LLMVoX: Autoregressive Streaming Text-to-Speech Model for Any LLM

Sambal Shikhar[1] **Mohammed Irfan Kurpath**[1] **Sahal Shaji Mullappilly**[1] **Jean Lahoud**[1]
Fahad Khan[1,2] **Rao Muhammad Anwer**[1] **Salman Khan**[1] **Hisham Cholakkal**[1]

[1]Mohamed Bin Zayed University of AI (MBZUAI), UAE
[2]Linköping University, Sweden

{sambal.shikar, hisham.cholakkal}@mbzuai.ac.ae

Abstract

Recent advancements in speech-to-speech dialogue systems leverage LLMs for multimodal interactions, yet they remain hindered by fine-tuning requirements, high computational overhead, and text-speech misalignment. Existing *speech-enabled LLMs* often degrade conversational quality by modifying the LLM, thereby compromising its linguistic capabilities. In contrast, we propose **LLMVoX**, a ***lightweight 30M-parameter***, *LLM-agnostic, autoregressive streaming TTS* system that generates high-quality speech with low latency, while fully preserving the capabilities of the base LLM. Our approach achieves a significantly lower Word Error Rate compared to speech-enabled LLMs, while operating at comparable latency and UTMOS score. By decoupling speech synthesis from LLM processing via a multi-queue token streaming system, LLMVoX supports seamless, infinite-length dialogues. Its *plug-and-play* design also facilitates extension to various tasks with different backbones. Furthermore, LLMVoX generalizes to new languages with only dataset adaptation, attaining a low Character Error Rate on an Arabic speech task. Additionally, we have integrated LLMVoX with a Vision-Language Model to create an omni-model with speech, text, and vision capabilities, without requiring additional multimodal training. Our code base and project page is available at mbzuai-oryx.github.io/LLMVoX

1 Introduction

Large Language Models (LLMs) have excelled in the new era of conversational AI, transforming how machines understand, generate, and interact with humans. While most LLMs were initially designed for text-based interactions, there are some recent efforts toward more natural and intuitive *speech-to-speech* dialogue systems, allowing users to engage with AI models through spoken language.

Existing speech-enabled LLMs typically aims to *unify text and speech processing* within a single, fine-tuned LLM. Recent models such as Kyōtai Moshi (Défossez et al., 2024), Mini-Omni (Xie and Wu, 2024), LLaMA-Omni (Fang et al., 2024), and Freeze-Omni (Wang et al., 2024) extend or modify pretrained text-based LLMs, enabling them to directly handle spoken inputs and outputs. Although these end-to-end systems can offer faster and streamlined speech generation, they require *large-scale fine-tuning* of LLM on multimodal data. This fine-tuning with speech data often compromises the original reasoning and expressive capabilities of the base LLM (Chen et al., 2024b; Défossez et al., 2024; Kalajdzievski, 2024; Zhai et al., 2023), while also imposing substantial computational and data requirements for speech adaptation. Moreover, these architectures often condition speech adaptation on LLM hidden states, making them inherently *LLM-dependent*, thereby requiring re-adaptation for each base LLM.

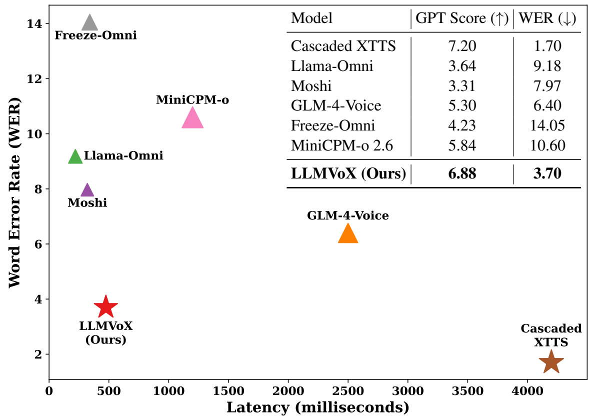

Figure 1: Speech quality (WER) vs latency (milliseconds) comparison of recent speech-enabled LLMs. Our LLMVoX is LLM-agnostic streaming TTS that generates high-quality speech (lower WER) comparable to XTTS (Casanova et al., 2024) while operating 10× faster. In the plot, △ represents LLM-dependent methods, and ★ denotes LLM-agnostic methods. The size of each symbol is proportional to the GPT score, indicating overall response quality. All methods are evaluated under similar settings and use similarly sized base LLMs.

Findings of the Association for Computational Linguistics: ACL 2025, pages 20481–20493
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

Alternatively, an *LLM-agnostic* approach is to leverage a ***cascaded pipeline***, where speech is converted to text via automatic speech recognition (ASR), processed by an LLM to generate a textual response, and finally passed through a text-to-speech (TTS) module for speech output. This cascaded approach offers several advantages, including the availability of diverse off-the-shelf ASR (Radford et al., 2023), LLM (Fang et al., 2024), and TTS (Casanova et al., 2024) models, the preservation of base LLM capabilities, improved speech quality, and an *LLM-agnostic* design that allows seamless adaptation to any base LLM in a plug-and-play manner, without the need for computationally expensive model retraining. However, such cascaded approaches often introduce high latency (see Cascaded-XTTS in Figure 1), making real-time interactions challenging. The primary reason for this high latency is the incompatibility between the autoregressive nature of LLM-based text generation and conventional TTS models, which typically process text inputs collectively, despite the text being available incrementally from LLM. This prevents speech generation from starting until the entire text response, or a large chunk of it, has been generated by the LLM. Furthermore, many existing TTS models rely on non-streaming speech decoders, leading to a larger delay between text and speech generation.

To address the aforementioned limitations of existing speech-enabled LLMs, we propose **LLMVoX**, an *autoregressive*, *LLM-agnostic* streaming framework. It aims to *preserve the underlying LLM's capabilities* by completely decoupling speech synthesis from the LLM, while enabling high-quality, low-latency speech generation (Figure 1) in an autoregressive setting, *running in parallel with the LLM's text generation*.

1.1 Contributions

Our LLMVoX leverages a lightweight transformer (Waswani et al., 2017) to generate *discretized speech tokens* in an autoregressive manner from streaming LLM text, making it straightforward to "plug" into any existing LLM pipeline without model retraining or fine-tuning.

LLMVoX adopts a multi-queue streaming approach to enable continuous and potentially *infinite-length* speech generation. By maintaining acoustic continuity and avoiding awkward pauses during extended dialogues, this design helps sustain a fluid user experience with minimal latency of 475 milliseconds for the entire cascaded pipeline including ASR (Radford et al., 2023), LLaMA-3.1-8B (Fang et al., 2024), and LLMVoX (Figure 1).

Furthermore, we demonstrate the generalization ability of the LLMVoX architecture to languages other than English by ***adapting it to Arabic*** for seamless plugging with Arabic LLM like Jais (Sengupta et al., 2023).This adaptation requires only a simple change in the LLMVoX training data to Arabic, without any architectural modifications, such as explicit Grapheme-to-Phoneme (G2P) conversion (Nguyen et al., 2023; Cherifi and Guerti, 2021; Jung et al., 2006), and can be similarly applied to any new language. Moreover, we integrated LLMVoX with a Vision Language Model (VLM) ***to obtain an omni-model*** with speech, text, and vision capabilities without explicit multimodal training.

The key contributions of our method are summarized below:

(i) We introduce LLMVoX, ***a lightweight 30M-parameter, LLM-agnostic, autoregressive streaming TTS*** framework that offers a plug-and-play solution for seamless integration with any off-the-shelf LLM or VLM—without fine-tuning or architectural modifications.

(ii) We use a ***multi-queue streaming mechanism*** that enables continuous, low-latency speech generation and *infinite-length speech*, effectively adapting to LLMs with different context lengths.

(iii) Our comprehensive experiments demonstrate that ***LLMVoX performs favorably compared to state-of-the-art speech-enabled LLMs*** in speech quality and latency while preserving the underlying LLM capabilities. Our cascaded system with LLMVoX achieves a WER of 3.70, maintains high speech quality with a UTMOS of 4.05, and delivers an end-to-end latency of 475ms (see Figure 1).

(iv) We demonstrate LLMVoX's ***ability to generalize to other languages, such as Arabic***, by simply modifying the training data-without any architectural changes. To this end, ***we generated 1,500 hours (450k pairs) of a synthetic, single-speaker Arabic text-speech dataset*** .

(v) Adapting LLMVoX to Arabic results in ***the first streaming, autoregressive Arabic speech generator that can be seamlessly integrated with any Arabic LLM***, such as Jais (Sengupta et al., 2023), to create Arabic speech-enabled LLMs. LLMVoX achieves a **CER** of $\sim 8\%$ comparable to even non-streaming Arabic TTS methods, while operating at lower latency—demonstrating the scalability and adaptability of our approach.

(vi) We further ***integrate LLMVoX with QWen 2.5-VL-7B VLM (Team, 2025) to obtain an omni-model with speech, text, and vision capabilities*** that do not require explicit multimodal training. This model ***performs favorably when compared to the state-of-the-art omni-model***, MiniCPM-o 2.6 (Yao et al., 2025), in visual *speech* question answering on LLaVA-Bench (in the wild) (Liu et al., 2024), while achieving 30% lower latency.

Here's a more concise version of the provided text on Streaming and Non-Streaming TTS:

2 Related Work

Streaming and Non-Streaming TTS: Offline non-autoregressive (NAR) TTS models such as FastSpeech (Ren et al., 2020), Tacotron (Wang et al., 2017), and F5-TTS (Chen et al., 2024c) cause significant conversational latency. Their non-incremental nature—requiring complete input text before synthesis and generating the entire speech output at once—leads to unnatural delays. These models also suffer from speech length constraints and poor adaptability, limiting their utility in dynamic conversational applications.

Autoregressive (AR) models like ParlerTTS (Lyth and King, 2024) and VALL-E (Wang et al., 2023) offer streaming audio output but induce latency by requiring full text input before synthesis. They also struggle with long speech generation due to fixed context lengths or memory issues. Furthermore, chunk-based streaming decoding with small windows can introduce noise.

TTS models like CosyVoice2 (Du et al., 2024), leveraging hybrid LMs and Flow Matching in an interleaved speech-text framework, support streaming text input and incremental speech generation. However, our English evaluations reveal significant quality trade-offs (pronunciation errors, omissions, noise), particularly with longer contexts. CosyVoice2 also demands substantial resources, evidenced by its 30,000-hour English training dataset.

Speech-enabled LLMs: Models such as Qwen-2 Audio (Chu et al., 2024), VITA (Fu et al., 2024), Ichigo (Dao et al., 2024), and Baichuan-Omni (Li et al., 2024) append speech adapters to LLMs for speech-to-text tasks, yet still rely on separate TTS modules, inheriting latency issues from cascaded pipelines. SpeechGPT (Zhang et al., 2023a), AudioPaLM (Rubenstein et al., 2023), EMOVA (Chen et al., 2024a), and AnyGPT (Zhan et al., 2024) integrate speech tokens directly into LLM vocabularies for end-to-end multimodal inference; however, as chain-of-modality methods, they incur latency by waiting for the complete text response before speech generation. Recent speech-enabled LLMs targeting low-latency interactions include Kyōtai Moshi (Défossez et al., 2024), which employs a dual-channel architecture with Mimi Neural Audio Codec for real-time dialogue; Mini-Omni (Xie and Wu, 2024), which combines text and speech modeling with batch-parallel inference to reduce delays; and LLaMA-Omni (Fang et al., 2024), which uses a CTC-based mechanism (latency ∼236ms). GLM-4-Voice (Zeng et al., 2024) trains on a trillion bilingual tokens with a low-bitrate (175bps) tokenizer for high-fidelity synthesis at higher compute cost; MiniCPM-o 2.6 (Yao et al., 2025, 2024) adopts an omni-modal LLM with a streaming speech decoder for real-time synthesis. Closer to our approach, Freeze-Omni (Wang et al., 2024) mitigates catastrophic forgetting by freezing the base LLM and integrating speech-specific modules. They employ a 3 stage training where LLM parameters are kept frozen throughout but in the final stage of training, Freeze-Omni conditions its speech decoder on LLM hidden states, necessitating retraining the speech components for any new base LLM, thereby limiting its plug-and-play capability.

Speech Tokenization: Mapping waveforms to discrete tokens compatible with transformers has advanced speech-to-speech modeling. Neural acoustic codecs (e.g., EnCodec (Défossez et al., 2022), LauraGPT (Du et al., 2023)) employ residual vector quantization (RVQ) for high-fidelity synthesis; hybrid approaches (e.g., SpeechTokenizer (Zhang et al., 2023b)) use hierarchical RVQ layers to enhance phonetic representation; and supervised tokenizers (e.g., CosyVoice (Du et al., 2024)) integrate vector quantization into ASR for improved text-speech alignment. Mimi (Défossez et al., 2024) employs split-RVQ for balanced phonetic discrimination and quality.

3 Methodology

Our proposed LLMVoX system in Figure 2 is a fully autoregressive Text-to-Speech (TTS) framework designed to convert text outputs from an upstream Large Language Model (LLM) into high-fidelity streaming speech. The central motivation behind our design is to decouple the speech synthesis component from the text-generation process so that the inherent reasoning and expressive ca-

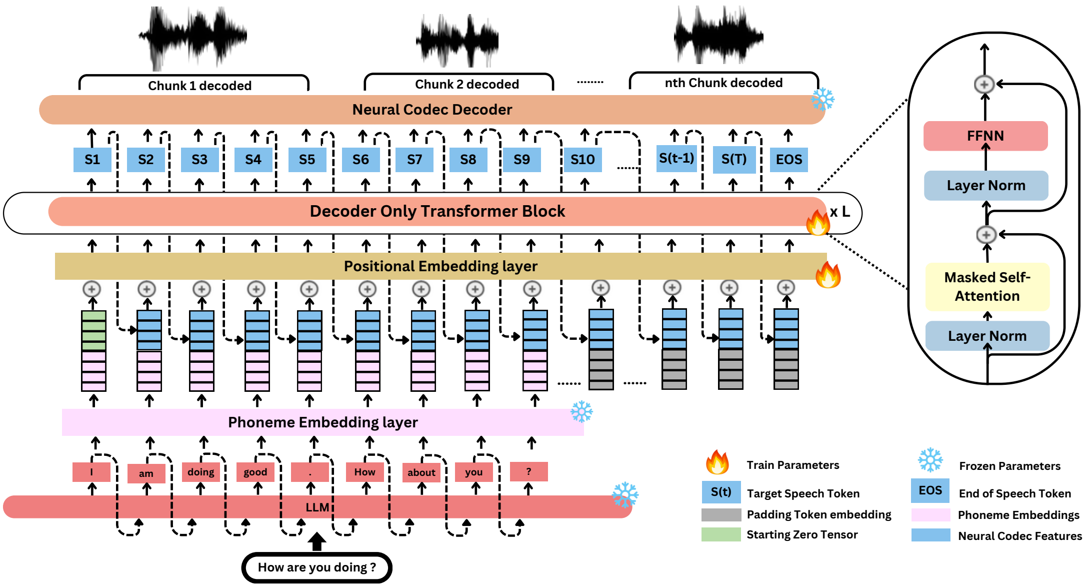

Figure 2: Overview of the proposed architecture. Text from the LLM is tokenized via a ByT5-based Grapheme-to-Phoneme(G2P) model, producing byte-level phoneme embeddings (teal). These are concatenated with the previous speech token's feature vector (blue), L2-normalized, and fed into a decoder-only Transformer to generate the next token. A neural codec (WavTokenizer) decoder (orange) reconstructs speech every n speech tokens predicted.

pabilities of the LLM remain unaltered while not compromising latency. By recasting TTS as a token prediction task over discrete acoustic units, we leverage Transformers architecture (Waswani et al., 2017) and neural audio representations to achieve natural, low-latency speech generation.

In our approach, the speech signal is represented as a sequence of discrete tokens drawn from a fixed vocabulary of 4096 entries. These tokens are generated by a neural audio codec, and the speech token is predicted token-by-token in an autoregressive manner. Figure 2 provides an overview of the overall architecture, where phoneme-aware embeddings derived from Grapheme-to-Phoneme (G2P) (Zhu et al., 2022) model are combined with previous acoustic context and processed by a decoder-only Transformer to predict the next speech token.

3.1 Neural Audio Tokenization

To model speech generation as an autoregressive task using Transformers (Wang et al., 2023), we use a neural audio codec that discretizes the continuous audio waveform using a single-layer residual vector quantization (RVQ) such as **WavTokenizer** (Ji et al., 2024). WavTokenizer yields a compact representation that supports high-quality speech reconstruction while keeping sequence lengths manageable. Given a 24 kHz waveform $\mathbf{x}$, the encoder $\mathrm{Enc}(\cdot)$ extracts latent feature vectors $\{\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_T\}$, where T is the number of tokens. Each feature $\mathbf{f}_t$ is quantized via $S_t = \mathrm{VQ}(\mathbf{f}_t)$ with $S_t \in \{1, \ldots, 4096\}$. Typically, 40–75 tokens represent one second of speech. The decoder $\mathrm{Dec}(\cdot)$ then reconstructs the audio waveform from these discrete token indices.

3.2 Byte-Level Grapheme-to-Phoneme Embedding

To infuse phonetic information into the synthesis process without incurring the overhead of explicit phoneme prediction, we employ the embedding layer of a ByT5-based Grapheme-to-Phoneme (G2P) model (Zhu et al., 2022). This decision is driven by two main considerations: (1) *Phonetic Richness*: This ByT5 based G2P model is fine-tuned on over 100 languages, so its embeddings capture subtle phonetic similarities and distinctions, ensuring accurate pronunciation, and (2) *Computational Efficiency:* By directly reusing the learned embeddings as a "table lookup", we avoid extra computation needed for explicit phoneme conversion, thus reducing latency.

Embedding Extraction and Padding Alignment. Let $\tilde{t}_1, \tilde{t}_2, \ldots, \tilde{t}_N$ denote the sequence of words produced by the LLM. Each word $\tilde{t}_i$ is decomposed into byte-level sub-tokens using the ByT5 tokenizer, i.e., $\tilde{t}_i \rightarrow [\beta_1^i, \beta_2^i, \ldots, \beta_{n_i}^i]$, where n_i is the number of sub-tokens for token $\tilde{t}_i$. Let M be the total number of sub-tokens from all text tokens. Each sub-token β_j^i is then mapped to an embedding vector as $\mathbf{b}_j^i = \text{Embed}_{\text{ByT5}}(\beta_j^i)$, where $\mathbf{b}_j^i \in \mathbb{R}^{256}$.

The ground-truth speech is tokenized into a sequence of T discrete speech tokens using WavTokenizer(Ji et al., 2024), where typically $T > M$. To align the length mismatch we pad the sub-token sequence to length T. Formally, the padded text embedding sequence $\{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_T\}$ is defined as:

$$\mathbf{b}_t = \begin{cases} \text{Embed}_{\text{ByT5}}(\beta_t), & \text{if } 1 \le t \le M, \\ \mathbf{b}_{\text{PAD}}, & \text{if } M < t \le T, \end{cases}$$

where β_t is the t-th sub-token and $\mathbf{b}_{\text{PAD}} \in \mathbb{R}^{256}$ is the embedding for the `<PAD>` token (obtained from the ByT5 embedding layer)(Xue et al., 2022). Although $\mathbf{b}_{\text{PAD}}$ does not encode phonetic information, the Transformer's self-attention mechanism will use context from the previous inputs to refine its representation.

3.3 Input Representation

At each time step t ($t = 1, \ldots, T$), the input vector is constructed by concatenating the phoneme embedding $\mathbf{b}_t \in \mathbb{R}^{256}$ with the latent acoustic feature vector $\mathbf{f}_{t-1} \in \mathbb{R}^{512}$ from the previous speech token S_{t-1}, forming $\mathbf{x}_t = [\mathbf{b}_t; \mathbf{f}_{t-1}] \in \mathbb{R}^{768}$. This vector is L2-normalized, and a learnable positional embedding $\mathbf{r}_t \in \mathbb{R}^{768}$ is added, yielding $\mathbf{z}_t = \mathbf{x}_t + \mathbf{r}_t$. The sequence $\{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_T\}$ is then fed into the decoder-only Transformer as shown in Figure 2.

3.4 Decoder-Only Transformer for Speech Token Generation

The core of our synthesis model is a lightweight decoder-only Transformer (4 layers) that autoregressively predicts the sequence of speech tokens $S_1, S_2, \ldots, S_T$. Our objective is to model the conditional probability $p\big(S_t \mid S_1, S_2, \ldots, S_{t-1}, \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_T\}, \theta\big)$ for each $t = 1, \ldots, T$, where θ denotes the transformer's. Moreover, At $t = 1$, no previous speech token is available. We thus initialize the acoustic context with a zero tensor ensuring that the model receives a consistent starting signal.

3.5 Training Objective and Procedure

Training LLMVoX involves minimizing the cross entropy loss over the ground-truth speech token sequence $\{S_1, \ldots, S_T\}$:

$$\mathcal{L} = -\sum_{t=1}^{T} \log p\big(S_t \mid S_{<t}, \mathbf{z}, \theta\big).$$

A causal mask is applied within the Transformer to enforce the autoregressive property.

Algorithm 1 Streaming Inference with Adaptive Chunk Size (Parallel Text Generation)

Require: Speech query $\mathbf{x}_{\text{user}}$
Ensure: Real-time speech $\hat{\mathbf{x}}$

$ASR\text{-}Text \leftarrow \text{ASR}(\mathbf{x}_{\text{user}})$
LLM-Text $\leftarrow \text{LLM}(ASR\text{-}Text)$ // Generate text tokens in parallel
Enqueue generated text tokens into FIFO queue $\mathcal{Q}_0$
Split $\mathcal{Q}_0$ into FIFO queues $\mathcal{Q}_1$ and $\mathcal{Q}_2$ (by sentence boundaries)
for all $i \in \{1, 2\}$ **in parallel do**
  $\{S_1, \ldots, S_M\} \leftarrow \text{LLMVoX}_i(\mathcal{Q}_i)$ // Generate speech tokens
  $chunk\_size \leftarrow n,\ startIdx \leftarrow 1$
  while $startIdx \le M$ **and** speech ongoing **do**
    $endIdx \leftarrow \min(startIdx + chunk\_size - 1, M)$
    Decode $\{S_{startIdx}, \ldots, S_{endIdx}\} \rightarrow \hat{\mathbf{x}}_i^{(m)}$; Enqueue into $\mathcal{P}_i$
    $startIdx \leftarrow endIdx + 1,\ chunk\_size \leftarrow 2 \cdot chunk\_size$
  end while
end for
Stream speech: Dequeue and stream chunks from $\mathcal{P}_1$ and $\mathcal{P}_2$ concurrently until complete.

4 Streaming Inference

We adopt a low-latency streaming inference pipeline (Figure 3 and Algorithm 1) for real-time speech dialogue system. Given the user's speech input $\mathbf{x}_{\text{user}}$, we first transcribe it using an ASR model (e.g., Whisper) to obtain $t_{\text{query}} = \text{ASR}(\mathbf{x}_{\text{user}})$. An LLM then generates a stream of words $\{\tilde{t}_1, \tilde{t}_2, \ldots, \tilde{t}_N\} = \text{LLM}(t_{\text{query}})$, which are alternately enqueued into two FIFO queues, $\mathcal{Q}_1$ and $\mathcal{Q}_2$, based on sentence boundaries. Two replica TTS modules, LLMVoX_1 and LLMVoX_2, concurrently dequeue words from $\mathcal{Q}_1$ and $\mathcal{Q}_2$ and predict speech tokens $\{S_1, S_2, \ldots, S_T\} = \text{LLMVoX}_i(\mathcal{Q}_i)$ for $i \in \{1, 2\}$. Every n speech token generated is then decoded into speech by WavTokenizer decoder and placed in producer queues $\mathcal{P}_1$ and $\mathcal{P}_2$ accordingly which is then streamed to the user immediately ensuring uninterrupted playback. The initial chunk size is n tokens, and after each segment is decoded, the chunk size doubles, leveraging the playback interval of previous speech to allow extra processing time as decoding larger chunks gives better speech output. This toggling mechanism seamlessly handles long or continuous text without requiring models with an extended or large **context window**.

5 Experimental Settings

Training Dataset: We use the *VoiceAssistant-400K* dataset from the Mini-Omni series (Xie and

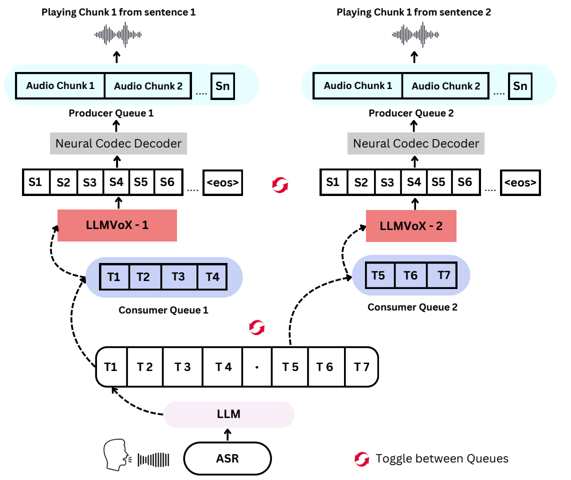

Figure 3: Overview of our streaming inference pipeline. Two replica TTS modules process text in parallel from two queues and place them into two producer queues.

Wu, 2024), which contains over 400K GPT-4o-generated question-answer pairs with corresponding synthesized speech, curated for speech assistant fine-tuning. Our training pipeline uses only the answer text and synthetic speech, resulting in approximately 2,200 hours of single-speaker English speech data. For Arabic, we collected 450K text entries of varying lengths from diverse Hugging Face corpora, cleaned the data, and generated corresponding speech using XTTS (Casanova et al., 2024) at a low-temperature setting, yielding about 1,500 hours of single-speaker Arabic speech data.

Training Configuration: Our streaming TTS model is a 4-layer, decoder-only Transformer ($n_{\text{embd}} = 768$, $n_{\text{head}} = 8$) trained with a micro-batch size of 4, `gradient_accumulation_steps` of 8, and a context block size of 8192 tokens. We use AdamW(Loshchilov et al., 2017) (`lr`=3×10^{-4}, `weight_decay`=0.1) with a 50K-step warmup, then decay the learning rate over 1M steps to 3×10^{-6}. Gradients are clipped at a norm of 1.0. The system runs on 4 A100 GPUs for around 3 days, using `bfloat16` precision. We use **flash-attention**(Dao et al., 2022) for efficient and fast training while also using **KV-Cache** while inferencing. Under these settings, we separately train English and Arabic models on 2,200 and 1,500 hours of single-speaker speech data, respectively.

6 Results and Evaluation

6.1 Evaluation Tasks

We evaluate LLMVoX on five key tasks: **General QA Capability** assesses the model's ability to generate coherent and informative responses to general queries, reflecting the preservation of the LLM's reasoning; **Knowledge Retention** measures the accuracy on fact-based questions to ensure robust information; **Speech Quality** examines the naturalness and clarity of the generated speech; **Speech-Text Alignment** verifies the consistency between the synthesized speech and corresponding text generated by the LLM. **Latency** is defined as the total elapsed time from when a query is submitted to when the model begins speaking.

6.2 Evaluation Datasets and Baselines

Datasets. We evaluate LLMVoX using diverse datasets spanning multiple dimensions. For **General QA**, we use questions from the AlpacaEval helpful-base and Vicuna subset (Li et al., 2023), excluding math-related queries. For **Knowledge QA**, 100 fact-based questions are sourced from Web Questions (Berant et al., 2013) and TriviaQA-verified (Joshi et al., 2017). To assess multilingual adaptability, we synthesize approximately 1,000 **Arabic** sentences from various domains. Additionally, for **Chunk Size Analysis**, we synthesize around 1,000 English sentences covering various topics, benchmarking the effects of chunk size on WER, CER, UTMOS, and latency. We also evaluate on Visual Speech Question Answering task (VSQA) on LLaVA-Bench (In-the-Wild) (Liu et al., 2024), which consists of 24 diverse images and 60 open-ended questions spanning various domains that suit conversational systems. We convert the text question to speech queries using XTTS (Casanova et al., 2024).

Comparison Models. LLMVoX is compared against recent speech-enabled LLMs which take in streaming input text and produce streaming audio output: **SpeechGPT** (Zhang et al., 2023a) (7B, expanded vocabulary), **Mini-Omni** (Xie and Wu, 2024) (0.5B, trained on VoiceAssistant-400K), **Llama-Omni** (Fang et al., 2024) (LLaMA-3.1-8B with CTC speech head), **Moshi** (Défossez et al., 2024) (7B Helium model, dual-channel processing), **GLM-4-Voice** (Zeng et al., 2024) (9B bilingual model with ultra-low bitrate tokenizer), and **Freeze-Omni** (Wang et al., 2024) (7B model with frozen LLM core) and **MiniCPM-o 2.6** (Yao et al., 2025). We also benchmark TTS models like **CosyVoice2** (Du et al., 2024) which takes in streaming input text and can produce streaming speech output. Autoregressive streaming speech output models like **ParlerTTs-mini-V1** (Wang et al.,

Model	Base LLM	Streaming	GPT-4o Score (↑)			UTMOS (↑)	WER (↓)	Latency (↓)
			General QA	*Knowledge*	*Avg.*	(1-5)	(%)	(ms)
Whisper+LLM+XTTS	LLaMA 3.1 8B	NS	6.70	7.70	7.20	4.23	1.70	4200
Whisper+LLM+F5-TTS	LLaMA 3.1 8B	NS	6.53	7.60	7.06	4.19	2.90	3200
Whisper+LLM+ParlerTTS-mini-V1	LLaMA 3.1 8B	SO	2.40	3.10	2.75	3.36	31.20	1950
SpeechGPT	LLaMA 2 13B	SO	1.40	2.20	1.80	3.86	66.57	4000
Mini-Omni	Qwen2 0.5B	FS	2.70	2.40	2.55	3.24	26.12	350
Llama-Omni	LLaMA 3.1 8B	FS	3.44	3.84	3.64	3.32	9.18	220
Moshi	Helium 7B	FS	2.71	3.91	3.31	3.92	7.97	320
GLM-4-Voice	GLM-4 9B	FS	5.24	5.67	5.30	3.97	6.40	2500
Freeze-Omni	Qwen2 7B	FS	3.48	4.98	4.23	**4.38**	14.05	340
MiniCPM-o 2.6	Qwen2.5 7B	FS	5.46	6.21	5.84	3.87	10.60	1200
Whisper+LLM+CosyVoice2	LLaMA 3.1 8B	FS	4.70	5.80	5.25	4.19	17.20	2100
Whisper+LLM+LLMVoX (Ours)	LLaMA 3.1 8B	FS	**6.14**	**7.62**	**6.88**	4.05	**3.70**	475

Table 1: Performance comparison of our framework, Whisper+LLM+LLMVoX, with other speech-enabled LLMs and cascaded systems. Our system utilizes **Whisper Small (224M)** for ASR, **LLaMA 3.1 8B** as the base Large Language Model, and **LLMVoX (30M)** for efficient speech synthesis. It achieves superior QA capabilities (General QA: 6.14, Knowledge QA: 7.62), while maintaining competitive speech quality (UTMOS 4.05), low latency (475ms), and excellent text-speech alignment (WER 3.70%). The 'Streaming' column indicates operational mode: NS (Non-Streaming, i.e., no streaming Text input/ Audio output), SO (Streaming Audio output only), and FS (Fully Streaming, i.e., streaming text input and streaming audio output). ↑ denotes higher values are preferable, while ↓ denotes lower values are preferable.

2023) which do no take in streaming text input were compared. We also benchmark a cascaded pipeline with non-streaming TTS like XTTS(Casanova et al., 2024) and F5-TTS (Chen et al., 2024c). All the models were evaluated on the basis of the best configuration given in the paper or the default configuration in the codebase. For Arabic TTS, no streaming comparison exists; hence we compare to non-streaming models - XTTS(Casanova et al., 2024), ArTST (Toyin et al., 2023), FastPitch (Łańcucki, 2021), Tacotron 2 (Elias et al., 2021) and Seamless (Barrault et al., 2023) in Table 3.

6.3 Evaluation Protocol

General QA and Knowledge Tasks: The questions are first converted into speech using XTTS with multiple speaker modes to introduce input variation. Model streaming speech responses are saved and transcribed using **Whisper-Large-v3** (Radford et al., 2023), and GPT-4o evaluates the quality and correctness of these transcriptions. For **General QA**, responses are scored from 1 to 10 based on coherence, informativeness, and fluency, following **MT-Bench protocols** (Zheng et al., 2023). For **Knowledge QA**, **GPT-4o** compares responses against ground-truth answers, with scores 0 for incorrect and 1 for correct response. The total accuracy score is then normalized from 1 to 10. Details of the evaluation prompts are given in Appendix 9.1.

Speech Quality: Naturalness is assessed using

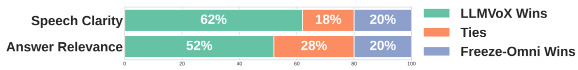

Figure 4: Human evaluation: Comparing with Freeze-Omni on Answer Relevance and Speech Quality.

UTMOS (Saeki et al., 2022), predicting MOS scores on a 1-5 scale.

Speech-Text Alignment: ASR Word Error Rate (WER) is calculated by comparing **Whisper-Large-v3** (Radford et al., 2023) transcriptions of the speech outputs with the LLM generated text averaged over General and Knowledge QA tasks.

Latency: Measured from the reception of speech input to the first speech output, capturing both processing and synthesis delays.

Human Evaluation: We compare our system with **Freeze-Omni**, one of the closely related approaches that freeze the base LLM. For setup details, see Appendix 9.2.

6.4 Experimental Results

Linguistic Capabilities: Our modular setup with Whisper for ASR, LLama 3.1 8B (Dubey et al., 2024) and LLMVoX achieves the highest GPT-4o score (see Table 1) among streaming models with 6.14 (General QA) and 7.62 (Knowledge QA) demonstrating its ability to preserve LLaMA 3.2 8B's language understanding capabilities. Although XTTS and F5-TTS slightly outperforms LLMVoX sharing the same base LLM due to lower WER, its high latency (4200ms vs 475ms) makes

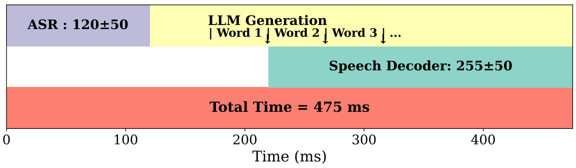

Figure 5: Breakdown of average end-to-end latency (in milliseconds) at a chunk size of 40 for a single query.

LLM	Params	Latency (s)
Qwen 2.5	0.5B	0.33
Lamma 3.2	3B	0.36
Lamma 3.1	8B	0.47
Phi 4	14B	0.95
Mixtral Small	24B	1.25
Qwen 2.5	32B	1.40
Lamma 3.3	70B	1.91

Table 2: End-to-end latency(ASR included) of LLMVoX with various LLMs at chunk size of 40.

Model	Streaming	WER (↓)	CER (↓)
XTTS	No	**0.062**	**0.017**
ArTST	No	0.264	0.125
FastPitch Arabic Finetuned	No	0.493	0.153
Tacotron 2 Arabic Finetuned	No	0.663	0.268
Tacotron 2 Arabic Finetuned	No	0.663	0.268
Seamless-M4t-Large	No	0.342	0.145
LLMVoX (Ours)	Yes	0.234	0.082

Table 3: Arabic TTS performance comparison. LLMVoX achieves competitive error rates in a streaming setup, operating at nearly 10x faster speed compared to state-of-the-art XTTS.

Model	WER	CER	GPT Score	Latency (s)
MiniCPM-o 2.6	0.053	0.036	6.32	1.45
LLMVoX (Ours)	0.042	0.022	6.41	1.05

Table 4: VSQA performance on LLaVA-Bench (In-the-Wild) with Qwen 2.5 VL 7B and VILA-1.5 8 B as the backbone.

it impractical for real-time use, highlighting the efficiency of LLMVoX. Notably, LLaMA-Omni, despite using the same LLaMA 3.1 8B base, underperforms in both QA tasks (3.44 vs. 6.14, 3.84 vs. 7.62), suggesting LLM degradation. Similarly, Freeze-Omni, despite freezing its LLM backbone, suffers from a high WER (14.05%), which lowers coherence and response quality. Also, based on human evaluation results in Figure 4, we observe that the response quality of our framework is much better than similar approach like Freeze-Omni that also its LLM parameters frozen. When compared to streaming input-output TTS models like CosyVoice2, LLMVoX performs much better in both General QA and Knowledge QA.

Speech Quality & Alignment: While Freeze-Omni yields a high UTMOS (Table 1), its WER is substantially high (14.05%), indicating a misalignment between the generated speech and text. In contrast, LLMVoX achieves the lowest WER at 3.70%, demonstrating superior text-to-speech consistency while maintaining a strong UTMOS score of 4.05. From the human evaluation results in Figure 4, our approach favours speech clarity compared to Freeze-Omni by a significant margin. Streaming input-ouput TTS models like Cosyvoice2 under the same setup as LLMVoX shows significantly high WER of 17.20.

Latency Analysis: One of the key challenges in real-time TTS is balancing low latency with high speech quality. LLMVoX successfully achieves this, delivering an end-to-end latency of 475ms, making it competitive with end-to-end streaming-capable models while significantly improving upon cascaded approaches like Whisper+LLM+XTTS (4200ms). While Llama-Omni achieves lower latency (220ms), its trade-off in WER (9.18%) and low UTMOS score of 3.32. In contrast, LLMVoX achieves a more optimal balance, reducing latency by nearly 86% compared to XTTS while maintaining superior WER. This is crucial for applications where both real-time response and textual accuracy are equally important, such as voice assistants. Figure 5 shows that LLMVoX starts generating speech tokens the moment LLM generates the first word, unlike other chain-of-modality models and cascaded pipelines, to achieve very low latency while operating in parallel to the LLM.

Observations on Chunk Size Impact: From Figure 6, we see that increasing the initial chunk size improves overall synthesis quality without significantly increasing latency. Key observations include: **UTMOS** improves from 3.75 to 4.41 as chunk size increases, suggesting speech reconstruction from larger chunk size results in smoother and more natural prosody. **WER** decreases from 0.041 to 0.036 confirming that larger chunks improve phonetic consistency. Latency remains under 1 second for chunk sizes as large as 160 ensuring real-time usability despite larger chunk sizes.

Latency Analysis with LLM Integration Table 2 shows that LLMVoX latency at a chunk size of 40 increases with LLM size. Smaller models like Qwen 2.5 (0.5B) and Lamma 3.2 (3B) achieve lower latencies (0.33–0.36s), while larger models such as Phi 4 (14B) and Lamma 3.3 (70B) exceed 1s. This indicates that while larger LLMs impose higher computational costs, architectural optimizations also impact latency.

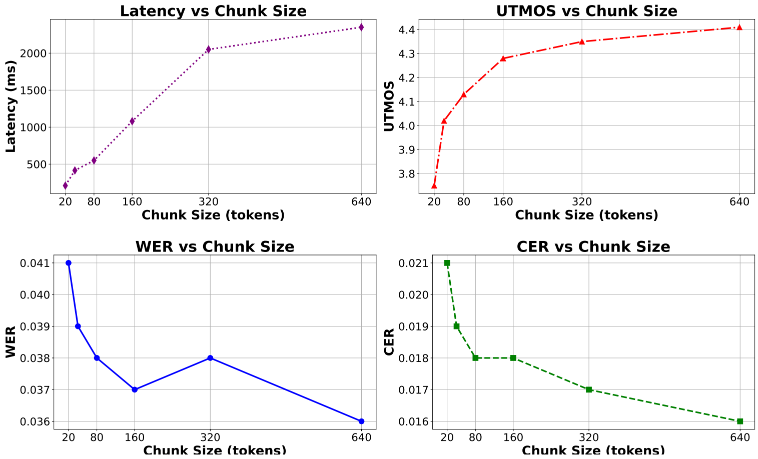

Figure 6: Effect of chunk size on WER, CER, UTMOS, and latency. Larger chunks enhance speech quality and reduce error rates.

6.5 Arabic Multilingual Performance:

On the curated Arabic eval set, LLMVoX achieves a CER of 8.2%, outperforming most non-streaming TTS methods except XTTS which was used to synthesize the Arabic Training data suggesting robust adaptability to new languages without explicit Grapheme-to-Phone(G2P) conversion or training.

6.6 Adaptability with Vision language Models

To demonstrate our method's versatility, we integrate LLMVoX into a multimodal pipeline for Visual Speech Question Answering (VSQA). Our setup combines **Whisper-Small** for ASR, **Qwen 2.5-VL-7B** (Team, 2025) for visual-language processing, and LLMVoX for speech synthesis. Table 4 compares our system with the omni-multimodal MiniCPM-o 2.6 model(Yao et al., 2025). We report word error rate (WER), character error rate (CER), and GPT-4o score. Our system achieves lower WER and a comparable GPT score, demonstrating that LLMVoX can be effectively plugged into state-of-the-art VLM pipelines for challenging speech VQA tasks.

6.7 Robustness to Varying Speech Lengths

We assessed LLMVoX's robustness to varying response lengths against CosyVoice2 0.5 Streaming by comparing their streaming Word Error Rate (WER %). Speech was generated from LLaMA 3.1 8B responses sourced from AlpacaEval (helpful-base and Vicuna subsets), for texts with a maximum of 1 to 5 sentences. Table 5 presents the results.

As shown in Table 5, CosyVoice2's quality significantly degrades with increasing response length; its WER rises from 12.2% to 17.1% with noticeable noise artifacts. In contrast, LLMVoX maintains consistent high quality across all tested lengths, with WER values stable between 3.72% and 4.1%. This robust performance is attributed to LLMVoX's compact 30M parameters, multi-queue token streaming, effective padding, and progressive decoding mechanism, which collectively enable effective handling of longer text sequences without quality degradation.

Max Sentences	CosyVoice2 0.5 Streaming (WER %)	LLMVoX (Ours) (WER %)
1	12.2	4.1
2	15.3	3.77
3	16.1	3.74
4	16.7	3.76
5	17.1	3.72

Table 5: Streaming WER (%) for CosyVoice2 0.5 Streaming vs. LLMVoX (Ours) as max response sentences increase. Lower WER is better.

Thus, LLMVoX demonstrates robust and reliable performance, essential for real-world applications where consistency across varying output lengths and low latency are critical.

7 Conclusion

We introduce LLMVoX, an LLM-agnostic autoregressive streaming TTS that decouples speech synthesis from text generation. Leveraging a lightweight Transformer and multi-queue streaming, LLMVoX delivers high-quality, continuous speech with minimal latency while preserving LLM reasoning. Experiments on English and Arabic tasks show that LLMVoX outperforms or matches other speech-enabled LLMs, offering a scalable solution for real-time multimodal AI.

8 Limitations

LLMVoX achieves low-latency streaming TTS without modifying the underlying LLM, but it has the following limitations. First, the system lacks voice cloning, which limits its ability to generate speaker-specific vocal characteristics—a key feature for personalized interactions. Second, while we use Whisper for ASR, it is not fully integrated into the streaming pipeline, leaving potential latency reductions unexplored. Future work will focus on incorporating voice cloning and extending the streaming architecture to the ASR input, further enhancing personalization and real-time performance.

Acknowledgments

We gratefully acknowledge support for this work from the Meta Llama Impact innovation Award, Meta Regional Research grant (Project OMER), Google research award, NVIDIA Academic grant and the MBZUAI-WIS research grant (P008).

References

Loïc Barrault, Yu-An Chung, Mariano Coria Meglioli, David Dale, Ning Dong, Mark Duppenthaler, Paul-Ambroise Duquenne, Brian Ellis, Hady Elsahar, Justin Haaheim, et al. 2023. Seamless: Multilingual expressive and streaming speech translation. *arXiv preprint arXiv:2312.05187*.

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.

Edresson Casanova, Kelly Davis, Eren Gölge, Görkem Göknar, Iulian Gulea, Logan Hart, Aya Aljafari, Joshua Meyer, Reuben Morais, Samuel Olayemi, et al. 2024. Xtts: a massively multilingual zero-shot text-to-speech model. *arXiv preprint arXiv:2406.04904*.

Kai Chen, Yunhao Gou, Runhui Huang, Zhili Liu, Daxin Tan, Jing Xu, Chunwei Wang, Yi Zhu, Yihan Zeng, Kuo Yang, et al. 2024a. Emova: Empowering language models to see, hear and speak with vivid emotions. *arXiv preprint arXiv:2409.18042*.

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

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

El-Hadi Cherifi and Mhania Guerti. 2021. Arabic grapheme-to-phoneme conversion based on joint multi-gram model. *International Journal of Speech Technology*, 24(1):173–182.

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*.

Alan Dao, Dinh Bach Vu, and Huy Hoang Ha. 2024. Ichigo: Mixed-modal early-fusion realtime voice assistant. *arXiv preprint arXiv:2410.15316*.

Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems*, 35:16344–16359.

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*.

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

Zhihao Du, Jiaming Wang, Qian Chen, Yunfei Chu, Zhifu Gao, Zerui Li, Kai Hu, Xiaohuan Zhou, Jin Xu, Ziyang Ma, et al. 2023. Lauragpt: Listen, attend, understand, and regenerate audio with gpt. *arXiv preprint arXiv:2310.04673*.

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

Isaac Elias, Heiga Zen, Jonathan Shen, Yu Zhang, Ye Jia, RJ Skerry-Ryan, and Yonghui Wu. 2021. Parallel tacotron 2: A non-autoregressive neural tts model with differentiable duration modeling. *arXiv preprint arXiv:2103.14574*.

Qingkai Fang, Shoutao Guo, Yan Zhou, Zhengrui Ma, Shaolei Zhang, and Yang Feng. 2024. Llama-omni: Seamless speech interaction with large language models. *arXiv preprint arXiv:2409.06666*.

Chaoyou Fu, Haojia Lin, Zuwei Long, Yunhang Shen, Meng Zhao, Yifan Zhang, Shaoqi Dong, Xiong Wang, Di Yin, Long Ma, et al. 2024. Vita: Towards open-source interactive omni multimodal llm. *arXiv preprint arXiv:2408.05211*.

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 S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*.

Youngim Jung, Aesun Yoon, and Hyuk-Chul Kwon. 2006. Grapheme-to-phoneme conversion of arabic numeral expressions for embedded tts systems. *IEEE transactions on audio, speech, and language processing*, 15(1):296–309.

Damjan Kalajdzievski. 2024. Scaling laws for forgetting when fine-tuning large language models. *arXiv preprint arXiv:2401.05605*.

Adrian Łańcucki. 2021. Fastpitch: Parallel text-to-speech with pitch prediction. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6588–6592. IEEE.

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.

Yadong Li, Haoze Sun, Mingan Lin, Tianpeng Li, Guosheng Dong, Tao Zhang, Bowen Ding, Wei Song, Zhenglin Cheng, Yuqi Huo, et al. 2024. Baichuan-omni technical report. *arXiv preprint arXiv:2410.08565*, 3(7).

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. 2024. Mmbench: Is your multi-modal model an all-around player? In *European conference on computer vision*, pages 216–233. Springer.

Ilya Loshchilov, Frank Hutter, et al. 2017. Fixing weight decay regularization in adam. *arXiv preprint arXiv:1711.05101*, 5.

Dan Lyth and Simon King. 2024. Natural language guidance of high-fidelity text-to-speech with synthetic annotations. *arXiv preprint arXiv:2402.01912*.

Linh The Nguyen, Thinh Pham, and Dat Quoc Nguyen. 2023. Xphonebert: A pre-trained multilingual model for phoneme representations for text-to-speech. *arXiv preprint arXiv:2305.19709*.

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.

Yi Ren, Chenxu Hu, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu. 2020. Fastspeech 2: Fast and high-quality end-to-end text to speech. *arXiv preprint arXiv:2006.04558*.

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

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

Neha Sengupta, Sunil Kumar Sahu, Bokang Jia, Satheesh Katipomu, Haonan Li, Fajri Koto, William Marshall, Gurpreet Gosal, Cynthia Liu, Zhiming Chen, et al. 2023. Jais and jais-chat: Arabic-centric foundation and instruction-tuned open generative large language models. *arXiv preprint arXiv:2308.16149*.

Qwen Team. 2025. Qwen2.5-vl.

Hawau Olamide Toyin, Amirbek Djanibekov, Ajinkya Kulkarni, and Hanan Aldarmaki. 2023. Artst: Arabic text and speech transformer. *arXiv preprint arXiv:2310.16621*.

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

Xiong Wang, Yangze Li, Chaoyou Fu, Lei Xie, Ke Li, Xing Sun, and Long Ma. 2024. Freeze-omni: A smart and low latency speech-to-speech dialogue model with frozen llm. *arXiv preprint arXiv:2411.00774*.

Yuxuan Wang, RJ Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, et al. 2017. Tacotron: Towards end-to-end speech synthesis. *arXiv preprint arXiv:1703.10135*.

A Waswani, N Shazeer, N Parmar, J Uszkoreit, L Jones, A Gomez, L Kaiser, and I Polosukhin. 2017. Attention is all you need. In *NIPS*.

Zhifei Xie and Changqiao Wu. 2024. Mini-omni: Language models can hear, talk while thinking in streaming. *arXiv preprint arXiv:2408.16725*.

Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. 2022. Byt5: Towards a token-free future with pre-trained byte-to-byte models. *Transactions of the Association for Computational Linguistics*, 10:291–306.

Yuan Yao, Tianyu Yu, Chongyi Wang, Junbo Cui, Bokai Xu, Hongji Zhu, Tianchi Cai, Fuwei Huang, Tianran Wang, Wenshuo Ma, Yixuan Zhou, Haoye Zhang, Zonghao Guo, Chi Chen, Haoyu Wang, Zhihui He, Haoyu Li, Hanyu Liu, Luoyuan Zhang, Ge Zhou, Siyuan Li, Zhi Zheng, Jie Zhou, Yuxuan Li, Kaihuo Zhang, Yudong Mei, Hanqing Zhao, Yueying Chen, Zhongwu Zhai, Hanbin Wang, Ganqu Cui, Ning Ding, Xu Han, Zhiyong Wu, Zhiyuan Liu, and Maosong Sun. 2025. Minicpm-o 2.6: A gpt-4o level mllm for vision, speech, and multimodal live streaming on your phone. https://github.com/OpenBMB/MiniCPM-o.

Yuan Yao, Tianyu Yu, Ao Zhang, Chongyi Wang, Junbo Cui, Hongji Zhu, Tianchi Cai, Haoyu Li, Weilin Zhao, Zhihui He, et al. 2024. Minicpm-v: A gpt-4v level mllm on your phone. *arXiv preprint arXiv:2408.01800*.

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*.

Yuexiang Zhai, Shengbang Tong, Xiao Li, Mu Cai, Qing Qu, Yong Jae Lee, and Yi Ma. 2023. Investigating the catastrophic forgetting in multimodal large language models. *arXiv preprint arXiv:2309.10313*.

Jun Zhan, Junqi Dai, Jiasheng Ye, Yunhua Zhou, Dong Zhang, Zhigeng Liu, Xin Zhang, Ruibin Yuan, Ge Zhang, Linyang Li, et al. 2024. Anygpt: Unified multimodal llm with discrete sequence modeling. *arXiv preprint arXiv:2402.12226*.

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

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

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36:46595–46623.

Jian Zhu, Cong Zhang, and David Jurgens. 2022. Byt5 model for massively multilingual grapheme-to-phoneme conversion. In *Interspeech*.

9 Appendix

9.1 Prompt for Evaluating Spoken Chatbots

This section describes the two primary GPT-4o prompts we use for evaluating spoken chatbot responses. Each prompt targets a different aspect of performance: (1) the overall quality of an answer (General QA) and (2) the correctness of the answer compared to reference responses (Knowledge).

9.1.1 General QA

[Instruction]
Please act as an impartial judge and evaluate the quality of the response provided by an AI assistant to the user question displayed below. Your evaluation should consider factors such as the helpfulness, relevance, accuracy, depth, creativity, and level of detail of the response. Begin your evaluation by providing a short explanation. Be as objective as possible. After providing your explanation, you must rate the response on a scale of 1 to 10 by strictly following this format: "Rating: [[5]]".
[Question]
{User's question goes here}
[The Start of Assistant's Answer]
{Assistant's response begins here}
[The End of Assistant's Answer]

9.1.2 Knowledge

[Instruction]
You will be given a question, the reference answers to that question, and an answer to be judged. Your task is to judge whether the answer to be judged is correct, given the question and reference answers. An answer is considered correct if it expresses the same meaning as at least one of the reference answers.
You should respond in JSON format. First provide a concise one-sentence analysis in the field "analysis", then your final judgment in the field "judgment", which can be "correct" or "incorrect".
[Question]
{User's question}
[Reference Answer]
{targets}
[Answer To Be Judged]
{answer_to_be_judged}
Example Output (in JSON format):

```
{
  "judgment": "correct",
  "analysis": "A concise explanation of
              correctness or incorrectness."
}
```

These prompts enable both qualitative (General QA) and correctness-based (Knowledge) evaluations of AI-generated spoken responses, ensuring a comprehensive assessment of the system's performance.

9.2 Human Evaluation Setup and Conclusion

We conducted a human evaluation to compare the streaming speech outputs of our proposed system with those of **Freeze-Omni**. Specifically, we randomly selected 30 questions from various domains and generated responses using both systems. These responses were distributed in batches of five per user, with a total of 20 users participating in the evaluation. For our system, we use Whisper-Small for ASR, LLaMA 3.1 8B as the LLM, and LLMVoX for streaming TTS, while **Freeze-Omni** served as the baseline. The streaming speech responses were recorded and a custom user interface was developed to facilitate evaluation. Participants listened to each response and rated the best response based on two metrics:
(i)Answer Relevance: Evaluates how factual, useful, and relevant the answer is to the question.
(ii)Speech Quality: Assesses the flow, word clarity, and pronunciation of the generated speech.
These choices were then aggregated to compare the overall performance of the two systems. The aggregated results are illustrated in Figure 4 Our human evaluation results indicate that our proposed system outperforms Freeze-Omni on both key metrics. Based on responses to the 30 questions, LLMVoX integrated with Whisper-Small for ASR and LLaMA 3.1 8B as the LLM received higher user ratings for both answer relevance and speech quality. Specifically, our model achieved wins in 52% of cases for answer relevance and 62% for speech quality, compared to Freeze-Omni's 20% wins on each metric. These findings suggest that decoupling speech synthesis from text generation not only preserves the linguistic capabilities of the LLM but also produces more natural, clear, and engaging speech output, demonstrating the effectiveness of our approach for real-time dialogue applications.