodal understanding, mainstream approaches align visual and audio features with text inputs via lightweight adapters, employing pretrained encoders like Whisper (Radford et al. 2023) and BEATs (Chen et al. 2022) for audio, or CLIP-pretrained (Radford et al. 2021) vision transformers (Dosovitskiy et al. 2020) used in LLaVA (Liu et al. 2023) and BLIP-2 (Li et al. 2023) for vision tasks. However, these models remain limited to text-only outputs, as generating continuous-value images and audio falls beyond their capabilities. To enable multimodal generation, methods like SpeechGPT (Zhang et al. 2023a) and AnyGPT (Zhan et al. 2024) tokenize non-text modalities into discrete units integrated into LLM vocabularies. Despite progress, challenges persist in effectively transferring LLM knowledge across modalities to enhance generalization and instruction-following capabilities.

2.2 Speech Language Modeling

The evolution of speech language modeling has centered on discrete audio representations and hierarchical architectures. Lakhotia et al. (2021) proposed the Generative Spoken Language Modeling (GSLM) framework, consisting of three main modules: a speech tokenizer, a speech language model, and a vocoder. Subsequent studies have expanded upon this foundation. AudioLM (Borsos et al. 2023) employs dual discrete speech token representations - phonetic tokens (Chung et al. 2021) and acoustic tokens (Zeghidour et al. 2021) - to respectively model coarse and fine-grained speech information. TWIST (Hassid et al. 2023) demonstrated that while modality gaps between speech and text persist, fine-tuning a textual language model on speech data yields superior performance compared to random cold-initialization of SpeechLMs. SpeechGPT (Zhang et al. 2023a) enhanced SpeechLM capabilities through multimodal training incorporating ASR, TTS, and chain-of-modality question answering tasks. Recent developments include VoxtLM (Maiti et al. 2024) and SUTLM (Chou et al. 2023), which employ joint training on text and speech through ASR, TTS, and speech/text continuation tasks, while SpiritLM (Nguyen et al. 2025) achieves performance improvements through training on interleaved speech-text data. A critical challenge in SpeechLMs remains the excessive length of audio token sequences, which complicates long-context modeling and slows inference. Moshi (Défossez et al. 2024) and GLM-4-Voice (Zeng et al. 2024a) address this through novel 12.5Hz speech tokenizers and high-fidelity speech decoders. Moshi employs residual vector quantization (RVQ) to tokenize speech data, whereas GLM-4-Voice utilizes only a single codebook in its quantization process. Additionally, they significantly scale up data usage compared to previous research to alleviate the data lack issue in speechLMs, GLM-4-Voice adopts a text-to-token model that directly converts text into corresponding speech tokens to generate synthetic speech-text interleaved data.

2.3 Flow Matching Models

In recent years, the development of flow matching methodologies has led to significant progress in continuous-time generative modeling. Building upon the foundational framework of Flow Matching (FM) introduced by Lipman et al. (2022), which is closely related to the Rectified Flow models proposed by Liu, Gong, and Liu (2022), researchers have demonstrated its efficacy in learning Ordinary Differential Equations (ODEs) through the conditional flow matching (CFM) objective. This approach circumvents the computational complexities inherent in traditional score-based diffusion models and numerical ODE solvers for Continuous Normalizing Flows (CNFs) (Chen et al. 2018), establishing a simplified velocity field regression framework. Subsequent innovations by Tong et al. (2023) and Pooladian et al. (2023) further integrate optimal transport (OT) principles to enhance FM's performance and efficiency, enabling the construction of ODEs with minimally varying vector fields during source-to-target distribution transport and improving numerical stability. The impact spans diverse generation tasks: Esser et al. (2024) demonstrate that flow matching models equipped with powerful transformer-based velocity predictors can efficiently generate high-resolution images in text-to-image synthesis; Zheng et al. (2024) and Yang et al. (2024b) show remarkable cross-modal generalization capabilities through OpenSora and CogVideoX in text-to-video generation; while Cheng et al. (2024) and Wang et al. (2024) extend the framework to audio generation via MMAudio and Frieren for text-to-audio and video-to-audio tasks.

3 Method

3.1 Speech Encoder and Decoder

We begin by introducing our speech encoder and decoder architecture. The system leverages Whisper (Radford et al. 2023), a state-of-the-art model for automatic speech recognition (ASR) and speech translation. Our implementation utilizes the whisper-large-v3 variant, which was pretrained on 1 million hours of weakly labeled audio and 4 million hours of pseudo-labeled audio generated by whisper-large-v2. The whisper encoder processes 16kHz audio-derived mel-spectrograms with a hop size of 160, yielding an initial frame rate of 100Hz. Through its 1D convolutional layers,

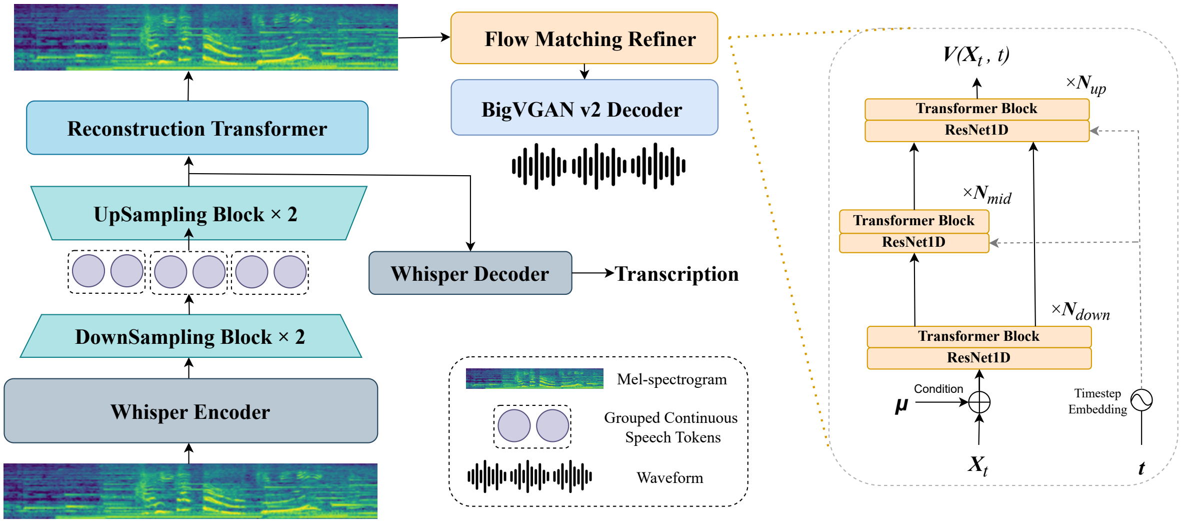

Figure 1: The structure of our speech encoder and decoder.

the encoder downsamples this input to 50Hz while producing latent features at the same rate.

The architecture then employs two downsampling blocks followed by two upsampling blocks, each with a ratio of 2, to compress the 50Hz features into 64-dimensional 12.5Hz continuous speech tokens and subsequently decompress them back to 50Hz. These processed features serve dual purposes: the whisper decoder utilizes them through cross-attention mechanisms for transcription prediction, while a dedicated reconstruction transformer generates mel-spectrograms for speech synthesis.

Our reconstruction transformer features two key enhancements: Rotary Positional Embedding (RoPE) (Su et al. 2024) and SwiGLU feed-forward networks (Shazeer 2020). To accommodate the BigVGAN vocoder (Lee et al. 2022) which requires 93.75Hz mel-spectrograms from 24kHz audio (hop size 256), the reconstruction transformer upsamples the 50Hz features to 100Hz during forward propagation and linearly interpolate them to the target 93.75Hz resolution.

The system further implements a flow matching model for mel-spectrogram refinement. Our velocity prediction network architecture concatenates coarse mel-spectrograms with noisy inputs along the frequency dimension to estimate velocity fields. Through comparative experiments with closely parameterized models, we found that UNet-based architectures (Ronneberger, Fischer, and Brox 2015) outperform diffusion transformers (Peebles and Xie 2023) in this refinement task. We hypothesize this performance advantage stems from UNet's superior handling of local features through its 1D convolutional layers, which proves more effective than the global attention mechanisms in diffusion transformers for this specific application.

We illustrate the structure of our speech encoder and decoder in Figure 1. Further architectural and implementation details are provided in Appendix A.

3.2 Bi-Level Speech Language Model

We now formally present our method using mathematical notation. For traditional discrete modeling, the conditional distribution is formulated via softmax and neural networks:

$$\begin{aligned} &P_\theta(x_T|x_1, x_2, \ldots, x_{T-1}) \\ =&\,\mathrm{softmax}(\mathbf{W}\cdot\mathbf{NN}_\theta(x_1, x_2, \ldots, x_{T-1}) + \mathbf{b}), \end{aligned} \tag{1}$$

where $\mathbf{W}, \mathbf{b}$ represent the language model head parameters and $\mathbf{NN}_\theta$ denotes the LLM. This formulation enables various sampling methods including temperature sampling, top-k sampling, and top-p sampling to be applied to the logits during inference. In our model, x_T can represent either discrete text tokens or grouped continuous speech tokens with dimensions of group-size × single-token dimensions. The model dynamically predicts whether to generate a text token or grouped speech tokens at each time step through a lightweight multi-layer perceptron (MLP) classifier that estimates the probability from latent features. For text generation, we employ the softmax function to model the distribution, whereas for speech generation, we utilize flow matching (Lipman et al. 2022) framework to model the data distribution. Formally,

$$\begin{aligned} &P_\theta(x_T \mid x_1, \ldots, x_{T-1}) \\ =&\begin{cases} P_{\text{softmax}}(x_T \mid x_1, x_2, \ldots, x_{T-1}) & \text{w. p. } \hat{p}, \\ P_{\text{FM}}(\mathbf{x}_T \mid x_1, x_2, \ldots, x_{T-1}) & \text{w. p. } 1-\hat{p}, \end{cases} \end{aligned} \tag{2}$$

where $\hat{p} = \mathrm{MLP}_\theta(\mathbf{NN}_\theta(x_1, x_2, \ldots, x_{T-1}))$ represents the predicted probability, and P_{softmax} is defined in eq. (1). We denote vector variables and matrices in boldface, while scalar variables and variables that can be either scalar or vector remain in regular format. For continuous speech tokens, the modeling becomes more intricate. As autoregressive transformers excel at capturing long-term semantic dependencies, while flow matching models efficiently generate fine-grained continuous speech with higher temporal resolution, we therefore model P_{FM} as:

$$\begin{aligned}&P_{\text{FM}}(\mathbf{x}_T \mid x_1, \ldots, x_{T-1}) = \text{ODESOLVE}(\mathcal{N}(\mathbf{0}, \mathbf{I}),\\ &\mathbf{v}_\theta(\mathbf{x}_{T-K}, \ldots, \mathbf{x}_{T-1}, \mathbf{NN}_\theta(x_1, \ldots, \mathbf{x}_{T-1}), \mathbf{x}_{T,t}, t)),\end{aligned} \tag{3}$$

where ODESOLVE denotes the process of randomly sampling noise from a standard normal distribution at $t = 0$, then solving the velocity ordinary differential equation (ODE) defined by the velocity predictor $\mathbf{v}_\theta$ until $t = 1$. Following the flow matching framework (Lipman et al. 2022), we employ a straight-line trajectory for noise addition to clean grouped speech tokens:

$$\mathbf{x}_{T,t} = t\mathbf{x}_T + (1-t)\epsilon, \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), t \in [0, 1] \tag{4}$$

This corresponds to a mean squared flow matching loss:

$$\begin{aligned}&\min_\theta \|\mathbf{v}_\theta(\mathbf{x}_{T-K}, \ldots, \mathbf{x}_{T-1},\\ &\quad \mathbf{NN}_\theta(x_1, x_2, \ldots, \mathbf{x}_{T-1}), \mathbf{x}_{T,t}, t) - (\mathbf{x}_T - \epsilon)\|_2^2\end{aligned} \tag{5}$$

The hyperparameter K determines the number of preceding speech token groups provided to the velocity network. For initial speech token groups without historical context, we apply zero-padding to these conditional variables. The velocity network does not access previous text LLM latent representations in this boundary situation. The continuous nature of speech signals ensures that adjacent preceding tokens contain valuable information for the denoising process. Here, $\mathbf{NN}_\theta(x_1, x_2, \ldots, x_{T-1})$ denotes the latent vector generated by the language model from previous variables. This architecture establishes a bi-level language model capable of joint modeling of discrete and continuous tokens. The primary language model captures long-range cross-modal dependencies and produces text tokens, while the secondary velocity network – with fewer parameters – generates continuous speech tokens using both local language model latents and neighboring speech tokens. Furthermore, the model supports grouped token processing, enabling efficient handling of correlated speech tokens in clustered units and reducing the token count processed by the language model.

In our implementation, we model $\mathbf{NN}_\theta$ using the Qwen3-4B (Yang et al. 2025) large language model, while the velocity prediction network is implemented through a diffusion transformer. We group two speech tokens while setting the number of previous speech token groups K to two. A speech adaptor constructed with two SwiGLU blocks processes the grouped speech tokens for the LLM. During the inference process, speech and text inputs are fed to the speech adaptor and text embedding layers respectively, after which the LLM produces a latent vector from the combined input. When the MLP classifier predicts a high probability of generating a text token, the logits are computed from the latent vector and standard discrete LLM sampling methods are applied for text generation. Otherwise, the secondary flow matching model is activated, generating grouped continuous speech tokens through a 40-step velocity ODE solving process. The generation process continues until an end-of-text token is produced. Figure 2 illustrates the structure of our speech language model. For detailed specifications of the model architecture, please refer to Appendix B.

3.3 Flow-DPO

The efficiency of preference optimization has been demonstrated for both LLMs and vision generative models (Ouyang et al. 2022; Rafailov et al. 2023; Liu et al. 2025). Our model can also perform direct preference optimization (DPO) training on grouped speech tokens, which represent continuous rather than discrete signals.

In traditional DPO settings for discrete domains, given pairwise preferences $\{\mathbf{y}, \mathbf{x}_w, \mathbf{x}_l\}$ where $\mathbf{x}_w \succ \mathbf{x}_l$, the objective maximizes the likelihood ratio of preferred versus dispreferred outputs under a model π_θ relative to a reference model π_{ref}. The loss function is formulated as:

$$\begin{aligned}&\mathcal{L}_{\text{DPO}} =\\ &-\mathbb{E}\left[\log\sigma\left(\beta\log\frac{\pi_\theta(\mathbf{x}_w|\mathbf{y})}{\pi_{\text{ref}}(\mathbf{x}_w|\mathbf{y})} - \beta\log\frac{\pi_\theta(\mathbf{x}_l|\mathbf{y})}{\pi_{\text{ref}}(\mathbf{x}_l|\mathbf{y})}\right)\right],\end{aligned} \tag{6}$$

where σ denotes the sigmoid function. This approach bypasses explicit reward modeling by leveraging the Bradley-Terry preference model (Bradley and Terry 1952). For continuous data like speech, however, direct density ratio estimation becomes intractable due to the high-dimensional nature of the output space. Flow-DPO (Liu et al. 2025) addresses this by reinterpreting the DPO objective through the lens of flow matching dynamics.

Based on the derivation of Diffusion-DPO (Wallace et al. 2024), Liu et al. (2025) propose the following Flow-DPO loss $\mathcal{L}_{\text{FD}}$, which directly optimizes the velocity field to satisfy preferences:

$$\begin{aligned}&-\mathbb{E}\Bigg[\log\sigma\Bigg(-\frac{\beta_t}{2}\Big(\\ &\qquad \|\mathbf{v}^w - \mathbf{v}_\theta(\mathbf{x}_t^w, t)\|^2 - \|\mathbf{v}^w - \mathbf{v}_{\text{ref}}(\mathbf{x}_t^w, t)\|^2\\ &\qquad -\big(\|\mathbf{v}^l - \mathbf{v}_\theta(\mathbf{x}_t^l, t)\|^2 - \|\mathbf{v}^l - \mathbf{v}_{\text{ref}}(\mathbf{x}_t^l, t)\|^2\big)\Big)\Bigg)\Bigg],\end{aligned} \tag{7}$$

where β_t is simply set as a constant. The expectation is taken over preference data samples ($\{\mathbf{x}_0^w, \mathbf{x}_0^l\} \sim \mathcal{D}$) and the noise schedule t. The Gaussian noise used to perturb the data samples is shared, with velocity vectors $\mathbf{v}^w$ and $\mathbf{v}^l$ are computed from clean data $\mathbf{x}_0^w, \mathbf{x}_0^l$ and the shared Gaussian noise according to flow matching principles. Since our model uses a flow matching method to generate grouped continuous speech tokens, we can adopt Flow-DPO in a similar way. The only discrepancy is that our diffusion transformer is conditioned on LLM latents and previous grouped speech tokens; we just need to incorporate these conditions into the velocity prediction network in eq. (7) to train our model.

4 Experiments

4.1 Datasets and Training Process

We collected various datasets to train our speech encoder, speech decoder and speech language model. Specifically, about 180k hours English speech data are collected from LibriSpeech (Panayotov et al. 2015), Libriheavy (Kang et al. 2024), Emilia (He et al. 2024), Peoples Speech (Galvez et al. 2021) , GigaSpeech (Chen et al. 2021), MLS (Pratap et al. 2020) and Vox Populi (Wang et al. 2021) datasets.

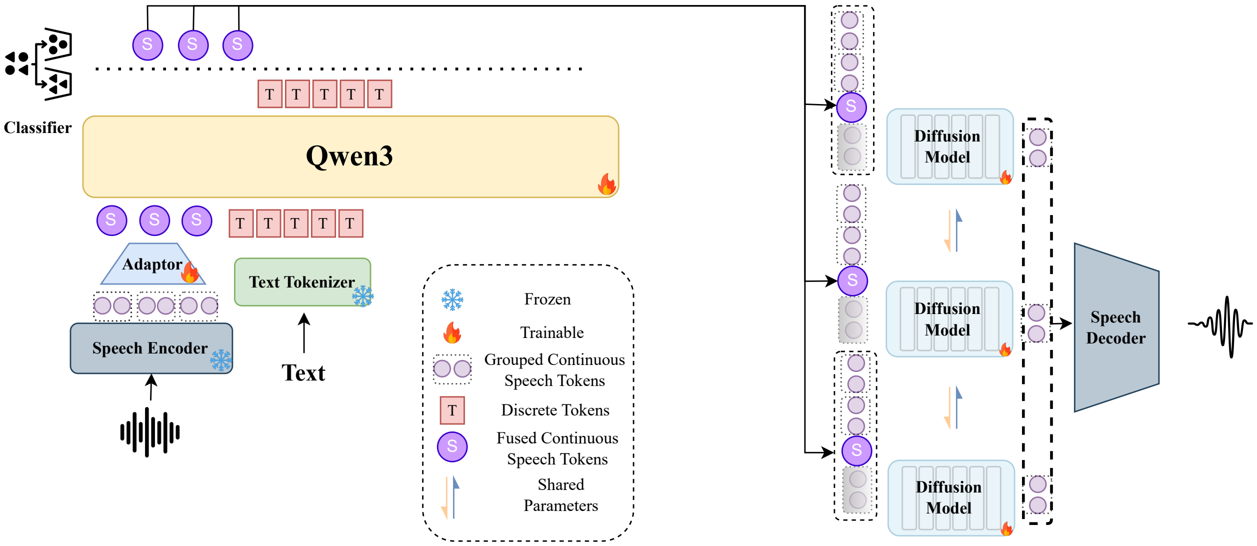

Figure 2: The structure of our Bi-Level Speech-Language Model.

The training loss function of our speech encoder and decoder combines three components: the sequence-to-sequence ASR loss from the whisper decoder, the mean square error between reconstructed and ground truth mel-spectrograms, and the mean square velocity loss of the flow matching mel-spectrogram refiner, with the latter two terms scaled by coefficients 1.5 and 5 respectively. As the whisper encoder and decoder utilize pretrained initialization while other components are randomly initialized, we employ three AdamW optimizers with betas (0.9, 0.98) for parameter updates. The first optimizer applies a constant learning rate of 1e-5 and 0.05 weight decay specifically for the whisper encoder-decoder parameters. The second optimizer manages the upsampling blocks, downsampling blocks, and reconstruction transformer with a cosine-decayed learning rate starting at 2e-4 and decaying to 2e-5 alongside 0.025 weight decay. The third optimizer exclusively handles the velocity network parameters using a cosine-decayed learning rate initialized at 4e-4 and decaying to 2e-5 with 0.025 weight decay. Our model is trained with a batch size of 128 for 250k steps on 8 NVIDIA H100 GPUs (80G memory each), after which we freeze all components except the flow matching mel-spectrogram refiner and continue training it for an additional 100k steps to improve performance.

For SpeechLM training, we expand synthetic data using KokoroTTS (Hexgrad 2025). Specifically, we generate 40k hours of speech from TinyStories (Eldan and Li 2023) and an additional 100k hours of longer-duration speech using FineWeb-Edu (HuggingFaceFW 2024). Unlike GLM-4-Voice (Zeng et al. 2024a), which synthesizes speech tokens via Poisson-distributed spans ($\lambda = 10$) with text ratio $\eta = 0.3$, our approach leverages KokoroTTS's sentence-level timestamps from concatenated short audio segments. We directly replace random text segments with 30% probability in FineWeb-Edu's 100k-hour synthetic speech to produce 10B interleaved speech-text pairs. For speech-only data, combining encoder/decoder training data and TinyStories synthesis yields 220k hours audios. Text data comprises 50B tokens from FineWeb-Edu, excluding texts used for interleaved synthesis. Following (Nguyen et al. 2025), we balance all three data types during training. We train the SpeechLM for 50B tokens, which is corresponding to about 71B original text and speech tokens since we group two speech tokens together.

For DPO dataset construction, we generate positive and negative samples with the SWAG dataset (Zellers et al. 2018), synthesizing a corpus totaling under 400 hours. In SWAG dataset, each initial sentence is followed by one correct and three incorrect continuations, we synthesize speech DPO samples as follows: positive samples from correct continuations and negative samples from incorrect ones. Using KokoroTTS (Hexgrad 2025), we collect three data configurations: 1) both sentences in speech modality; 2) speech-text interleaved sequences starting with speech; and 3) interleaved sequences starting with text. During training, DPO loss computation focuses on second-sentence predictions conditioned on the first sentence. We apply discrete DPO loss for text continuations and flow-DPO loss for speech continuations. These three data types are uniformly sampled during training.

We train the SpeechLM on 8 NVIDIA H100 GPUs using a 256 batch size with AdamW optimization. The initial learning rate is 5e-4, decayed to 5e-5 via cosine scheduler. For the first 5% of training tokens, we freeze the LLM while updating only the speech adaptor and diffusion transformer; for the subsequent 5% tokens, we update exclusively the LLM. Experimental results indicate that this phased adaptation notably improves training efficiency. Finally, the flow-DPO phase maintains a 5e-5 learning rate with training limited to 30 million tokens.

Model	Frame Rate	WER(↓)	VisQOL(↑)	MOSNet(↑)
Ground Truth	–	4.62	–	3.27
RVQGAN	75Hz	–	1.74	2.74
SemantiCodec	50Hz	–	2.43	3.12
SpeechTokenizer*	50Hz	9.97	1.53	2.67
SpeechTokenizer	50Hz	6.32	3.07	3.10
Spirit-Base	25Hz	11.66	–	–
Spirit-Expressive	38.5Hz	10.60	–	–
Moshi (Mimi)	12.5Hz	8.36	2.82	2.89
GLM-4-Voice	12.5Hz	8.43	2.52	**3.39**
Ours - UNet	12.5Hz	**6.13**	**3.08**	3.14
Ours - Transformer	12.5Hz	6.95	2.81	3.06

Table 1: Speech Reconstruction Results: We evaluate our models' content preservation ability and speech reconstruction quality using Word Error Rate (WER), VisQOL (Hines et al. 2015), and MOSNet (Lo et al. 2019), respectively. Baseline results are derived from (Zeng et al. 2024a,b; Défossez et al. 2024).*SpeechTokenizer is a lower-bitrate version with only 3 RVQ levels tested in Moshi (Défossez et al. 2024).

4.2 Evaluation Metrics

Following Défossez et al. (2024); Zeng et al. (2024a), we conduct comprehensive evaluations of our speech encoder and decoder's content preservation capability and reconstruction quality using the LibriSpeech dataset. To assess content preservation, we calculate the Word Error Rate (WER) by comparing ground truth transcripts with automatic speech recognition outputs generated through the ASR model from Nguyen et al. (2023). For evaluating reconstruction quality, we employ two complementary metrics: the VisQOL score (Hines et al. 2015) for perceptual similarity assessment between original and reconstructed audio, along with MOSNet (Lo et al. 2019) for predicting mean opinion scores of reconstructed audio quality.

Regarding our speech language model evaluation, we adopt four established benchmarks following Nguyen et al. (2025): sWUGGY, sBLIMP, Topic-StoryCloze, and StoryCloze. These tasks systematically evaluate language modeling capabilities through contrastive likelihood assessments, where models must distinguish correct continuations from distractors. Specifically, sWUGGY (Nguyen et al. 2020) probes lexical knowledge through nonce word paradigms, while sBLIMP examines grammatical understanding through minimal syntactic pairs. For narrative comprehension evaluation, we utilize spoken adaptations of StoryCloze and Topic-StoryCloze as described in Hassid et al. (2023); Nguyen et al. (2025). The StoryCloze benchmark tests high-level semantic reasoning by requiring models to identify coherent story continuations after processing narrative beginnings, with three multimodal evaluation settings: speech-to-speech continuation (S), text-to-speech continuation (T→S), and speech-to-text continuation (S→T). As for the Topic-StoryCloze task, negative suffixes are sampled from distinct semantic categories to evaluate the model's capacity for holistic semantic comprehension.

During evaluation, we applied token count normalization to the log-likelihood following Nguyen et al. (2025). Besides, we need to emphasize that while the log-likelihood of continuous speech tokens can be estimated via the instantaneous change of variables formula proposed by Chen et al. (2018), the continuous log-likelihood (derived from probability density functions) and discrete log-likelihood (calculated through probability values) are not directly comparable. Therefore, perplexity-based evaluation metrics are excluded from our experiments. For further details regarding the instantaneous change of variables formula and continuous log-likelihood estimation, please refer to Appendix C.

4.3 Speech Encoder and Decoder

We evaluate our speech encoder and decoder with two different kinds of mel-spectrogram refining networks on the LibriSpeech dataset (Panayotov et al. 2015). Experimental results show that UNet performs better than the diffusion transformer for our model, which can be attributed to the fact that given a coarse reconstructed mel-spectrogram, the refining task requires more local rather than global processing. The UNet with 1D convolutional blocks proves more suitable for this task.

When comparing with other discrete token-based models, we consider RVQGAN (Kumar et al. 2023), SemanticCodec (Liu et al. 2024), Speech Tokenizer (Zhang et al. 2023b), and the speech tokenizers in SpiritLM (Nguyen et al. 2025), Moshi (Défossez et al. 2024), and GLM-4-Voice (Zeng et al. 2024a). Experimental results demonstrate that our speech encoder and decoder can effectively retain semantic information in speech signals while maintaining reconstruction quality comparable to advanced discrete tokenizers and their corresponding decoders. Notably, while GLM-4-Voice achieves a high MOSNet score of 3.39, the ground truth MOSNet score is only 3.27. This discrepancy suggests its tokenizer and speech decoder may not faithfully reconstruct input speech, a conclusion supported by its lower VisQOL score of 2.52. The VisQOL score is computed using both ground truth and reconstructed speech, whereas MOSNet only predicts the mean opinion score for reconstructed audio. Additionally, the training process for our continuous speech encoder and decoder proves simpler

Model	WUGGY(↑)	BLIMP(↑)	Topic-StoryCloze(↑)			StoryCloze(↑)		
	S	S	S	T→S	S→T	S	T→S	S→T
GSLM	64.8	54.2	66.6	∅	∅	53.3	∅	∅
VoxtLM	66.1	57.1	–	–	–	–	–	–
TWIST	73.9	59.0	76.4	∅	∅	55.4	∅	∅
SpiritLM Base	69.0	58.3	82.9	72.7	88.6	61.0	59.5	64.6
SpiritLM Expr.	65.0	54.2	75.4	61.6	73.2	56.9	54.6	58.8
Moshi	72.6	58.8	83.0	–	–	60.8	–	–
GLM-4-Voice	–	–	82.9	**85.0**	93.6	**62.4**	**63.3**	**76.3**
BSLM	**74.1**	**60.2**	**84.1**	81.1	**93.8**	61.3	60.9	73.0

Table 2: Speech language modeling results. Where - denotes scores that are not publicly accessible, and ∅ represents tasks that are not supported by the corresponding model.

Models	GSLM	VoxtLM	TWIST	SpiritLM	Moshi	GLM-4-Voice	BSLM
Parameters	100M	1.3B	7B	7B	7B	**9B**	4B
Training Tokens	1B	–	36B	100B	∼720B	**∼1T**	50B*

Table 3: Model configurations. Training token quantities are estimated based on available data. *Note that our model clusters adjacent speech tokens to reduce token counts for the LLM. When calculated at the original 12.5Hz sampling rate, the total reaches approximately 71B text and speech tokens.

than their vector-quantized counterparts, as we eliminate the need for RVQ methods, exponential-moving average maintenance for codebooks, or commitment loss constraints on codebook vectors.

4.4 Bi-Level Speech Language Model

For evaluating our model's speech language modeling capabilities, we select several baseline models including GSLM (Lakhotia et al. 2021), VoxtLM (Maiti et al. 2024), TWIST-7B (Hassid et al. 2023), SpiritLM (Nguyen et al. 2025), Moshi (Défossez et al. 2024), and GLM-4-Voice (Zeng et al. 2024a). Experimental results demonstrate that our model achieves competitive performance across four evaluation metrics despite using significantly fewer training tokens, thereby validating the effectiveness of both our joint modeling architecture and token grouping methodology.

However, our findings simultaneously reveal the continued importance of model and dataset scaling for optimal performance. Notably, GLM-4-Voice – with 9B parameters and training on approximately 1T tokens – maintains higher performance on several metrics, particularly the StoryCloze benchmark, even as our method demonstrates improved learning efficiency. Current computational resource constraints prevent us from pursuing large-scale training experiments, leaving this direction for future research.

4.5 Ablation Study

To validate the effectiveness of our design and hyperparameter selections, we conducted comprehensive ablation studies constrained by computational resources. Given these limitations, we determined critical hyperparameters (group size and number of previous groups fed to the diffusion transformer) using a smaller 0.6B model that requires less training overhead. Our key findings reveal:

1. Synthetic speech-text interleaving generation using Poisson-distributed span lengths ($\lambda = 10$) with total length ratio $\eta = 0.3$ (following GLM-4-Voice (Zeng et al. 2024a)) underperforms our sentence-level segmentation approach for our model. We hypothesize this degradation stems from the Poisson strategy creating numerous short segments without clear modality transition signals, whereas our sentence-boundary interleaving preserves natural semantic continuity through line breaks.
2. Both group size and historical group count exhibit non-monotonic impacts on model performance, with optimal performance achieved at moderate values. This phenomenon can be explained through dual mechanisms: (a) While larger group sizes expose the LLM to longer speech token sequences during training, excessive sizes impair temporal resolution sensitivity; (b) Providing more historical groups facilitates generation continuity but accelerates error accumulation from autoregressive predictions in previous steps.
3. Following established practices (Xie and Wu 2024a,b), our implementation benefits significantly from separate pretraining of the speech adaptor, diffusion transformer, and LLM components prior to joint end-to-end optimization. This phased training strategy is more stable and efficient compared to pure joint training approaches.

5 Conclusion

In this study, we introduce a novel approach to address the challenges in unified speech-text language modeling by proposing a novel architecture that bridges the modality gap

Model	WUGGY(↑)	BLIMP(↑)	Topic-StoryCloze(↑)			StoryCloze(↑)		
	S	S	S	T→S	S→T	S	T→S	S→T
BSLM	**74.1**	**60.2**	**84.1**	**81.1**	**93.8**	**61.3**	**60.9**	**73.0**
Poisson Data Interleaving	72.9	58.6	82.7	80.2	92.9	60.8	60.3	72.7
BSLM-0.6B	65.6	57.6	75.9	70.7	80.3	56.2	55.8	60.5
Group Size = 1	65.1	57.4	73.7	69.8	79.5	55.9	55.6	60.2
Group Size = 3	64.7	57.1	73.9	70.0	79.3	56.3	55.4	60.0
Previous = 1	65.2	57.0	75.2	70.3	79.9	55.6	55.3	60.1
Previous = 3	65.4	57.5	75.5	70.8	80.1	56.0	55.7	60.5
No Adapting Training Stage	64.3	56.9	73.3	69.2	79.3	55.4	54.9	59.7

Table 4: Ablation study results for 4B model and 0.6B model.

between discrete text tokens and continuous speech representations. Our bi-level hierarchical framework combines the strengths of autoregressive transformers for global semantic modeling and diffusion transformers for local continuous token generation, effectively resolving the conflict between speech's continuous nature and text's discrete essence. By employing grouped token processing and localized speech generation through the diffusion component, our model achieves improvements in training efficiency while maintaining competitive performance compared to conventional discrete token-based SpeechLMs. Experimental results demonstrate that our model achieves comparable performance to discrete token-based SpeechLMs with fewer training tokens, while the proposed architecture effectively reduces token sequence lengths through a speech token grouping mechanism without compromising output quality.

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Borsos, Z.; Marinier, R.; Vincent, D.; Kharitonov, E.; Pietquin, O.; Sharifi, M.; Roblek, D.; Teboul, O.; Grangier, D.; Tagliasacchi, M.; et al. 2023. Audiolm: a language modeling approach to audio generation. *IEEE/ACM transactions on audio, speech, and language processing*, 31: 2523–2533.

Bradley, R. A.; and Terry, M. E. 1952. Rank analysis of incomplete block designs: I. The method of paired comparisons. *Biometrika*, 39(3/4): 324–345.

Chen, G.; Chai, S.; Wang, G.; Du, J.; Zhang, W.-Q.; Weng, C.; Su, D.; Povey, D.; Trmal, J.; Zhang, J.; et al. 2021. Gigaspeech: An evolving, multi-domain asr corpus with 10,000 hours of transcribed audio. *arXiv preprint arXiv:2106.06909*.

Chen, R. T.; Rubanova, Y.; Bettencourt, J.; and Duvenaud, D. K. 2018. Neural ordinary differential equations. *Advances in neural information processing systems*, 31.

Chen, S.; Wu, Y.; Wang, C.; Liu, S.; Tompkins, D.; Chen, Z.; and Wei, F. 2022. Beats: Audio pre-training with acoustic tokenizers. *arXiv preprint arXiv:2212.09058*.

Cheng, H. K.; Ishii, M.; Hayakawa, A.; Shibuya, T.; Schwing, A.; and Mitsufuji, Y. 2024. Taming multimodal joint training for high-quality video-to-audio synthesis. *arXiv preprint arXiv:2412.15322*.

Chou, J.-C.; Chien, C.-M.; Hsu, W.-N.; Livescu, K.; Babu, A.; Conneau, A.; Baevski, A.; and Auli, M. 2023. Toward joint language modeling for speech units and text. *arXiv preprint arXiv:2310.08715*.

Chung, Y.-A.; Zhang, Y.; Han, W.; Chiu, C.-C.; Qin, J.; Pang, R.; and Wu, Y. 2021. W2v-bert: Combining contrastive learning and masked language modeling for self-supervised speech pre-training. In *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 244–250. IEEE.

Défossez, A.; Mazaré, L.; Orsini, M.; Royer, A.; Pérez, P.; Jégou, H.; Grave, E.; and Zeghidour, N. 2024. Moshi: a speech-text foundation model for real-time dialogue. *arXiv preprint arXiv:2410.00037*.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Du, Z.; Chen, Q.; Zhang, S.; Hu, K.; Lu, H.; Yang, Y.; Hu, H.; Zheng, S.; Gu, Y.; Ma, Z.; et al. 2024. Cosyvoice: A scalable multilingual zero-shot text-to-speech synthesizer based on supervised semantic tokens. *arXiv preprint arXiv:2407.05407*.

Eldan, R.; and Li, Y. 2023. Tinystories: How small can language models be and still speak coherent english? *arXiv preprint arXiv:2305.07759*.

Esser, P.; Kulal, S.; Blattmann, A.; Entezari, R.; Müller, J.; Saini, H.; Levi, Y.; Lorenz, D.; Sauer, A.; Boesel, F.; et al. 2024. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*.

Galvez, D.; Diamos, G.; Ciro, J.; Cerón, J. F.; Achorn, K.; Gopi, A.; Kanter, D.; Lam, M.; Mazumder, M.; and Reddi, V. J. 2021. The people's speech: A large-scale diverse english speech recognition dataset for commercial usage. *arXiv preprint arXiv:2111.09344*.

Hassid, M.; Remez, T.; Nguyen, T. A.; Gat, I.; Conneau, A.; Kreuk, F.; Copet, J.; Defossez, A.; Synnaeve, G.; Dupoux, E.; et al. 2023. Textually pretrained speech language models. *Advances in Neural Information Processing Systems*, 36: 63483–63501.

He, H.; Shang, Z.; Wang, C.; Li, X.; Gu, Y.; Hua, H.; Liu, L.; Yang, C.; Li, J.; Shi, P.; et al. 2024. Emilia: An extensive, multilingual, and diverse speech dataset for large-scale speech generation. In *2024 IEEE Spoken Language Technology Workshop (SLT)*, 885–890. IEEE.

Hexgrad. 2025. Kokoro-82M (Revision d8b4fc7).

Hines, A.; Skoglund, J.; Kokaram, A. C.; and Harte, N. 2015. ViSQOL: an objective speech quality model. *EURASIP Journal on Audio, Speech, and Music Processing*, 2015: 1–18.

Hsu, W.-N.; Bolte, B.; Tsai, Y.-H. H.; Lakhotia, K.; Salakhutdinov, R.; and Mohamed, A. 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.

HuggingFaceFW. 2024. fineweb-edu (Revision 22b0aca).

Hutchinson, M. 1990. A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines. *Communications in Statistics-Simulation and Computation*, 19(2): 433–450.

Jaech, A.; Kalai, A.; Lerer, A.; Richardson, A.; El-Kishky, A.; Low, A.; Helyar, A.; Madry, A.; Beutel, A.; Carney, A.; et al. 2024. Openai o1 system card. *arXiv preprint arXiv:2412.16720*.

Jiang, A. Q.; Sablayrolles, A.; Roux, A.; Mensch, A.; Savary, B.; Bamford, C.; Chaplot, D. S.; Casas, D. d. l.; Hanna, E. B.; Bressand, F.; et al. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Kang, W.; Yang, X.; Yao, Z.; Kuang, F.; Yang, Y.; Guo, L.; Lin, L.; and Povey, D. 2024. Libriheavy: A 50,000 hours ASR corpus with punctuation casing and context. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 10991–10995. IEEE.

Kumar, R.; Seetharaman, P.; Luebs, A.; Kumar, I.; and Kumar, K. 2023. High-fidelity audio compression with improved rvqgan. *Advances in Neural Information Processing Systems*, 36: 27980–27993.

Lakhotia, K.; Kharitonov, E.; Hsu, W.-N.; Adi, Y.; Polyak, A.; Bolte, B.; Nguyen, T.-A.; Copet, J.; Baevski, A.; Mohamed, A.; et al. 2021. On generative spoken language modeling from raw audio. *Transactions of the Association for Computational Linguistics*, 9: 1336–1354.

Lee, S.-g.; Ping, W.; Ginsburg, B.; Catanzaro, B.; and Yoon, S. 2022. Bigvgan: A universal neural vocoder with large-scale training. *arXiv preprint arXiv:2206.04658*.

Li, J.; Li, D.; Savarese, S.; and Hoi, S. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, 19730–19742. PMLR.

Lipman, Y.; Chen, R. T.; Ben-Hamu, H.; Nickel, M.; and Le, M. 2022. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*.

Liu, H.; Li, C.; Wu, Q.; and Lee, Y. J. 2023. Visual instruction tuning. *Advances in neural information processing systems*, 36: 34892–34916.

Liu, H.; Xu, X.; Yuan, Y.; Wu, M.; Wang, W.; and Plumbley, M. D. 2024. Semanticodec: An ultra low bitrate semantic audio codec for general sound. *IEEE Journal of Selected Topics in Signal Processing*.

Liu, J.; Liu, G.; Liang, J.; Yuan, Z.; Liu, X.; Zheng, M.; Wu, X.; Wang, Q.; Qin, W.; Xia, M.; et al. 2025. Improving Video Generation with Human Feedback. *arXiv preprint arXiv:2501.13918*.

Liu, X.; Gong, C.; and Liu, Q. 2022. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*.

Lo, C.-C.; Fu, S.-W.; Huang, W.-C.; Wang, X.; Yamagishi, J.; Tsao, Y.; and Wang, H.-M. 2019. Mosnet: Deep learning based objective assessment for voice conversion. *arXiv preprint arXiv:1904.08352*.

Maiti, S.; Peng, Y.; Choi, S.; Jung, J.-w.; Chang, X.; and Watanabe, S. 2024. Voxtlm: Unified decoder-only models for consolidating speech recognition, synthesis and speech, text continuation tasks. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 13326–13330. IEEE.

Nguyen, T. A.; de Seyssel, M.; Rozé, P.; Rivière, M.; Kharitonov, E.; Baevski, A.; Dunbar, E.; and Dupoux, E. 2020. The zero resource speech benchmark 2021: Metrics and baselines for unsupervised spoken language modeling. *arXiv preprint arXiv:2011.11588*.

Nguyen, T. A.; Hsu, W.-N.; d'Avirro, A.; Shi, B.; Gat, I.; Fazel-Zarani, M.; Remez, T.; Copet, J.; Synnaeve, G.; Hassid, M.; et al. 2023. Expresso: A benchmark and analysis of discrete expressive speech resynthesis. *arXiv preprint arXiv:2308.05725*.

Nguyen, T. A.; Muller, B.; Yu, B.; Costa-Jussa, M. R.; Elbayad, M.; Popuri, S.; Ropers, C.; Duquenne, P.-A.; Algayres, R.; Mavlyutov, R.; et al. 2025. Spirit-lm: Interleaved spoken and written language model. *Transactions of the Association for Computational Linguistics*, 13: 30–52.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35: 27730–27744.

Panayotov, V.; Chen, G.; Povey, D.; and Khudanpur, S. 2015. Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, 5206–5210. IEEE.

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

Pooladian, A.-A.; Ben-Hamu, H.; Domingo-Enrich, C.; Amos, B.; Lipman, Y.; and Chen, R. T. 2023. Multisample flow matching: Straightening flows with minibatch couplings. *arXiv preprint arXiv:2304.14772*.

Pratap, V.; Xu, Q.; Sriram, A.; Synnaeve, G.; and Collobert, R. 2020. Mls: A large-scale multilingual dataset for speech research. *arXiv preprint arXiv:2012.03411*.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, 8748–8763. PmLR.

Radford, A.; Kim, J. W.; Xu, T.; Brockman, G.; McLeavey, C.; and Sutskever, I. 2023. Robust speech recognition via large-scale weak supervision. In *International conference on machine learning*, 28492–28518. PMLR.

Rafailov, R.; Sharma, A.; Mitchell, E.; Manning, C. D.; Ermon, S.; and Finn, C. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36: 53728–53741.

Ronneberger, O.; Fischer, P.; and Brox, T. 2015. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, 234–241. Springer.

Shazeer, N. 2020. Glu variants improve transformer. *arXiv preprint arXiv:2002.05202*.

Su, J.; Ahmed, M.; Lu, Y.; Pan, S.; Bo, W.; and Liu, Y. 2024. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568: 127063.

Tong, A.; Fatras, K.; Malkin, N.; Huguet, G.; Zhang, Y.; Rector-Brooks, J.; Wolf, G.; and Bengio, Y. 2023. Improving and generalizing flow-based generative models with minibatch optimal transport. *arXiv preprint arXiv:2302.00482*.

Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Wallace, B.; Dang, M.; Rafailov, R.; Zhou, L.; Lou, A.; Purushwalkam, S.; Ermon, S.; Xiong, C.; Joty, S.; and Naik, N. 2024. Diffusion model alignment using direct preference optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8228–8238.

Wang, C.; Riviere, M.; Lee, A.; Wu, A.; Talnikar, C.; Haziza, D.; Williamson, M.; Pino, J.; and Dupoux, E. 2021. VoxPopuli: A large-scale multilingual speech corpus for representation learning, semi-supervised learning and interpretation. *arXiv preprint arXiv:2101.00390*.

Wang, Y.; Guo, W.; Huang, R.; Huang, J.; Wang, Z.; You, F.; Li, R.; and Zhao, Z. 2024. Frieren: Efficient video-to-audio generation network with rectified flow matching. *Advances in Neural Information Processing Systems*, 37: 128118–128138.

Xie, Z.; and Wu, C. 2024a. Mini-omni: Language models can hear, talk while thinking in streaming. *arXiv preprint arXiv:2408.16725*.

Xie, Z.; and Wu, C. 2024b. Mini-omni2: Towards open-source gpt-4o with vision, speech and duplex capabilities. *arXiv preprint arXiv:2410.11190*.

Yang, A.; Li, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Gao, C.; Huang, C.; Lv, C.; Zheng, C.; Liu, D.; Zhou, F.; Huang, F.; Hu, F.; Ge, H.; Wei, H.; Lin, H.; Tang, J.; Yang, J.; Tu, J.; Zhang, J.; Yang, J.; Yang, J.; Zhou, J.; Zhou, J.; Lin, J.; Dang, K.; Bao, K.; Yang, K.; Yu, L.; Deng, L.; Li, M.; Xue, M.; Li, M.; Zhang, P.; Wang, P.; Zhu, Q.; Men, R.; Gao, R.; Liu, S.; Luo, S.; Li, T.; Tang, T.; Yin, W.; Ren, X.; Wang, X.; Zhang, X.; Ren, X.; Fan, Y.; Su, Y.; Zhang, Y.; Zhang, Y.; Wan, Y.; Liu, Y.; Wang, Z.; Cui, Z.; Zhang, Z.; Zhou, Z.; and Qiu, Z. 2025. Qwen3 Technical Report. arXiv:2505.09388.

Yang, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Li, C.; Liu, D.; Huang, F.; Wei, H.; et al. 2024a. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

Yang, Z.; Teng, J.; Zheng, W.; Ding, M.; Huang, S.; Xu, J.; Yang, Y.; Hong, W.; Zhang, X.; Feng, G.; et al. 2024b. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*.

Zeghidour, N.; Luebs, A.; Omran, A.; Skoglund, J.; and Tagliasacchi, M. 2021. Soundstream: An end-to-end neural audio codec. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30: 495–507.

Zellers, R.; Bisk, Y.; Schwartz, R.; and Choi, Y. 2018. Swag: A large-scale adversarial dataset for grounded commonsense inference. *arXiv preprint arXiv:1808.05326*.

Zeng, A.; Du, Z.; Liu, M.; Wang, K.; Jiang, S.; Zhao, L.; Dong, Y.; and Tang, J. 2024a. Glm-4-voice: Towards intelligent and human-like end-to-end spoken chatbot. *arXiv preprint arXiv:2412.02612*.

Zeng, A.; Du, Z.; Liu, M.; Zhang, L.; Jiang, S.; Dong, Y.; and Tang, J. 2024b. Scaling speech-text pre-training with synthetic interleaved data. *arXiv preprint arXiv:2411.17607*.

Zhan, J.; Dai, J.; Ye, J.; Zhou, Y.; Zhang, D.; Liu, Z.; Zhang, X.; Yuan, R.; Zhang, G.; Li, L.; et al. 2024. Anygpt: Unified multimodal llm with discrete sequence modeling. *arXiv preprint arXiv:2402.12226*.

Zhang, D.; Li, S.; Zhang, X.; Zhan, J.; Wang, P.; Zhou, Y.; and Qiu, X. 2023a. Speechgpt: Empowering large language models with intrinsic cross-modal conversational abilities. *arXiv preprint arXiv:2305.11000*.

Zhang, X.; Zhang, D.; Li, S.; Zhou, Y.; and Qiu, X. 2023b. Speechtokenizer: Unified speech tokenizer for speech large language models. *arXiv preprint arXiv:2308.16692*.

Zheng, Z.; Peng, X.; Yang, T.; Shen, C.; Li, S.; Liu, H.; Zhou, Y.; Li, T.; and You, Y. 2024. Open-sora: Democratizing efficient video production for all. *arXiv preprint arXiv:2412.20404*.

Supplementary Appendix

A Speech Encoder and Decoder

Here we present the architectural details of our speech encoder and decoder components.

For the Whisper encoder and decoder (Radford et al. 2023), we retain the original configuration from whisper-large-v3. The downsampling block consists of a 1D convolutional layer with stride parameter 2 for 2× downsampling, followed by two transformer blocks incorporating Rotary Position Embedding (RoPE) (Su et al. 2024) and SwiGLU blocks with an MLP ratio of $\frac{8}{3}$. The upsampling block first applies linear interpolation for feature expansion, then processes the upsampled tensor through two identical transformer blocks for refinement. Between the downsampling and upsampling blocks, a linear layer projects latent vectors into 64-dimensional continuous speech tokens at 12.5Hz, followed by another linear layer that converts these tokens back into higher-dimensional latent vectors. Both downsampling and upsampling blocks employ transformer layers with 784-dimensional hidden states and 6 attention heads. The total parameter count for these components reaches approximately 63M.

The reconstruction transformer shares the architectural structure of the transformer blocks used in downsampling/upsampling operations, but with modified hyperparameters: a hidden dimension of 1152 and 9 attention heads. Building upon Whisper's encoder architecture, which achieves 2× downsampling of mel-spectrograms through convolutional layers prior to transformer processing, our reconstruction transformer first processes 50Hz latent representations through 20 initial layers. Subsequently, we perform 2× upsampling via linear interpolation, followed by 4 subsequent transformer blocks that refine the resulting 100Hz latents. The final processing stage applies linear interpolation with a 0.9375 scaling factor coupled with a linear projection layer, ultimately producing 93.75Hz mel-spectrograms with 100 frequency bands for waveform synthesis using the BigVGAN vocoder. The reconstruction transformer has 386M parameters in total.

In our investigation of velocity networks for flow matching refinement, we conducted comparative studies between two model architectures with approximately 32M parameters each. The first architecture employs a 12-layer diffusion transformer (Peebles and Xie 2023), where each layer features: (1) a 384-dimensional hidden state with 6 attention heads, (2) Rotary Position Embedding (RoPE) (Su et al. 2024), and SwiGLU blocks using an MLP ratio of $\frac{8}{3}$. The second architecture implements a modified UNet (Ronneberger, Fischer, and Brox 2015) with symmetric structure: 3 blocks in both downsampling/upsampling pathways and 4 blocks in the middle section. The middle section implements feature compression through 2× downsampling. Each block contains two sequential components: (1) one 1D convolutional residual block (512 channels, kernel size 3, GELU activation), followed by (2) an attention mechanism for global feature integration, configured with a hidden dimension of 512 and 8 attention heads. The time embedding is added to the latents along frequency dimension in the residual blocks.

During training, we process 30-second audio clips corresponding to 3000-frame mel-spectrograms at 100Hz. For clips shorter than this duration, we apply zero-padding at the audio's end. Crucially, all loss computations are restricted to non-padded regions of the spectrograms.

B Speech Language Model

Now we provide more information about our speech language model.

For the configuration of the LLM, we follow Qwen3 (Yang et al. 2025) without modifications. Besides, we set both the group size and the number of previous token groups provided to the flow matching network to 2. Regarding the speech adaptor, it comprises two SwiGLU blocks with an MLP ratio of $\frac{8}{3}$, where the input, intermediate, and output dimensions are 128, 1152, and 2560 respectively. The total parameter count amounts to approximately 19M.

The secondary flow matching model implements its velocity network through a 4-layer diffusion transformer. This architecture employs a hidden dimension of 1152 with 9 attention heads, incorporating Rotary Position Embedding (RoPE) (Su et al. 2024) and SwiGLU blocks with an MLP ratio of $\frac{8}{3}$. Following standard diffusion transformer designs (Peebles and Xie 2023), we implement time embedding injection through adaptive layer normalization and scaling layers. The complete diffusion transformer comprises approximately 103M parameters, which is much smaller than the LLM and only consumes less than 10% time during inference. Notably, we employ distinct linear projection layers for three input components: previously grouped speech tokens, LLM latents, and noisy latents. These components are concatenated in the specified sequence before being fed into the transformer.

Noting that speech tokens are continuous 64-dimensional vectors while text tokens are discrete indices, we design a specialized method to enable parallel processing. Our speech language model receives three input components: a speech tensor, a text tensor, and their corresponding positional tensors. Although adjacent elements within speech or text tensors may not correspond to neighboring positions in the interleaved speech-text sequence, we strategically concatenate these elements to enhance parallel processing efficiency. The continuous speech tensor is projected to the LLM's latent dimension through a linear layer, while the discrete text tensor is fed into the embedding layer. Utilizing the positional tensors for speech and text, we employ PyTorch's scatter function to generate interleaved speech-text latent representations for LLM input, and then they are transformed into latent features of the LLM. During training, based on predefined data types for each position, we compute flow matching loss with the secondary diffusion transformer for speech-target positions and standard cross-entropy loss for text-target positions. Additionally, we incorporate a cross-entropy loss from the MLP classifier (which determines output modality of our model) into the previous two loss functions, scaled by a coefficient of 0.1.

C Evaluation

For discrete text data, the log-likelihood of individual data samples can be readily computed in our Speech LM implementation. In contrast, when handling continuous speech data, we employ the instantaneous change of variables formula from NeuralODE (Chen et al. 2018) to enable log-likelihood computation with our flow matching model. This mathematical foundation demonstrates how we can effectively calculate likelihoods for continuous speech representations through differential equation-based transformations.

Theorem 1 (Instantaneous Change of Variables) Let $\mathbf{x}_t$ be a finite continuous random variable with probability density function $p(\mathbf{x}_t, t)$ dependent on time t, where the temporal evolution of probability density is governed by the flow-matching velocity ODE $\frac{d\mathbf{x_t}}{dt} = \mathbf{v}(\mathbf{x}_t, t)$. Assuming $\mathbf{v}$ is uniformly Lipschitz in $\mathbf{x}$ and continuous in t, then the probability density function and velocity field satisfy the following differential equation:

$$\frac{\partial \log p(\mathbf{x}_t, t)}{\partial t} = -\mathrm{tr}\left(\frac{\partial \mathbf{v}}{\partial \mathbf{x}_t}(\mathbf{x}_t, t)\right) \tag{8}$$

For the proof of this theorem, please refer to (Chen et al. 2018). Using the autograd function provided by PyTorch, we can efficiently compute gradients of scalar values. However, in the instantaneous change of variable formula, computing the trace of the Jacobian matrix for the velocity function at different time points becomes computationally expensive, as it would require enumerating each component of the vector field $\mathbf{v}$ through autograd.

To address this, we employ unbiased Hutchinson's trace estimator (Hutchinson 1990) for efficient approximation. Specifically, for a random vector ϵ satisfying $\mathbb{E}[\epsilon] = \mathbf{0}$ and $\mathbb{E}[\epsilon\epsilon^\top] = \mathbf{I}$, the trace of a matrix $\mathbf{A}$ can be estimated through $\mathbb{E}[\epsilon^\top \mathbf{A}\epsilon]$. In our implementation, to estimate $\mathrm{tr}\left(\frac{\partial \mathbf{v}}{\partial \mathbf{x}_t}(\mathbf{x}_t, t)\right)$, we sample $\epsilon \sim \mathbf{N}(\mathbf{0}, \mathbf{I})$ and compute $\epsilon^\top \frac{\partial}{\partial \mathbf{x}_t}(\mathbf{v}(\mathbf{x}_t, t)^\top \epsilon)$. This approach leverages the efficiency of scalar backward propagation in autograd while avoiding explicit Jacobian computation.

To estimate the log-likelihood of a given data point at time $t = 1$, we use Euler method to numerically solve the flow-matching velocity ODE $\frac{d\mathbf{x}_t}{dt} = \mathbf{v}(\mathbf{x}_t, t)$ to obtain a discrete trajectory approximation. At each discrete time point along the trajectory, we estimate the log-likelihood change using Hutchinson's trace estimator, where multiple independent Gaussian noise vectors ϵ are sampled simultaneously to reduce the estimator's variance. For the initial log-likelihood at $t = 0$, the standard Gaussian prior enables direct analytical computation. Summing the initial log-likelihood with the accumulated estimated changes yields the final continuous log-likelihood estimate for the input data.

Now that we have shown how to estimate the continuous log-likelihood for standard flow matching models, since our secondary model is a conditioned flow matching model, we can simply pass the conditioning variables to the model and estimate the likelihood in the same manner. We solve the ODE with 40 discrete steps and use 20 independent standard Gaussian variables to estimate the trace term at each discrete time step in implementation.

It is worth noting that the discrete log-likelihood and continuous log-likelihood are not directly comparable, as the former is based on probability values while the latter relies on probability density functions. For both Topic-StoryCloze and StoryCloze benchmarks, we evaluate our model using three data types: speech-to-speech continuation (S), text-to-speech continuation (T→S), and speech-to-text continuation (S→T). When the output modality is text, we employ the discrete log-likelihood, whereas the continuous log-likelihood is used for speech outputs. Additionally, since our model employs an MLP classifier to predict the probability of outputting either a text token or grouped speech tokens from a LLM latent vector, we multiply these probability values by the discrete or continuous log-likelihood values to obtain the final log-likelihood values for text tokens or grouped speech tokens. Both metrics undergo standard normalization by the number of tokens during evaluation, following established practices in prior work (Nguyen et al. 2025; Hassid et al. 2023).