

---

# TiCo: Time-Controllable Training for Spoken Dialogue Models

---

Kai-Wei Chang<sup>♣\*</sup> Wei-Chih Chen<sup>◇\*</sup> En-Pei Hu<sup>◇</sup> Hung-yi Lee<sup>◇</sup> James Glass<sup>♣</sup>

♣ MIT   ◇ NTU

kwchang@mit.edu

## Abstract

We propose **TiCo**, a simple post-training method for enabling spoken dialogue models (SDMs) to follow time-constrained instructions and generate responses with controllable duration. This capability is valuable for real-world spoken language systems such as voice assistants and interactive agents, where controlling response duration can improve interaction quality. However, despite their strong ability to generate natural spoken responses, existing models lack time awareness and struggle to follow duration-related instructions (e.g., “Please generate a response lasting about 15 seconds”). Through an empirical evaluation of both open-source and commercial SDMs, we show that they frequently fail to satisfy such time-control requirements. TiCo addresses this limitation by enabling models to estimate elapsed speaking time during generation through Spoken Time Markers (STM) (e.g., `<10.6 seconds>`). These markers help the model maintain awareness of time and adjust the remaining content to meet the target duration. TiCo is simple and efficient: it requires only a small amount of data and no additional question-answer pairs, relying instead on self-generation and reinforcement learning. Experimental results show that TiCo significantly improves adherence to duration constraints while preserving response quality.

## 1 Introduction

“*Time is money*,” as famously stated by Benjamin Franklin, highlights the fundamental value of time in human life. In human–computer interaction, time is a critical resource that directly impacts usability, deployment cost, and safety-critical decision making. This is especially true for *Spoken Dialogue Models* (SDMs) [1–3], which are gaining increasing attention in real-world applications, such as personal assistants, wearable devices, and healthcare systems [4, 5]. In these scenarios, a response must not only be accurate and natural, but often also strictly bounded in duration. For example, a voice assistant may be required to provide a traffic update while driving; a wearable device may require concise spoken feedback due to battery or bandwidth constraints. Similarly, in medical or emergency scenarios, a voice assistant may need to deliver brief yet informative instructions under strict time pressure. In all of these cases, the ability to control response duration is a key requirement for practical deployment. Despite its importance, time controllability remains largely underexplored in SDMs.

In the domain of text Large Language Models (LLMs), prior studies have shown that models often struggle to follow explicit length-constraint instructions [6]. Moreover, LLM outputs often exhibit verbosity or length bias, a phenomenon associated with preference-based evaluation and alignment [7, 8]. This tendency weakens instruction-following capability and negatively affects user experience. While benchmarks, prompting and training strategies have begun to address length

---

\*Equal contribution

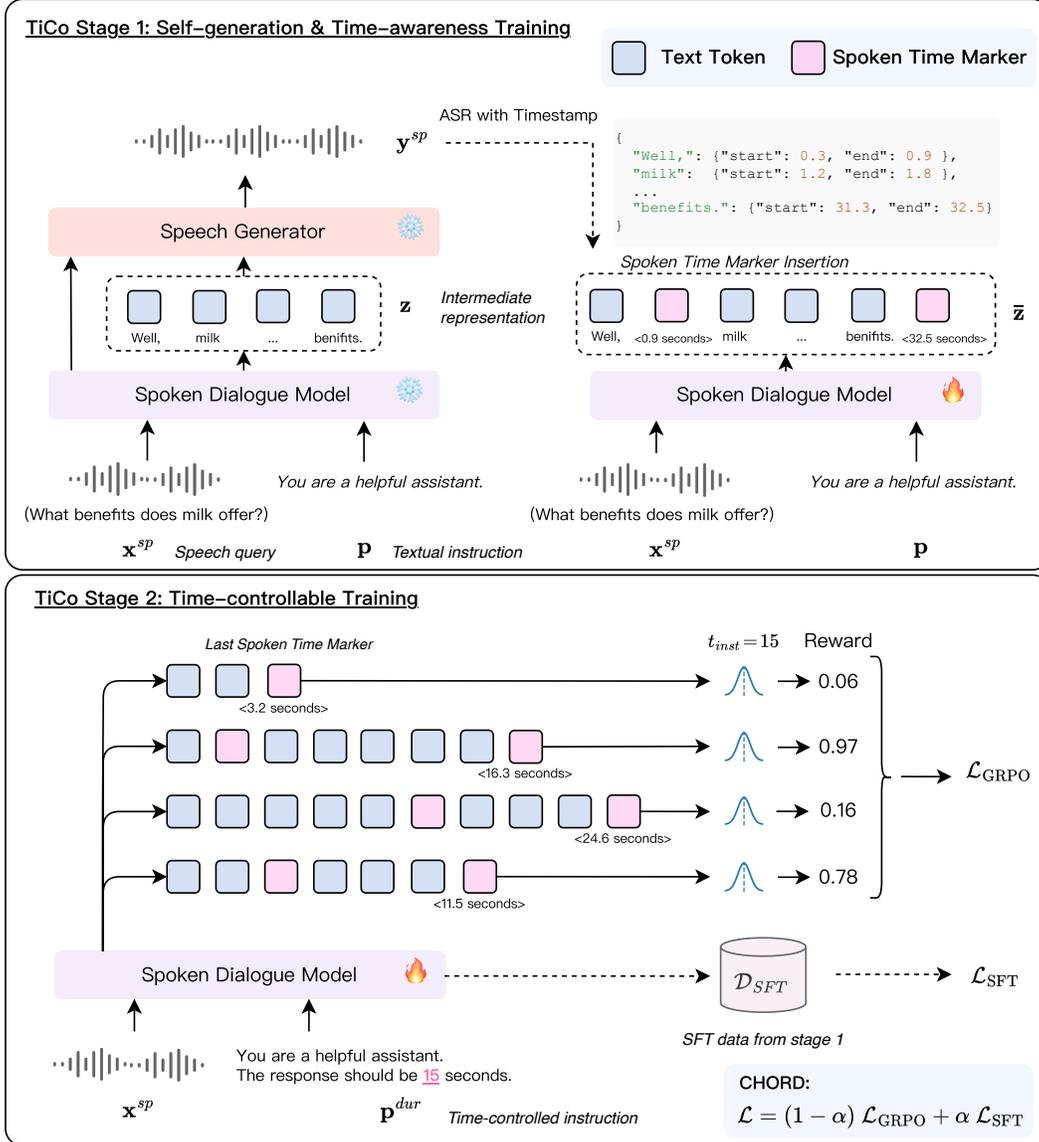


Figure 1: Overview of **TiCo**, a two-stage framework for time-controllable speech generation. **Stage 1 (top)**: The model leverages self-generation to produce responses annotated with **Spoken Time Markers (STMs)**, which serve as supervision for learning *time awareness*, i.e., associating intermediate generation states with temporal progress and estimating elapsed speaking time. **Stage 2 (bottom)**: The model is further optimized via RLVR, where rewards are derived from STMs, enabling the model to regulate response duration in real time.

controllability in text LLMs [6, 9–11], this research direction remains active and continues to attract attention due to its substantial practical importance.

However, controlling response duration in SDMs is considerably more challenging than controlling output length in text. In speech generation, word count is only a proxy for actual duration. A single word may contain different numbers of syllables, and speech duration is known to vary with phonetic composition, linguistic context, and prosodic structure [12]. Moreover, speaking rate may vary across speakers, speaking styles, and communicative conditions, which depends on both speaker and listener [13]. As a result, simply constraining the number of generated words does not guarantee accurate control over the final speech duration. This mismatch makes duration control a unique and more demanding problem in spoken dialogue systems.

Given the limited study of time controllability in SDMs, we first introduce **TiCo-Bench**, a benchmark designed to evaluate the time-controllable instruction-following capability of SDMs. Our evaluation reveals that existing SDMs struggle to reliably satisfy explicit time constraints.

To address this challenge, we propose **TiCo** (Figure 1), a simple yet effective *post-training* framework that enables SDMs to estimate and regulate generated speech duration in real time through **Spoken Time Markers**. The core idea of TiCo is to equip the model with an internal mechanism for *time awareness* during generation, allowing it to track temporal progress during generation and adjust its responses accordingly.

Specifically, TiCo consists of a two-stage training procedure. In the first stage, the model leverages *self-generation* to construct supervision data for learning duration awareness, enabling it to associate intermediate generation states with temporal progress and estimate the elapsed speaking time. In the second stage, *Reinforcement Learning with Verifiable Rewards (RLVR)* [14] is applied, where rewards are automatically verified based on the Spoken Time Markers, to further shape the response distribution and improve compliance with duration-related instructions. This stage encourages the model to better satisfy target time constraints while preserving the response quality, including helpfulness and coherence.

Our contributions are summarized as follows:

- We propose **TiCo**, a two-stage training framework that enables spoken dialogue models to generate **Spoken Time Markers (STMs)** during inference and perform real-time control over response duration.
- We introduce **TiCo-Bench**, the first benchmark designed to evaluate the time controllability of spoken dialogue models (SDMs), measuring whether they can follow explicit duration-related instructions.
- We conduct extensive experiments showing that TiCo significantly improves duration controllability while preserving response quality, and further demonstrate that the learned capability generalizes beyond the duration range seen during training.

## 2 Related Works

### 2.1 Spoken Dialogue Models

Spoken Dialogue Models (SDMs) [1, 2] aim to enable natural human-computer interaction by directly understanding and generating spoken conversations. Unlike traditional voice assistants that rely on cascaded ASR, text generation, and TTS modules, recent SDMs increasingly adopt end-to-end or tightly integrated modeling paradigms [15, 3].

However, compared to text-based LLMs operating in a semantically rich textual space, speech is considered to be significantly more challenging to process due to the high variability and complexity of acoustic signals<sup>2</sup>. As a result, many recent SDMs introduce *intermediate representations*, most commonly text, to support *semantic planning* during generating the speech response. This includes reasoning to improve response quality [21], tool calling [22] to leverage external modules, and more direct guidance over spoken content [23, 24]. Specifically, the SDM first takes the input query (in either text or speech form) to generate an intermediate representation, which is then consumed by a speech generator to produce the final output speech representation (e.g., phonetic tokens and acoustic tokens [1, 25, 26]), and subsequently synthesized into a waveform. We provide a survey of representative SDMs and their intermediate representations in the Appendix G.

Recently, several benchmarks have begun to evaluate SDMs beyond response quality, incorporating dimensions such as speaking style, time-awareness, and controllability. For instance, Full-Duplex-Bench [27, 28] evaluates full-duplex conversation by quantifying the ability of SDMs to engage in simultaneous speaking. ParaS2S [29] focuses on paralinguistic awareness, assessing how well SDMs capture aspects such as speaking styles and emotional expression. F-Actor [30] explores controllable conversational behavior in SDMs, showing that SDMs can be guided with instructions over attributes such as speaker voice, topic, and interaction style. In addition, and most related to

---

<sup>2</sup>This challenge has been largely explored in prior work such as the “*Textless NLP*” paradigm [14, 16–20], where quantized speech representations are treated as “pseudo text” to improve training efficiency and efficacy.

TiCo, Game-Time [31] instead targets time-awareness, including adherence to temporal instructions, tempo control, and overlapping speech generation. Despite the emergence of such benchmarks on controllability and time-awareness, to the best of our knowledge, TiCo is the first method that explicitly enables *time-controllable* generation for SDMs through a simple post-training approach.

It is worth noting that TiCo differs fundamentally from *duration modeling* in TTS systems [32, 33]. While duration modeling in TTS primarily focuses on aligning text with synthesized speech, TiCo instead targets time-controllable spoken response generation. This setting requires spoken dialogue models (SDMs) to perform semantic planning and reasoning while dynamically adapting to time-related constraints during generation. Moreover, TiCo is orthogonal to prior work on *temporal understanding* [34], which aims to equip speech models with the ability to interpret temporal information in input audio (e.g., “What is the time interval of the query ‘a dog barking’ in the audio?”) [35]. In contrast, TiCo focuses on time awareness in the *generation process*, rather than temporal comprehension of the input.

## 2.2 Length-Control Large Language Models

Precise regulation over the generation length of LLMs is critical for adapting these systems to constrained downstream applications and fulfilling specific user requirements. To enforce exact length constraints without incurring the computational overhead of retraining, researchers have proposed various training-free and decoding-time interventions. These methods encompass advanced sampling algorithms [36], task-specific zero-shot prompting strategies [37], and decoding interventions that dynamically increase the weight of the End-of-Sequence (EOS) token [38].

For more fine-grained regulation, instruction tuning approaches explicitly integrate positional awareness to continuously monitor the remaining generation length. Recent methodologies achieve this by incorporating length-tracking signals directly into the generation process, such as modifying the input encodings to reflect the distance to the target length [39], interleaving latent tracking tokens alongside the output sequence [11], or injecting explicit positional markers to enforce strict length constraints [40]. These generalized alignment strategies enable models to strictly adhere to predefined length constraints and reliably execute complex text manipulation operations including precise copy-and-paste tasks.

Beyond explicit instruction tuning, a parallel line of research leverages reinforcement learning (RL) and preference optimization to align large language models with human expectations. However, standard alignment processes frequently introduce verbosity biases, where models tend to associate longer responses with higher quality. To address this fundamental issue, advanced RL-based approaches have been refined to decouple explicit length biases from actual response evaluation [41, 42]. Expanding beyond the mere mitigation of superficial verbosity, recent RL paradigms have shifted focus toward actively managing the internal computational reasoning duration of models. Depending on the objective, these frameworks either establish constrained environments to enforce concise reasoning steps [43], or deliberately prolong these internal cognitive trajectories to expand the problem-solving capabilities of the system prior to generating a final answer [44, 45, 10].

## 3 TiCo

A speech-to-speech Spoken Dialogue Model (SDM) can be viewed as a conditional generative model that produces a spoken response  $\mathbf{y}^{\text{SP}}$  given the user’s input speech query  $\mathbf{x}^{\text{SP}}$  and a textual instruction  $\mathbf{p}$  (e.g., a system prompt).

Modern SDMs often introduce *intermediate representations*  $\mathbf{z}$  to bridge high-level semantic reasoning and low-level speech synthesis. Concretely, an **intermediate sequence generator**  $p_\theta$  first generates an intermediate representation conditioned on the user input:

$$\mathbf{z} \sim p_\theta(\mathbf{z} \mid \mathbf{x}^{\text{SP}}, \mathbf{p}). \quad (1)$$

The final spoken response is then generated by a **speech generator**  $q_\phi$ :

$$\mathbf{y}^{\text{SP}} \sim q_\phi(\mathbf{y}^{\text{SP}} \mid \mathbf{z}, \mathbf{x}^{\text{SP}}, \mathbf{p}). \quad (2)$$

Different architectures impose different conditional independence assumptions on Eq. (3). In cascaded systems, the speech synthesis module has no access to the original user speech or instruction, reducing

the generation to  $q_\phi(\mathbf{y}^{\text{SP}} | \mathbf{z})$ . In end-to-end models, the generation of  $\mathbf{y}^{\text{SP}}$  may additionally depend on  $\mathbf{x}^{\text{SP}}$  and  $\mathbf{p}$ <sup>3</sup>.

### 3.1 TiCo Stage1: Time-Awareness Training

This stage (Figure 1 (top)) trains the model to generate *Spoken Time Markers* as part of the intermediate representation  $\mathbf{z}$ , so that  $\mathbf{z}$  encodes not only semantic content but also its expected temporal alignment with the final spoken response  $\mathbf{y}^{\text{SP}}$  under the conditioning context  $(\mathbf{x}^{\text{SP}}, \mathbf{p})$ . These markers are inserted into  $\mathbf{z}$  through a self-generation process and used as prediction targets during training.

**Spoken Time Marker.** A Spoken Time Marker is a special token indicating the estimated cumulative speaking duration up to a given position in the intermediate representation. Conceptually, these markers serve as a discretized alignment signal between the intermediate semantic plan  $\mathbf{z}$  and the realized spoken response  $\mathbf{y}^{\text{SP}}$  under the same conditioning context  $(\mathbf{x}^{\text{SP}}, \mathbf{p})$ . Inspired by TimeMarker [46], we represent these markers in textual form, e.g., `<6.8 seconds>`.

Estimating duration at the intermediate level is non-trivial. A single word may correspond to multiple syllables, and its acoustic duration may vary depending on context and speaking rate. Explicit duration estimation is therefore required to bridge the gap between the intermediate representation and the final speech realization.

**Training Data Construction.** Let  $\mathcal{D} = \{(\mathbf{x}^{\text{SP}}, \mathbf{p})\}$  denote a pool of input speech query–instruction pairs. In this stage, we construct time-aware training targets through *self-generation* followed by ASR-based alignment. Specifically, given each input  $(\mathbf{x}^{\text{SP}}, \mathbf{p}) \in \mathcal{D}$ , the model first freely generates an intermediate representation  $\mathbf{z}$  and its corresponding spoken response  $\mathbf{y}^{\text{SP}}$ .

We then apply ASR-based alignment to estimate the temporal correspondence between  $\mathbf{z}$  and  $\mathbf{y}^{\text{SP}}$ . Based on the aligned timestamps, we define a sequence of Spoken Time Markers  $\mathbf{t} = [t_1, \dots, t_M]$ , where each  $t_j$  denotes the estimated cumulative speaking duration at an aligned position in  $\mathbf{z}$ . We interleave these markers with the intermediate tokens to obtain an augmented sequence:

$$\tilde{\mathbf{z}} = [z_1, \dots, z_i, t_j, \dots, z_N, t_M]. \quad (3)$$

As a result, the augmented sequence  $\tilde{\mathbf{z}}$  encodes not only semantic content, but also alignment-induced timing information that links  $\mathbf{z}$  to the final spoken response under the same input condition  $(\mathbf{x}^{\text{SP}}, \mathbf{p})$ .

This process yields an aligned training set  $\mathcal{D}_{\text{SFT}} = \{(\mathbf{x}^{\text{SP}}, \mathbf{p}, \tilde{\mathbf{z}})\}$ . We model the augmented intermediate sequence autoregressively as

$$p_\theta(\tilde{\mathbf{z}} | \mathbf{x}^{\text{SP}}, \mathbf{p}) = \prod_{n=1}^{|\tilde{\mathbf{z}}|} p_\theta(\tilde{z}_n | \tilde{\mathbf{z}}_{<n}, \mathbf{x}^{\text{SP}}, \mathbf{p}). \quad (4)$$

We then optimize the standard supervised fine-tuning (SFT) objective:

$$\mathcal{L}_{\text{SFT}} = -\mathbb{E}_{(\mathbf{x}^{\text{SP}}, \mathbf{p}, \tilde{\mathbf{z}}) \sim \mathcal{D}_{\text{SFT}}} \left[ \sum_{n=1}^{|\tilde{\mathbf{z}}|} \log p_\theta(\tilde{z}_n | \tilde{\mathbf{z}}_{<n}, \mathbf{x}^{\text{SP}}, \mathbf{p}) \right]. \quad (5)$$

It’s worth noting that self-generation offers two advantages: (1) it removes the need for collecting paired question-answer supervision, and (2) the generated responses follow the model’s own output distribution, which improves training stability [47].

### 3.2 TiCo Stage 2: Time-Controllable Training

This stage (Figure 1 (bottom)) further trains the model to follow time-constrained instructions. We augment the textual instruction  $\mathbf{p}$  with a duration constraint and denote the resulting instruction by  $\mathbf{p}^{\text{dur}}$ , where the target duration is denoted by  $t_{\text{inst}}$ . Since Spoken Time Markers reside in the intermediate representation, we apply reinforcement learning to the intermediate-sequence generator  $p_\theta(\tilde{\mathbf{z}} | \mathbf{x}^{\text{SP}}, \mathbf{p}^{\text{dur}})$ .

<sup>3</sup>For example, in Qwen-Omni’s “Thinker-Talker” design [23, 24]

Specifically, we adopt GRPO [14] to optimize time controllability, and incorporate CHORD [48] as a dynamically weighted auxiliary objective that integrates off-policy expert trajectories into the on-policy RL process. This regularization steers the policy toward the expert trajectories in the Stage-1-constructed dataset  $\mathcal{D}_{\text{SFT}}$  while preserving on-policy exploration. In practice, we find this mechanism crucial for stabilizing training, as GRPO alone frequently leads to reward hacking.

Given an input  $(\mathbf{x}^{\text{sp}}, \mathbf{p}^{\text{dur}})$ , we sample a group of  $G$  candidate augmented intermediate sequences from the old policy:

$$\tilde{\mathbf{z}}^{(g)} \sim p_{\theta_{\text{old}}}(\cdot | \mathbf{x}^{\text{sp}}, \mathbf{p}^{\text{dur}}), \quad g = 1, \dots, G. \quad (6)$$

**Reward Design.** The main reward measures the accuracy of the predicted total duration:

$$\mathcal{R}_{\text{main}}^{(g)} = F\left(t_{\text{inst}} - t_{\text{last}}^{(g)}\right), \quad (7)$$

where  $t_{\text{inst}}$  is the target duration specified in the instruction and  $t_{\text{last}}^{(g)}$  is the duration indicated by the final generated time marker in  $\tilde{\mathbf{z}}^{(g)}$ . We instantiate  $F$  as a Gaussian function, i.e.,  $F(\Delta t) = \exp(-(\Delta t)^2/(2\sigma^2))$ , where  $\sigma$  controls the tolerance to duration errors.

We additionally introduce several auxiliary rewards to stabilize training and mitigate reward hacking, including a **“presence reward”** that encourages the model to generate at least one time marker, a **“monotonicity reward”** that encourages time markers to increase monotonically, a **“repetition penalty”** that discourages repeatedly generating identical time markers, and a **“copy penalty”** that discourages trivial copying of the instructed duration. The detailed definitions of these auxiliary rewards, as well as the corresponding ablation study, are provided in Appendix B. The overall reward for the  $g$ -th sample is

$$R^{(g)} = \mathcal{R}_{\text{main}}^{(g)} + \mathcal{R}_{\text{aux}}^{(g)}. \quad (8)$$

We then optimize the intermediate-sequence generator with GRPO:

$$\mathcal{L}_{\text{GRPO}} = -\mathbb{E} \left[ \frac{1}{G} \sum_{g=1}^G \frac{1}{|\tilde{\mathbf{z}}^{(g)}|} \sum_{n=1}^{|\tilde{\mathbf{z}}^{(g)}|} \ell_{g,n}(\theta) \right], \quad (9)$$

where

$$\ell_{g,n}(\theta) = \min\left(\rho_{g,n}(\theta)\hat{A}^{(g)}, \text{clip}(\rho_{g,n}(\theta), 1 - \varepsilon, 1 + \varepsilon)\hat{A}^{(g)}\right) - \beta \mathbb{D}_{\text{KL}}[p_{\theta} \| p_{\text{ref}}]_n, \quad (10)$$

and

$$\rho_{g,n}(\theta) = \frac{p_{\theta}\left(\tilde{z}_n^{(g)} | \tilde{\mathbf{z}}_{<n}^{(g)}, \mathbf{x}^{\text{sp}}, \mathbf{p}^{\text{dur}}\right)}{p_{\theta_{\text{old}}}\left(\tilde{z}_n^{(g)} | \tilde{\mathbf{z}}_{<n}^{(g)}, \mathbf{x}^{\text{sp}}, \mathbf{p}^{\text{dur}}\right)}. \quad (11)$$

Here,  $\hat{A}^{(g)}$  denotes the group-relative normalized advantage for sample  $g$ , computed from  $\{R^{(g)}\}_{g=1}^G$  following GRPO.  $p_{\text{ref}}$  denotes the reference policy (i.e., the Stage 1 checkpoint before RL training), and  $\beta$  is the KL penalty coefficient.

Following CHORD [48], we additionally regularize training with expert trajectories from the first stage. The final training loss at optimization step  $s$  is

$$\mathcal{L}^{(s)} = (1 - \alpha_s) \mathcal{L}_{\text{GRPO}} + \alpha_s \mathcal{L}_{\text{SFT}}, \quad (12)$$

where  $\alpha_s$  is a step-dependent coefficient as described in CHORD [48]. Specifically,  $\alpha_s$  gradually decays over the course of training, allowing the regularizing effect of the SFT loss to diminish as the model improves.

## 4 Experiments

### 4.1 TiCo-Bench

**Dataset Construction.** As illustrated in Figure 2, TiCo-Bench is constructed by deriving samples from existing spoken and textual question datasets to provide a rigorous evaluation across diverse

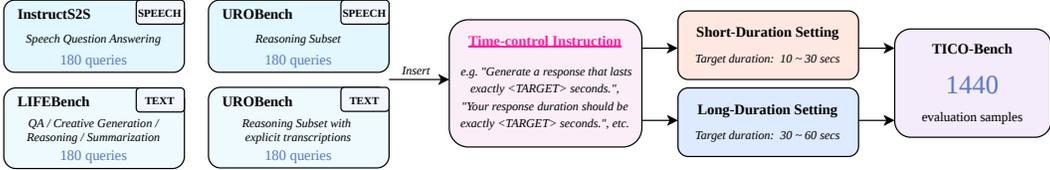


Figure 2: **Overview of TiCo-Bench construction.** Base queries are collected from four distinct text and speech datasets (totaling 720 queries). Explicit time-control instructions are then inserted into these queries. By applying both a short-duration setting (10–30 secs) and a long-duration setting (30–60 secs) to each query, the initial dataset is doubled, resulting in a final benchmark of 1440 evaluation samples.

tasks and modalities. Specifically, the benchmark is sourced from InstructS2S [49], UROBench [50], and LIFEbench [6]. InstructS2S is utilized to evaluate fundamental understanding capabilities. To assess performance in more complex scenarios, we incorporate UROBench, explicitly extracting its reasoning subset to evaluate cognitive processing. Furthermore, we leverage LIFEbench, an existing length-control benchmark in the text domain, whose task categories encompass question answering, creative generation, reasoning, and summarization.

Regarding input modalities, InstructS2S and UROBench are native speech datasets, whereas LIFEbench is a text-based dataset. We directly utilize the audio questions from InstructS2S and UROBench as speech queries to evaluate the spoken language models. Conversely, the text questions in LIFEbench serve as textual queries. To enable a more detailed ablation study regarding input modalities, we also utilize the explicit text transcriptions provided in UROBench, establishing a parallel set of text queries for comparative analysis.

To construct the final benchmark, we sample 180 questions from each dataset source. Because UROBench is repurposed to provide both speech and text queries, this creates four distinct evaluation subsets, totaling 720 unique base queries. We augment these queries with randomly assigned, time-controlled instructions. These instructions are formulated as textual prompts to explicitly guide the generative models toward the target response length (e.g., ‘Your response duration should be exactly 25 seconds.’). Every query is evaluated under two distinct temporal constraints: a short setting (10 to 30 seconds) and a long setting (30 seconds to 1 minute). This systematic augmentation yields a rigorous benchmark of 1,440 evaluation samples.

**Metrics.** We evaluate duration controllability using two metrics. Let  $d_i$  denote the actual duration of the generated speech for the  $i$ -th sample, and let  $t_{\text{inst},i}$  denote the target duration specified in the instruction.

The **Mean Absolute Error (MAE)** measures the average absolute deviation in seconds:

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^N |d_i - t_{\text{inst},i}|, \quad (13)$$

while the **Mean Absolute Percentage Error (MAPE)** normalizes the deviation by the target duration:

$$\text{MAPE} = \frac{1}{N} \sum_{i=1}^N \frac{|d_i - t_{\text{inst},i}|}{t_{\text{inst},i}} \times 100\%. \quad (14)$$

MAE captures the absolute magnitude of duration error, whereas MAPE provides a scale-invariant measure that enables fair comparison across different target durations.

We further evaluate response quality using a **GPT-score**. We first transcribe the generated speech using ASR, and then prompt an LLM (GPT-5-mini) to assess the response quality. The model assigns a score from 1 to 5 for each sample, and we report the average score.

**Baselines.** We compare against three categories of baselines in TiCo-Bench: (1) open-source SDMs, (2) commercial models, and (3) cascaded systems. For the cascaded strong baselines, we employ an LLM prompted to generate a response that satisfies the target duration constraint as closely as possible, and then use a text-to-speech system to synthesize the corresponding speech. Specifically, we utilize

GPT-5.2 [51] as a frontier commercial LLM and Qwen2.5-7B-Instruct [52] as a representative SoTA open-source language model. For the TTS component, IndexTTS-2 [53] is employed to generate high-quality speech from the LLM response. Detailed prompts used for the cascaded system can be found at the Appendix C.

To ensure that evaluation reflects generation quality rather than truncation artifacts, all SDMs are allocated a sufficiently large token budget to cover responses of up to 1 minute of speech.

## 4.2 Experimental Setup

We adopted MS-SWIFT (SWIFT) [54]<sup>4</sup> to train the model through out this paper. We adopt Qwen-2.5-Omni 7B [23] as the backbone model. Spoken Time Markers are inserted into the output of the “Thinker”. In both training stages of TiCo, only the “Thinker” is trained, while the “Talker” remains fixed. During inference, Spoken Time Markers are used only for intermediate planning and are removed via simple regex before feeding the cleaned sequence  $z$  into the “Talker” for speech generation.

Although our experiments are conducted on this architecture, TiCo is not restricted to a specific SDM design. The Spoken Time Marker mechanism can be applied to any spoken dialogue model that generates an intermediate representation prior to speech synthesis.

We sample 4,000 speech questions from InstructS2S [49] as in-domain training data (with 400 held out for evaluation). The training data do not overlap with the in-domain test set in TiCo-Bench. Word-level timestamps for constructing Spoken Time Markers are obtained using Whisper medium [55]. For simplicity, we insert a Spoken Time Marker after sentence-level punctuation marks (e.g., commas, periods, and exclamation marks). The distribution of Spoken Time Markers in the training data is shown in Figure 3.

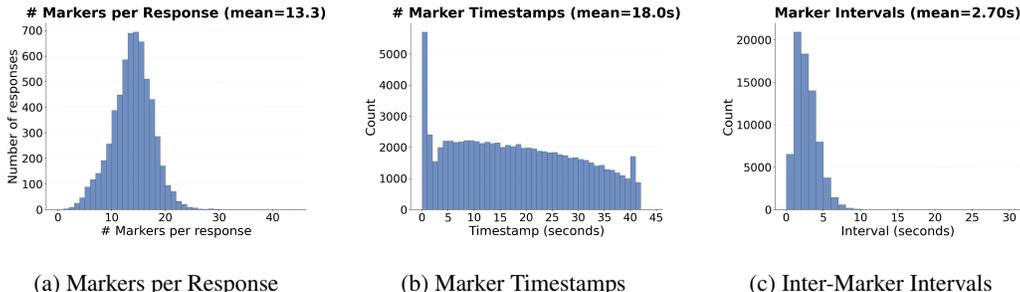


Figure 3: Distribution of Spoken Time Markers in the First stage training data.

During training, the maximum number of generated tokens for Qwen-2.5-Omni 7B is set to 2,048, corresponding to approximately 41 seconds of speech. At inference time, this limit is increased to 4,096 to support longer responses. This configuration is primarily adopted for efficiency and to evaluate the model’s ability to generalize to longer outputs, as TiCo-Bench extends up to one minute. In principle, the model can also be trained on longer-response data if desired. Additional training details are provided in the Appendix B.

## 5 Results

### 5.1 TiCo-Bench

As shown in Table 1, TiCo consistently outperforms all baselines across both datasets (in-domain InstructS2S and out-of-domain UROBench) and both duration settings. TiCo achieves the best overall performance with 4.54s MAE and 14.9% MAPE, substantially improving over its base model Qwen2.5-Omni-7B (13.01 s / 42.3%) and the strongest baseline, Cascade (GPT) (10.41 s / 29.0%). These results demonstrate that TiCo post-training significantly improves the time controllability of the backbone SDM.

<sup>4</sup><https://github.com/modelscope/ms-swift>

Table 1: **TiCo-Bench (Speech)** evaluation of time-controlled instruction-following performance on **speech-query** tasks under short (10s–30s) and long (30s–1min) settings. Model categories are indicated by color:   Cascaded,   Commercial,   Open-sourced, and   Proposed. Results are reported as MAE (seconds) / MAPE (%). Lower is better.

<i>TiCo-Bench (Speech)</i>	Short (10s–30s)		Long (30s–1min)		Overall	
	InstructS2S	UROBench	InstructS2S	UROBench	MAE / MAPE	GPT-score
Cascade (GPT)	4.09 / 19.6%	5.92 / 28.8%	12.17 / 25.7%	19.44 / 41.7%	10.41 / 29.0%	<b>4.15</b>
Cascade (Qwen)	8.12 / 37.7%	10.38 / 51.2%	21.77 / 46.8%	29.72 / 64.2%	17.50 / 50.0%	3.39
GPT-audio	7.79 / 36.3%	18.11 / 92.0%	12.42 / 27.5%	17.41 / 41.1%	13.93 / 49.2%	3.88
Kimi Audio	24.45 / 129.3%	34.46 / 196.7%	25.96 / 58.0%	32.28 / 74.1%	29.29 / 114.5%	1.67
LFM Audio	13.08 / 75.0%	12.52 / 62.6%	23.74 / 48.4%	35.06 / 75.6%	21.10 / 65.4%	2.63
Mimo-Audio	20.71 / 115.0%	12.45 / 74.7%	20.44 / 45.0%	18.37 / 39.9%	17.99 / 68.7%	3.32
Qwen2.5-Omni-7B	7.55 / 43.6%	8.62 / 50.6%	16.60 / 34.6%	19.27 / 40.4%	13.01 / 42.3%	3.57
<b>TiCo (Proposed)</b>	<b>3.16 / 15.6%</b>	<b>3.71 / 19.4%</b>	<b>5.16 / 11.3%</b>	<b>6.13 / 13.4%</b>	<b>4.54 / 14.9%</b>	3.56

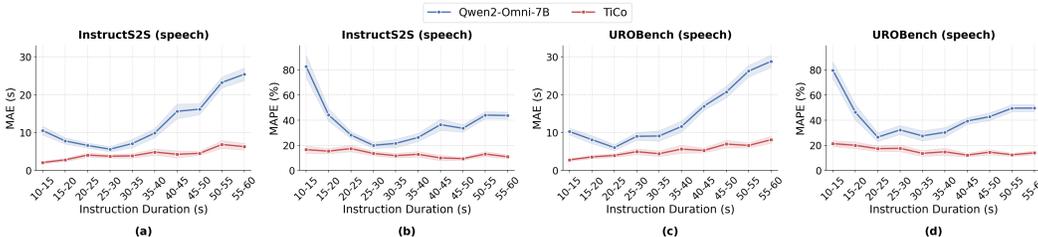


Figure 4: Duration MAE and MAPE of Qwen2-Omni-7B and TiCo across instructed-duration bins. TiCo maintains consistently lower error across all duration ranges.

It’s worth noting that the cascaded systems are more competitive than most SDM baselines, indicating that explicit duration planning with a strong LLM is helpful. Nevertheless, they remain clearly inferior to TiCo, suggesting that text-level duration estimation alone is insufficient for speech-level control, where the final duration also depends on speaking rate and speech realization.

Another notable trend is that most direct speech models perform much worse on the short subsets than on the long subsets in terms of relative error, indicating difficulty in satisfying tight time budgets. This observation is consistent with findings in text LLMs, where models often prefer longer responses. In contrast, TiCo maintains low error across both short and long settings and is the only model to achieve below 20% MAPE on all subsets.

## 5.2 Generalization to Longer Responses and Text Queries

We further examine whether TiCo generalizes beyond the conditions seen during post-training.

**Generalization to longer responses.** Although TiCo is post-trained with a maximum sequence length of 2048 audio tokens, corresponding to roughly 41 seconds of speech, as shown in Figure 3 (b), it generalizes well to substantially longer responses. As shown in Figure 4, TiCo maintains consistently low MAE and MAPE across instructed-duration bins on both the in-domain InstructS2S and out-of-domain UROBench benchmarks. In contrast, the error of the backbone Qwen2.5-Omni-7B increases noticeably as the target duration becomes longer, especially in the 45–60s bins. Notably, TiCo’s relative error on long-duration bins is comparable to, or even lower than, that on short-duration bins, indicating that TiCo can extrapolate its time-control capability to durations up to 1 minute.

**Generalization from speech to text queries.** TiCo is post-trained only with speech queries, yet it transfers well to text-query inputs. As shown in Table 2, TiCo achieves the best overall performance with 5.35 s MAE and 18.0% MAPE, substantially outperforming its backbone Qwen2.5-Omni-7B (13.63 s / 48.9%). Moreover, on LIFEbench, where cascaded systems are available, TiCo also surpasses the strongest cascaded baseline, Cascade (GPT), under both short and long settings. These

results suggest that the duration-control capability learned during post-training is not limited to the speech-query format seen during training.

Table 2: **TiCo-Bench (Text)** evaluation of time-controlled instruction-following performance on **text-query** tasks under short (10s–30s) and long (30s–1min) settings. Model categories are indicated by color: Cascaded, Commercial, Open-sourced, and Proposed. Results are reported as MAE (seconds) / MAPE (%). Lower is better.

TiCo-Bench (Text) Model	Short (10s–30s)		Long (30s–1min)		Overall	
	LIFEBench	UROBench-text	LIFEBench	UROBench-text	MAE / MAPE	GPT-score
Cascade (GPT)	4.83 / 23.5%	5.92 / 28.8%	8.35 / 18.4%	19.44 / 41.7%	9.64 / 28.1%	3.58
Cascade (Qwen)	7.98 / 40.4%	10.38 / 51.2%	19.16 / 42.9%	29.72 / 64.2%	16.81 / 49.7%	2.86
GPT-audio	14.33 / 73.6%	18.75 / 97.1%	11.07 / 26.3%	15.94 / 38.0%	15.02 / 58.8%	2.78
LFM Audio	18.84 / 101.1%	12.43 / 60.4%	35.74 / 79.1%	39.21 / 85.4%	26.56 / 81.5%	1.95
Mimo-Audio	12.66 / 62.0%	9.40 / 46.8%	31.75 / 69.7%	29.59 / 63.8%	20.85 / 60.6%	1.39
Qwen2.5-Omni-7B	13.74 / 79.1%	7.73 / 44.9%	13.86 / 30.8%	19.18 / 40.6%	13.63 / 48.9%	2.67
<b>TiCo (Proposed)</b>	<b>4.46 / 22.7%</b>	<b>4.08 / 21.0%</b>	<b>6.62 / 14.8%</b>	<b>6.25 / 13.5%</b>	<b>5.35 / 18.0%</b>	2.76

### 5.3 Spoken Time Token Prediction Analysis

TiCo relies on Spoken Time Token generation during semantic planning. Figure 5 compares the instructed duration with (i) the final response duration and (ii) the duration indicated by the last Spoken Time Token. The results show that these two errors are closely aligned, suggesting that the last Spoken Time Token provides a reliable approximation of the final response duration. However, we do observe a room for further improvement. Enhancing the training strategy for more accurate time prediction is left for future work.

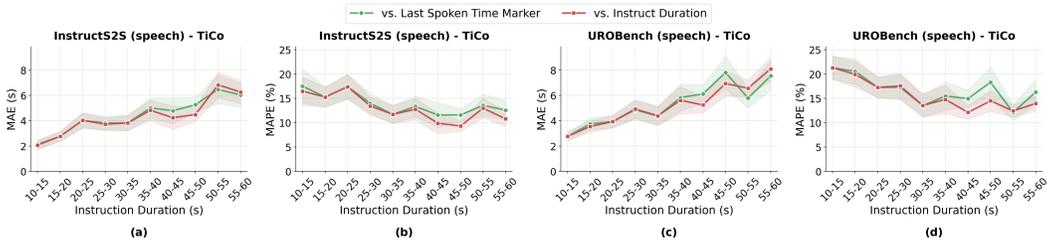


Figure 5: Duration error of TiCo across instructed-duration bins, comparing two reference signals: the instructed duration  $t_{inst}$  and the final Spoken Time Marker  $t_{last}$ . The close alignment indicates that the final time marker accurately estimates realized speech duration.

### 5.4 Qualitative Results

The qualitative results shown below demonstrate that the model can generate responses with varying lengths and adapt to different target durations while preserving the core semantic content. For a 15-second target, the response is concise and focused, whereas for a 40-second target, the model extends the utterance through additional elaboration and follow-up content while maintaining the same underlying answer. More examples are presented in Appendix F.

How deep is the ocean?, target duration = 15 s

**Prompt:** How deep is the ocean? Please respond with exactly 15 seconds.

**Speech response:**

Well, <0.9 seconds> you know, <1.6 seconds> the ocean is really deep in some places. <3.8 seconds> The Mariana Trench is the deepest part, <7.2 seconds> it's

about 36 kilometers deep. <9.4 seconds> But overall, <11.0 seconds> the average depth of the world’s oceans is around 3.7 kilometers. <15.0 seconds>

**Target duration:** 15.0 seconds

**Actual duration:** 15.2 seconds

How deep is the ocean?, target duration = 40 s

**Prompt:** How deep is the ocean? Please respond with exactly 40 seconds.

**Speech response:**

Well, <1.5 seconds> you know, <2.3 seconds> the ocean is really deep in some places. <4.6 seconds> The Mariana Trench is one of the deepest parts. <7.9 seconds> It’s about 36 kilometers deep. <10.8 seconds> But overall, <12.5 seconds> if we’re talking about an average depth for the whole ocean, <15.5 seconds> it’s around 3.7 kilometers. <18.0 seconds> However, <19.4 seconds> different areas can have very different depths. <22.3 seconds> Some shallow bays might be just a few meters deep while others in the middle of the ocean could be much deeper than that. <29.0 seconds> So, <30.0 seconds> it’s not really a fixed number like 3.7 kilometers everywhere. <33.0 seconds> If you want to know more about why the ocean has such varying depths or anything else related, <39.0 seconds> feel free to ask. <40.0 seconds>

**Target duration:** 40.0 seconds

**Actual duration:** 41.6 seconds

## 6 Conclusion

In this work, we introduced **TiCo**, a simple yet effective post-training framework that enables spoken dialogue models to follow explicit time-controlled instructions through **Spoken Time Markers** and a two-stage training process for time awareness and time-controllability. We also presented **TiCo-Bench**, the first benchmark designed to evaluate time-controllable instruction following in spoken dialogue models. Experimental results demonstrate that TiCo substantially improves duration controllability over strong open-source, commercial, and cascaded baselines, while maintaining response quality comparable to the backbone model. Moreover, TiCo generalizes beyond the duration range observed during training and transfers effectively from speech-query to text-query inputs, suggesting that temporal control can be learned as a robust intermediate planning capability. In future work, we aim to further improve the precision of time-marker prediction, extend training to longer and more diverse conversational scenarios, and explore how time control can be integrated with other controllable dialogue behaviors in spoken language systems.

## References

- [1] Siddhant Arora, Kai-Wei Chang, Chung-Ming Chien, Yifan Peng, Haibin Wu, Yossi Adi, Emmanuel Dupoux, Hung-yi Lee, Karen Livescu, and Shinji Watanabe. On the landscape of spoken language models: A comprehensive survey. *Transactions on Machine Learning Research*.
- [2] Wenqian Cui, Dianzhi Yu, Xiaoqi Jiao, Ziqiao Meng, Guangyan Zhang, Qichao Wang, Steven Y Guo, and Irwin King. Recent advances in speech language models: A survey. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13943–13970, 2025.
- [3] Shengpeng Ji, Yifu Chen, Minghui Fang, Jialong Zuo, Jingyu Lu, Hanting Wang, Ziyue Jiang, Long Zhou, Shujie Liu, Xize Cheng, et al. Wavchat: A survey of spoken dialogue models. *arXiv preprint arXiv:2411.13577*, 2024.
- [4] Allan de Barcelos Silva, Marcio Miguel Gomes, Cristiano André Da Costa, Rodrigo da Rosa Righi, Jorge Luis Victoria Barbosa, Gustavo Pessin, Geert De Doncker, and Gustavo Federizzi. Intelligent personal assistants: A systematic literature review. *Expert systems with applications*, 147:113193, 2020.

- [5] Scott J Adams, Julián N Acosta, and Pranav Rajpurkar. How generative ai voice agents will transform medicine. *npj Digital Medicine*, 8(1):353, 2025.
- [6] Wei Zhang, Zhenhong Zhou, Kun Wang, Junfeng Fang, Yuanhe Zhang, Rui Wang, Ge Zhang, Xavier Li, Li Sun, Lingjuan Lyu, et al. Lifebench: Evaluating length instruction following in large language models. *arXiv preprint arXiv:2505.16234*, 2025.
- [7] Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B Hashimoto. Length-controlled alpacaeval: A simple way to debias automatic evaluators. *arXiv preprint arXiv:2404.04475*, 2024.
- [8] Zhengyu Hu, Linxin Song, Jieyu Zhang, Zheyuan Xiao, Tianfu Wang, Zhengyu Chen, Nicholas Jing Yuan, Jianxun Lian, Kaize Ding, and Hui Xiong. Explaining length bias in LLM-based preference evaluations. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 6763–6794, November 2025.
- [9] Juncheng Xie and Hung-yi Lee. Prompt-based one-shot exact length-controlled generation with llms. *arXiv preprint arXiv:2508.13805*, 2025.
- [10] Renlong Jie, Xiaojun Meng, Lifeng Shang, Xin Jiang, and Qun Liu. Prompt-based length controlled generation with reinforcement learning. *arXiv preprint arXiv:2308.12030*, 2023.
- [11] Seoha Song, Junhyun Lee, and Hyeonmok Ko. Hansel: Output length controlling framework for large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 25146–25154, 2025.
- [12] Dennis H Klatt. Linguistic uses of segmental duration in english: Acoustic and perceptual evidence. *The journal of the acoustical society of America*, 59(5):1208–1221, 1976.
- [13] Björn Lindblom. Explaining phonetic variation: A sketch of the h&h theory. In *Speech production and speech modelling*, pages 403–439. Springer, 1990.
- [14] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.
- [15] James Glass. Challenges for spoken dialogue systems. In *Proceedings of the 1999 IEEE ASRU Workshop*, volume 696. MIT Laboratory for Computer Science Cambridge, 1999.
- [16] Adam Polyak, Yossi Adi, Jade Copet, Eugene Kharonov, Kushal Lakhota, Wei-Ning Hsu, Abdelrahman Mohamed, and Emmanuel Dupoux. Speech resynthesis from discrete disentangled self-supervised representations. In *Proc. Interspeech 2021*, pages 3615–3619, 2021.
- [17] Michael Hassid, Tal Remez, Tu Anh Nguyen, Itai Gat, Alexis Conneau, Felix Kreuk, Jade Copet, Alexandre Defossez, Gabriel Synnaeve, Emmanuel Dupoux, et al. Textually pretrained speech language models. *Advances in Neural Information Processing Systems*, 36:63483–63501, 2023.
- [18] Kai-Wei Chang, Wei-Cheng Tseng, Shang-Wen Li, and Hung-yi Lee. Speechprompt: An exploration of prompt tuning on generative spoken language model for speech processing tasks. *arXiv preprint arXiv:2203.16773*, 2022.
- [19] Kai-Wei Chang, Haibin Wu, Yu-Kai Wang, Yuan-Kuei Wu, Hua Shen, Wei-Cheng Tseng, Iu-thing Kang, Shang-Wen Li, and Hung-yi Lee. Speechprompt: Prompting speech language models for speech processing tasks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 32:3730–3744, 2024.
- [20] Sravya Popuri, Peng-Jen Chen, Changhan Wang, Juan Pino, Yossi Adi, Jiatao Gu, Wei-Ning Hsu, and Ann Lee. Enhanced direct speech-to-speech translation using self-supervised pre-training and data augmentation. 2022.
- [21] Cheng-Han Chiang, Xiaofei Wang, Linjie Li, Chung-Ching Lin, Kevin Lin, Shujie LIU, Zhendong Wang, Zhengyuan Yang, Hung yi Lee, and Lijuan Wang. STITCH: Simultaneous thinking and talking with chunked reasoning for spoken language models. In *The Fourteenth International Conference on Learning Representations*, 2026. URL <https://openreview.net/forum?id=5Z1eMhCeTb>.

- [22] Siddhant Arora, Haidar Khan, Kai Sun, Xin Luna Dong, Sajal Choudhary, Seungwhan Moon, Xinyuan Zhang, Adithya Sagar, Surya Teja Appini, Kaushik Patnaik, et al. Stream rag: Instant and accurate spoken dialogue systems with streaming tool usage. *arXiv preprint arXiv:2510.02044*, 2025.
- [23] Jin Xu, Zhifang Guo, Jinzheng He, Hangrui Hu, Ting He, Shuai Bai, Keqin Chen, Jialin Wang, Yang Fan, Kai Dang, Bin Zhang, Xiong Wang, Yunfei Chu, and Junyang Lin. Qwen2.5-omni technical report, 2025. URL <https://arxiv.org/abs/2503.20215>.
- [24] Jin Xu, Zhifang Guo, Hangrui Hu, Yunfei Chu, Xiong Wang, Jinzheng He, Yuxuan Wang, Xian Shi, Ting He, Xinfa Zhu, et al. Qwen3-omni technical report. *arXiv preprint arXiv:2509.17765*, 2025.
- [25] Yiwei Guo, Zhihan Li, Hankun Wang, Bohan Li, Chongtian Shao, Hanglei Zhang, Chenpeng Du, Xie Chen, Shujie Liu, and Kai Yu. Recent advances in discrete speech tokens: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.
- [26] Haibin Wu, Ho-Lam Chung, Yi-Cheng Lin, Yuan-Kuei Wu, Xuanjun Chen, Yu-Chi Pai, Hsiu-Hsuan Wang, Kai-Wei Chang, Alex Liu, and Hung-yi Lee. Codec-superb: An in-depth analysis of sound codec models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 10330–10348, 2024.
- [27] Guan-Ting Lin, Jiachen Lian, Tingle Li, Qirui Wang, Gopala Anumanchipalli, Alexander H Liu, and Hung-yi Lee. Full-duplex-bench: A benchmark to evaluate full-duplex spoken dialogue models on turn-taking capabilities. *arXiv preprint arXiv:2503.04721*, 2025.
- [28] Guan-Ting Lin, Shih-Yun Shan Kuan, Jiatong Shi, Kai-Wei Chang, Siddhant Arora, Shinji Watanabe, and Hung-yi Lee. Full-duplex-bench-v2: A multi-turn evaluation framework for duplex dialogue systems with an automated examiner. *arXiv preprint arXiv:2510.07838*, 2025.
- [29] Shu-wen Yang, Ming Tu, Andy T Liu, Xinghua Qu, Hung-yi Lee, Lu Lu, Yuxuan Wang, and Yonghui Wu. Paras2s: Benchmarking and aligning spoken language models for paralinguistic-aware speech-to-speech interaction. *arXiv preprint arXiv:2511.08723*, 2025.
- [30] Maike Züfle, Ondrej Klejch, Nicholas Sanders, Jan Niehues, Alexandra Birch, and Tsz Kin Lam. F-actor: Controllable conversational behaviour in full-duplex models. *arXiv preprint arXiv:2601.11329*, 2026.
- [31] Kai-Wei Chang, En-Pei Hu, Chun-Yi Kuan, Wenze Ren, Wei-Chih Chen, Guan-Ting Lin, Yu Tsao, Shao-Hua Sun, Hung-yi Lee, and James Glass. Game-time: Evaluating temporal dynamics in spoken language models. *arXiv preprint arXiv:2509.26388*, 2025.
- [32] Yi Ren, Chenxu Hu, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu. FastSpeech 2: Fast and high-quality end-to-end text to speech. In *International Conference on Learning Representations*, 2021.
- [33] Tianxin Xie, Yan Rong, Pengfei Zhang, Wenwu Wang, and Li Liu. Towards controllable speech synthesis in the era of large language models: A systematic survey. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 764–791, 2025.
- [34] Arvind Krishna Sridhar, Yinyi Guo, and Erik Visser. Enhancing temporal understanding in audio question answering for large audio language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 3: Industry Track)*, pages 1026–1035, 2025.
- [35] Hualei Wang, Yiming Li, Shuo Ma, Hong Liu, and Xiangdong Wang. Listening between the frames: Bridging temporal gaps in large audio-language models. *arXiv preprint arXiv:2511.11039*, 2025.
- [36] Yuxuan Gu, Wenjie Wang, Xiaocheng Feng, Weihong Zhong, Kun Zhu, Lei Huang, Ting Liu, Bing Qin, and Tat-Seng Chua. Length controlled generation for black-box llms. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16878–16895, 2025.

- [37] Fabian Retkowsky and Alex Waibel. Zero-shot strategies for length-controllable summarization. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 551–572, 2025.
- [38] Zeno Belligoli, Emmanouil Stergiadis, Eran Fainman, and Ilya Gusev. Controlling summarization length through eos token weighting. *arXiv preprint arXiv:2506.05017*, 2025.
- [39] Bradley Butcher, Michael O’Keefe, and James Titchener. Precise length control for large language models. *Natural Language Processing Journal*, 11:100143, 2025.
- [40] Noah Wang, Feiyu Duan, Yibo Zhang, Wangchunshu Zhou, Ke Xu, Wenhao Huang, and Jie Fu. Positionid: Llms can control lengths, copy and paste with explicit positional awareness. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 16877–16915, 2024.
- [41] Wei Liu, Yang Bai, Chengcheng Han, Rongxiang Weng, Jun Xu, Xuezhi Cao, Jingang Wang, and Xunliang Cai. Length desensitization in direct preference optimization. *arXiv preprint arXiv:2409.06411*, 2024.
- [42] Gengxu Li, Tingyu Xia, Yi Chang, and Yuan Wu. Length-controlled margin-based preference optimization without reference model. *arXiv preprint arXiv:2502.14643*, 2025.
- [43] Chang Liu, Yiran Zhao, Lawrence Liu, Yaoqi Ye, Csaba Szepesvári, and Lin F Yang. Laconic: Length-aware constrained reinforcement learning for llm. *arXiv preprint arXiv:2602.14468*, 2026.
- [44] Mingjie Liu, Shizhe Diao, Ximing Lu, Jian Hu, Xin Dong, Yejin Choi, Jan Kautz, and Yi Dong. Prorl: Prolonged reinforcement learning expands reasoning boundaries in large language models. *arXiv preprint arXiv:2505.24864*, 2025.
- [45] Pranjal Aggarwal and Sean Welleck. L1: Controlling how long a reasoning model thinks with reinforcement learning. *arXiv preprint arXiv:2503.04697*, 2025.
- [46] Shimin Chen, Xiaohan Lan, Yitian Yuan, Zequn Jie, and Lin Ma. Timemarker: A versatile video-llm for long and short video understanding with superior temporal localization ability. *arXiv preprint arXiv:2411.18211*, 2024.
- [47] Ke-Han Lu, Zhehuai Chen, Szu-Wei Fu, Chao-Han Huck Yang, Sung-Feng Huang, Chih-Kai Yang, Chee-En Yu, Chun-Wei Chen, Wei-Chih Chen, Chien-yu Huang, et al. Desta2. 5-audio: Toward general-purpose large audio language model with self-generated cross-modal alignment. *arXiv preprint arXiv:2507.02768*, 2025.
- [48] Wenhao Zhang, Yuexiang Xie, Yuchang Sun, Yanxi Chen, Guoyin Wang, Yaliang Li, Bolin Ding, and Jingren Zhou. On-policy rl meets off-policy experts: Harmonizing supervised fine-tuning and reinforcement learning via dynamic weighting, 2026. URL <https://arxiv.org/abs/2508.11408>.
- [49] Qingkai Fang, Shoutao Guo, Yan Zhou, Zhengrui Ma, Shaolei Zhang, and Yang Feng. Llama-omni: Seamless speech interaction with large language models. In *The Thirteenth International Conference on Learning Representations*, 2025.
- [50] Ruiqi Yan, Xiquan Li, Wenxi Chen, Zhikang Niu, Chen Yang, Ziyang Ma, Kai Yu, and Xie Chen. URO-bench: Towards comprehensive evaluation for end-to-end spoken dialogue models. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 17211–17242, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.933. URL <https://aclanthology.org/2025.findings-emnlp.933/>.
- [51] OpenAI. Update to gpt-5 system card: Gpt-5.2. Technical report, OpenAI, December 2025. URL [https://cdn.openai.com/pdf/3a4153c8-c748-4b71-8e31-aecbde944f8d/oai\\_5\\_2\\_system-card.pdf](https://cdn.openai.com/pdf/3a4153c8-c748-4b71-8e31-aecbde944f8d/oai_5_2_system-card.pdf).

- [52] Qwen Team. Qwen2.5: A party of foundation models, September 2024. URL <https://qwenlm.github.io/blog/qwen2.5/>.
- [53] Siyi Zhou, Yiquan Zhou, Yi He, Xun Zhou, Jinchao Wang, Wei Deng, and Jingchen Shu. IndexTTS2: A breakthrough in emotionally expressive and duration-controlled auto-regressive zero-shot text-to-speech. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 35139–35148, 2026.
- [54] Yuze Zhao, Jintao Huang, Jinghan Hu, Xingjun Wang, Yunlin Mao, Daoze Zhang, Zeyinzi Jiang, Zhikai Wu, Baole Ai, Ang Wang, Wenmeng Zhou, and Yingda Chen. Swift: a scalable lightweight infrastructure for fine-tuning, 2024. URL <https://arxiv.org/abs/2408.05517>.
- [55] Jérôme Louradour. whisper-timestamped. <https://github.com/linto-ai/whisper-timestamped>, 2023.
- [56] Tu Anh Nguyen, Eugene Kharitonov, Jade Copet, Yossi Adi, Wei-Ning Hsu, Ali Elkahky, Paden Tomasello, Robin Algayres, Benoit Sagot, Abdelrahman Mohamed, et al. Generative spoken dialogue language modeling. *Transactions of the Association for Computational Linguistics*, 11:250–266, 2023.
- [57] Dong Zhang, Shimin Li, Xin Zhang, Jun Zhan, Pengyu Wang, Yaqian Zhou, and Xipeng Qiu. Speechgpt: Empowering large language models with intrinsic cross-modal conversational abilities. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 15757–15773, 2023.
- [58] Tu Anh Nguyen, Benjamin Muller, Bokai Yu, Marta R Costa-Jussa, Maha Elbayad, Sravya Popuri, Christophe Ropers, Paul-Ambroise Duquenne, Robin Algayres, Ruslan Mavlyutov, et al. Spirit-lm: Interleaved spoken and written language model. *Transactions of the Association for Computational Linguistics*, 13:30–52, 2025.
- [59] Alexandre Défossez, Laurent Mazaré, Manu Orsini, Amélie Royer, Patrick Pérez, Hervé Jégou, Edouard Grave, and Neil Zeghidour. Moshi: a speech-text foundation model for real-time dialogue. *arXiv preprint arXiv:2410.00037*, 2024.
- [60] Bandhav Veluri, Benjamin N Peloquin, Bokai Yu, Hongyu Gong, and Shyamnath Gollakota. Beyond turn-based interfaces: Synchronous llms as full-duplex dialogue agents. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 21390–21402, 2024.
- [61] Zhifei Xie and Changqiao Wu. Mini-omni2: Towards open-source gpt-4o with vision, speech and duplex capabilities. *arXiv preprint arXiv:2410.11190*, 2024.
- [62] Xiong Wang, Yangze Li, Chaoyou Fu, Yike Zhang, Yunhang Shen, Lei Xie, Ke Li, Xing Sun, and Long Ma. Freeze-omni: A smart and low latency speech-to-speech dialogue model with frozen llm. In *International Conference on Machine Learning*, pages 63345–63354. PMLR, 2025.
- [63] Aohan Zeng, Zhengxiao Du, Mingdao Liu, Kedong Wang, Shengmin Jiang, Lei Zhao, Yuxiao Dong, and Jie Tang. Glm-4-voice: Towards intelligent and human-like end-to-end spoken chatbot. *arXiv preprint arXiv:2412.02612*, 2024.
- [64] Wenxi Chen, Ziyang Ma, Ruiqi Yan, Yuzhe Liang, Xiquan Li, Ruiyang Xu, Zhikang Niu, Yanqiao Zhu, Yifan Yang, Zhanxun Liu, et al. Slam-omni: Timbre-controllable voice interaction system with single-stage training. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 2262–2282, 2025.
- [65] Chaoyou Fu, Haojia Lin, Xiong Wang, Yi-Fan Zhang, Yunhang Shen, Xiaoyu Liu, Haoyu Cao, Zuwei Long, Heting Gao, Ke Li, et al. Vita-1.5: Towards gpt-4o level real-time vision and speech interaction. *arXiv preprint arXiv:2501.01957*, 2025.
- [66] Tianpeng Li, Jun Liu, Tao Zhang, Yuanbo Fang, Da Pan, Mingrui Wang, Zheng Liang, Zehuan Li, Mingan Lin, Guosheng Dong, et al. Baichuan-audio: A unified framework for end-to-end speech interaction. *arXiv preprint arXiv:2502.17239*, 2025.

- [67] Ding Ding, Zeqian Ju, Yichong Leng, Songxiang Liu, Tong Liu, Zeyu Shang, Kai Shen, Wei Song, Xu Tan, Heyi Tang, et al. Kimi-audio technical report. *arXiv preprint arXiv:2504.18425*, 2025.
- [68] Qingkai Fang, Yan Zhou, Shoutao Guo, Shaolei Zhang, and Yang Feng. Llama-omni 2: Llm-based real-time spoken chatbot with autoregressive streaming speech synthesis. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 18617–18629, 2025.
- [69] Boyong Wu, Chao Yan, Chen Hu, Cheng Yi, Chengli Feng, Fei Tian, Feiyu Shen, Gang Yu, Haoyang Zhang, Jingbei Li, et al. Step-audio 2 technical report. *arXiv preprint arXiv:2507.16632*, 2025.
- [70] Yi-Jen Shih, Desh Raj, Chunyang Wu, Wei Zhou, SK Bong, Yashesh Gaur, Jay Mahadeokar, Ozlem Kalinli, and Mike Seltzer. Can speech llms think while listening? *arXiv preprint arXiv:2510.07497*, 2025.
- [71] Siddhant Arora, Jinchuan Tian, Hayato Futami, Jiatong Shi, Yosuke Kashiwagi, Emiru Tsunoo, and Shinji Watanabe. Chain-of-thought reasoning in streaming full-duplex end-to-end spoken dialogue systems. *arXiv preprint arXiv:2510.02066*, 2025.
- [72] Alexander Amini, Anna Banaszak, Harold Benoit, Arthur Böök, Tarek Dakhran, Song Duong, Alfred Eng, Fernando Fernandes, Marc Härkönen, Anne Harrington, et al. Lfm2 technical report. *arXiv preprint arXiv:2511.23404*, 2025.
- [73] Dong Zhang, Gang Wang, Jinlong Xue, Kai Fang, Liang Zhao, Rui Ma, Shuhuai Ren, Shuo Liu, Tao Guo, Weiji Zhuang, et al. Mimo-audio: Audio language models are few-shot learners. *arXiv preprint arXiv:2512.23808*, 2025.
- [74] Rajarshi Roy, Jonathan Raiman, Sang-gil Lee, Teodor-Dumitru Ene, Robert Kirby, Sungwon Kim, Jaehyeon Kim, and Bryan Catanzaro. Personaplex: Voice and role control for full duplex conversational speech models. *arXiv preprint arXiv:2602.06053*, 2026.
- [75] OpenBMB. Minicpm-o: A gemini 2.5 flash level mllm for vision, speech, and full-duplex multimodal live streaming on your phone. <https://github.com/OpenBMB/MiniCPM-o>, 2026. GitHub repository.
- [76] Zalán Borsos, Raphaël Marinier, Damien Vincent, Eugene Kharitonov, Olivier Pietquin, Matt Sharifi, Dominik Roblek, Olivier Teboul, David Grangier, Marco Tagliasacchi, et al. Audiolm: a language modeling approach to audio generation. *IEEE/ACM transactions on audio, speech, and language processing*, 31:2523–2533, 2023.
- [77] Pooneh Mousavi, Gallil Maimon, Adel Moumen, Darius Petermann, Jiatong Shi, Haibin Wu, Haici Yang, Anastasia Kuznetsova, Artem Ploujnikov, Ricard Marxer, et al. Discrete audio tokens: More than a survey! *arXiv preprint arXiv:2506.10274*, 2025.
- [78] Xin Zhang, Dong Zhang, Shimin Li, Yaqian Zhou, and Xipeng Qiu. Spechtokenizer: Unified speech tokenizer for speech language models. In *The Twelfth International Conference on Learning Representations*.

## A Author Contributions

All authors contributed significantly to the design of the method, benchmark construction, evaluation, and the writing and refinement of the paper. While all authors were involved in multiple aspects of the project, their primary contributions are summarized below:

**Kai-Wei Chang.** Proposed the initial idea of time-controllable SDMs and the TiCo training framework (SFT + RL), and led the main writing of the paper.

**Wei-Chih Chen.** Proposed the design of TiCo-Bench, constructed the dataset and evaluation protocols, and conducted extensive evaluations of various models on the benchmark.

**En-Pei Hu.** Developed the cascaded methods and evaluation protocols in TiCo-Bench, and explored reinforcement learning approaches in TiCo.

**Hung-yi Lee and James Glass.** Provided overall guidance on the project, contributed deep technical expertise, helped shape the research direction, and offered critical feedback on the experimental methodology.

## B Training Details

**Stage 1: Time-Awareness SFT.** We fine-tune Qwen2.5-Omni-7B with LoRA ( $r=8, \alpha=16$ ) on all linear layers, keeping the vision encoder frozen. The training set consists of 4,000 samples (400 held out for validation). We train for 5 epochs with a batch size of 2 per GPU  $\times$  4 GPUs and gradient accumulation of 4 steps (effective batch size 32), using a cosine learning rate schedule with peak  $5 \times 10^{-5}$  and 10% warmup. Maximum sequence length is 1,024 tokens. Training uses bfloat16 precision with gradient checkpointing.

**Stage 2: Time-Controllable GRPO with CHORD.** Starting from the Stage 1 checkpoint, we apply GRPO with CHORD [48] to optimize duration controllability. The LoRA configuration uses  $r=8, \alpha=32$ . We train for 3 epochs with a per-GPU batch size of 1 and gradient accumulation of 8 (effective batch size 32), learning rate  $5 \times 10^{-6}$  with cosine schedule and 10% warmup. Each prompt generates  $G=4$  candidate completions with maximum completion length of 512 tokens. The clipping parameter is  $\varepsilon=0.2$  and the KL penalty coefficient is  $\beta=0.04$ .

**Reward Design.** The main reward function is

$$\mathcal{R}_{\text{main}}^{(g)} = F\left(t_{\text{inst}} - t_{\text{last}}^{(g)}\right), \quad (15)$$

where  $t_{\text{inst}}$  is the target duration specified in the instruction and  $t_{\text{last}}^{(g)}$  is the duration indicated by the final generated time marker in  $\tilde{\mathbf{z}}^{(g)}$ . The function  $F$  is defined as a Gaussian:

$$F(\Delta t) = \exp\left(-\frac{(\Delta t)^2}{2\sigma^2}\right), \quad (16)$$

where  $\sigma$  controls the tolerance to duration deviations. In our experiments, we set  $\sigma = 5$ .

We further incorporate auxiliary reward functions to prevent reward hacking:

- **Presence reward**  $\mathcal{R}_{\text{pres}}^{(g)}$ : encourages the model to generate at least one Spoken Time Marker,

$$\mathcal{R}_{\text{pres}}^{(g)} = \mathbb{I}[M_g \geq 1], \quad (17)$$

where  $M_g$  denotes the number of time markers in  $\tilde{\mathbf{z}}^{(g)}$ .

- **Monotonicity reward**  $\mathcal{R}_{\text{mono}}^{(g)}$ : encourages generated time markers to be strictly increasing. We compute the fraction of consecutive pairs that are strictly increasing:

$$\mathcal{R}_{\text{mono}}^{(g)} = \frac{1}{M_g - 1} \sum_{j=1}^{M_g - 1} \mathbb{I}[t_{j+1}^{(g)} > t_j^{(g)}]. \quad (18)$$

- **Repetition penalty**  $\mathcal{R}_{\text{rep}}^{(g)}$ : penalizes repeated time marker values:

$$\mathcal{R}_{\text{rep}}^{(g)} = -\left(1 - \frac{|\{t_1^{(g)}, \dots, t_{M_g}^{(g)}\}|}{M_g}\right), \quad (19)$$

where  $|\cdot|$  denotes set cardinality. The penalty is 0 when all markers are unique and  $-1$  when all are identical.

- **Copy penalty**  $\mathcal{R}_{\text{copy}}^{(g)}$ : penalizes non-final time markers that trivially copy the instructed duration  $t_{\text{inst}}$ :

$$\mathcal{R}_{\text{copy}}^{(g)} = -\frac{1}{M_g} \sum_{j=1}^{M_g-1} \mathbb{I} \left[ |t_j^{(g)} - t_{\text{inst}}| < \tau \right], \quad (20)$$

where  $\tau=0.5$  s is the tolerance threshold. The final marker  $t_{M_g}^{(g)}$  is excluded since matching the target duration at the end is the desired behavior.

The overall reward for the  $g$ -th sample is

$$R^{(g)} = \mathcal{R}_{\text{main}}^{(g)} + \mathcal{R}_{\text{pres}}^{(g)} + \mathcal{R}_{\text{mono}}^{(g)} + \mathcal{R}_{\text{rep}}^{(g)} + \mathcal{R}_{\text{copy}}^{(g)}. \quad (21)$$

Note that  $\mathcal{R}_{\text{rep}}^{(g)}$  and  $\mathcal{R}_{\text{copy}}^{(g)}$  are non-positive by construction, so no explicit subtraction is needed.

**CHORD.** CHORD interleaves SFT updates with GRPO updates using a mixing coefficient  $\mu$  that decays from  $\mu_{\text{peak}}=0.8$  to  $\mu_{\text{valley}}=0.3$  over 500 steps, preventing catastrophic forgetting of general conversational ability. Both stages are trained on 4 NVIDIA A6000 GPUs, and the entire two-stage pipeline completes in less than one day.

## C Cascaded System Prompt Templates

We use a unified system prompt across GPT and Qwen for the cascaded LLM baseline.

### LLM System Prompt for Cascaded System

*You are writing a final script for text-to-speech (TTS). Your response will be synthesized directly into speech. Follow the duration instruction as strictly as possible. Output only the final spoken text, with natural punctuation. Do not output markdown, bullets, JSON, XML tags, stage directions, or extra commentary. Do not mention these instructions.*

## D Generalization Study of Textual Queries

During training, in both the first and second stages, the model always receives speech queries as input; that is, all questions are presented in spoken form. Here, we evaluate whether the trained model can generalize to queries provided in textual form. The results show that the model maintains strong performance even when the input queries are given as text.

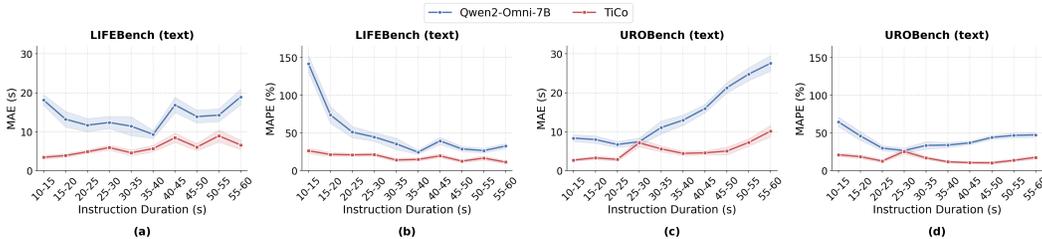


Figure 6: Text benchmarks: duration error of Qwen2-Omni-7B vs. TiCo measured against instructed duration. From left to right: LIFE Bench MAE (s), LIFE Bench MAPE (%), URO Bench MAE (s), URO Bench MAPE (%). Shaded regions indicate  $\pm 1$  SEM.

## E Detailed GPT score in TiCo-Bench subsets

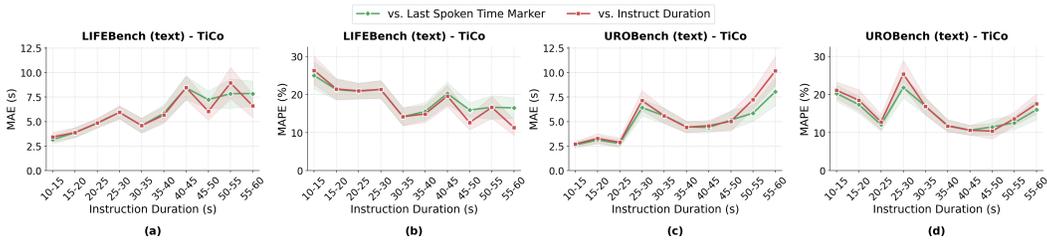


Figure 7: Text benchmarks (TiCo): duration error measured against instructed duration vs. last time marker. From left to right: LIFEbench MAE (s), LIFEbench MAPE (%), UROBench MAE (s), UROBench MAPE (%). Shaded regions indicate  $\pm 1$  SEM.

Table 3: **TiCo-Bench (Speech)** evaluation of time-controlled instruction-following performance on **speech-query** tasks. Results are reported as a scale of 1 to 5. Higher is better.

<i>TiCo-Bench (Speech)</i> Model	Short (10s–30s)		Long (30s–1min)		Overall
	InstructS2S	UROBench	InstructS2S	UROBench	
Cascade (GPT)	<b>4.09</b>	<b>4.18</b>	<b>4.14</b>	<b>4.18</b>	<b>4.15</b>
Cascade (Qwen)	3.41	3.35	3.36	3.43	3.39
GPT-audio	4.07	3.96	4.08	3.42	3.88
Kimi Audio	1.99	1.48	1.79	1.41	1.67
LFM Audio	3.17	2.28	2.92	2.15	2.63
Mimo-Audio	3.22	3.44	3.16	3.45	3.32
Qwen2.5-Omni-7B	3.64	3.63	3.44	3.56	3.57
<b>TiCo (Proposed)</b>	3.39	3.59	3.44	3.83	3.56

Table 4: **TiCo-Bench (Text)** evaluation of time-controlled instruction-following performance on **text-query** tasks. Results are reported as a scale of 1 to 5. Higher is better.

<i>TiCo-Bench (Speech)</i> Model	Short (10s–30s)		Long (30s–1min)		Overall
	LIFEbench	UROBench-text	LIFEbench	UROBench-text	
Cascade (GPT)	<b>2.86</b>	<b>4.18</b>	<b>3.10</b>	<b>4.18</b>	<b>3.58</b>
Cascade (Qwen)	2.32	3.35	2.34	3.43	2.86
GPT-audio	2.55	3.23	2.28	3.05	2.78
LFM Audio	1.93	2.16	1.62	2.08	1.95
Mimo-Audio	1.42	1.53	1.24	1.38	1.39
Qwen2.5-Omni-7B	2.30	3.32	2.10	2.94	2.67
<b>TiCo (Proposed)</b>	2.23	3.33	2.23	3.26	2.76

## F Qualitative Examples

### F.1 How deep is the ocean? (Speech query)

How deep is the ocean?, target duration = 15 s

**Prompt:** *How deep is the ocean? Please respond with exactly 15 seconds.*

**Speech response:**

Well, <0.9 seconds> you know, <1.6 seconds> the ocean is really deep in some places. <3.8 seconds> The Mariana Trench is the deepest part, <7.2 seconds> it's about 36 kilometers deep. <9.4 seconds> But overall, <11.0 seconds> the average depth of the world's oceans is around 3.7 kilometers. <15.0 seconds>

**Target duration:** 15.0 seconds

**Actual duration:** 15.2 seconds

How deep is the ocean?, target duration = 25 s

**Prompt:** *How deep is the ocean? Please respond with exactly 25 seconds.*

**Speech response:**

Well, <1.0 seconds> you know, <1.9 seconds> the ocean is really deep in some places. <4.3 seconds> The Mariana Trench is the deepest part, <7.6 seconds> it's about 36,000 feet or so deep. <10.8 seconds> But overall, <12.6 seconds> the average depth of the world's oceans is around 13,000 feet. <16.4 seconds> It's amazing how vast and deep they are, <19.2 seconds> right? <20.0 seconds> If you want to know more about the ocean's depths or anything else related, <25.0 seconds>

**Target duration:** 25.0 seconds

**Actual duration:** 25.0 seconds

How deep is the ocean?, target duration = 40 s

**Prompt:** *How deep is the ocean? Please respond with exactly 40 seconds.*

**Speech response:**

Well, <1.5 seconds> you know, <2.3 seconds> the ocean is really deep in some places. <4.6 seconds> The Mariana Trench is one of the deepest parts. <7.9 seconds> It's about 36 kilometers deep. <10.8 seconds> But overall, <12.5 seconds> if we're talking about an average depth for the whole ocean, <15.5 seconds> it's around 3.7 kilometers. <18.0 seconds> However, <19.4 seconds> different areas can have very different depths. <22.3 seconds> Some shallow bays might be just a few meters deep while others in the middle of the ocean could be much deeper than that. <29.0 seconds> So, <30.0 seconds> it's not really a fixed number like 3.7 kilometers everywhere. <33.0 seconds> If you want to know more about why the ocean has such varying depths or anything else related, <39.0 seconds> feel free to ask. <40.0 seconds>

**Target duration:** 40.0 seconds

**Actual duration:** 41.6 seconds

### F.2 Discuss an event from history (Speech query)

History event, target duration = 10 s

**Prompt:** *Discuss an event from history. Please respond with exactly 10 seconds.*

**Speech response:**

Well, <1.5 seconds> let's talk about the fall of the Berlin Wall in 1989. <4.2 seconds> It was a huge event that symbolized the end of the Cold War. <7.3 seconds> It brought East and West Germany together again. <10.0 seconds>

**Target duration:** 10.0 seconds  
**Actual duration:** 12.0 seconds

**History event, target duration = 20 s**

**Prompt:** Discuss an event from history. Please respond with exactly 20 seconds.

**Speech response:**

Well, <1.5 seconds> let's talk about the signing of the Magna Carta in 1215. <4.8 seconds> It was really important because it limited the power of the king and protected the rights of the nobles. <9.6 seconds> It also had an impact on later democratic ideas. <13.7 seconds> You know, <15.0 seconds> it set a precedent for the rule of law. <18.0 seconds> So what do you think about this event? <20.0 seconds>

**Target duration:** 20.0 seconds  
**Actual duration:** 19.2 seconds

**History event, target duration = 30 s**

**Prompt:** Please respond with exactly 30 seconds.

**Speech response:**

Well, <1.5 seconds> let's talk about the signing of the Magna Carta in 1215. <4.8 seconds> It was really important because it limited the power of the king to some extent. <9.6 seconds> You know, <10.7 seconds> before that kings could pretty much do whatever they wanted with their subjects' rights. <14.9 seconds> But this document said things like no one can be imprisoned without proper trial. <19.6 seconds> It set a precedent for later legal systems all over the world. <24.0 seconds> And it kind of started the idea of due process. <28.0 seconds> So what do you think about this event? <30.0 seconds>

**Target duration:** 30.0 seconds  
**Actual duration:** 31.1 seconds

### F.3 What is quantum mechanics? (Text query)

**Quantum mechanics, target duration = 25 s**

**Prompt:** What is quantum mechanics? Please respond with exactly 25 seconds.

**Speech response:**

Well, <1.0 seconds> quantum mechanics is this really strange science that deals with really small things like atoms and subatomic particles. <7.0 seconds> It has some wacky ideas, <9.0 seconds> you know, <10.0 seconds> like particles can be in multiple places at once or they can be entangled so that if one changes state the other does too instantaneously no matter how far apart. <23.0 seconds> It's super important for understanding how the world works on a very fundamental level. <25.0 seconds>

**Target duration:** 25.0 seconds  
**Actual duration:** 29.1 seconds

**Quantum mechanics, target duration = 50 s**

**Prompt:** What is quantum mechanics? Please respond with exactly 50 seconds.

**Speech response:**

Well, <1.9 seconds> quantum mechanics is really something special. <4.3 seconds> It's a branch of physics that deals with the behavior of really small things like atoms and subatomic particles. <12.6 seconds> You know, <14.0 seconds> it has some really strange ideas. <16.0 seconds> For example, <17.8 seconds> particles can be in multiple places at once until you measure them. <21.8 seconds> And

there are these things called wave-particle duality where particles act both like waves and particles. <29.2 seconds> Also, <30.6 seconds> the Heisenberg uncertainty principle says that we can't precisely know certain pairs of properties at the same time. <37.8 seconds> It's all very counterintuitive compared to what we see in our everyday world. <43.0 seconds> If you want to learn more about specific aspects or have any other questions, <48.0 seconds>

**Target duration:** 50.0 seconds

**Actual duration:** 44.0 seconds

#### E.4 Why is Mars considered a candidate for human colonization? (Text query)

##### Mars colonization, target duration = 10 s

**Prompt:** *Why is Mars considered a candidate for human colonization? Please respond with exactly 10 seconds.*

**Speech response:**

Well, <1.5 seconds> Mars has resources like water ice, <3.8 seconds> which could be used for life support. <5.4 seconds> It also has an atmosphere that might be modified to make it more habitable. <9.2 seconds> So yeah, <10.0 seconds> it's a good candidate.

**Target duration:** 10.0 seconds

**Actual duration:** 12.0 seconds

##### Mars colonization, target duration = 30 s

**Prompt:** *Why is Mars considered a candidate for human colonization? Please respond with exactly 30 seconds.*

**Speech response:**

Well, <1.5 seconds> Mars has some things going for it. <4.2 seconds> It's closer to Earth than other potential places like Jupiter's moons. <8.6 seconds> There might be water ice on its surface, <11.7 seconds> which could be used for life support systems. <14.9 seconds> Also, <16.3 seconds> its gravity is lower compared to Earth's, <19.0 seconds> making it easier to build structures there. <21.7 seconds> And it has an atmosphere that can potentially be modified over time. <25.5 seconds> So all these factors make Mars a good candidate for human colonization. <30.0 seconds>

**Target duration:** 30.0 seconds

**Actual duration:** 27.9 seconds

##### Mars colonization, target duration = 60 s

**Prompt:** *Why is Mars considered a candidate for human colonization? Please respond with exactly 60 seconds.*

**Speech response:**

Well, <1.5 seconds> Mars is considered a candidate for human colonization for several reasons. <4.9 seconds> First off, <6.2 seconds> it has some resources like water ice at the poles which could be used to make oxygen and hydrogen for fuel. <13.7 seconds> Also, <15.0 seconds> there might be minerals on Mars that we can mine in the future. <18.5 seconds> The gravity on Mars is about one-third of Earth's, <22.2 seconds> so it would be easier for humans to move around compared to being in zero gravity. <27.0 seconds> And scientists think that over time, <30.0 seconds> we could potentially modify the Martian environment to be more habitable. <34.0 seconds> For example, <35.6 seconds> building habitats that can protect against radiation and extreme temperatures. <40.0 seconds> There is also the possibility of finding signs of past life on Mars, <44.0 seconds> which makes it even more interesting. <46.0 seconds>

If you want to know more about this or have other questions related to space exploration, <50.0 seconds> feel free to ask. <51.0 seconds>

**Target duration:** 60.0 seconds

**Actual duration:** 57.4 seconds

## G Spoken Dialogue Model Survey

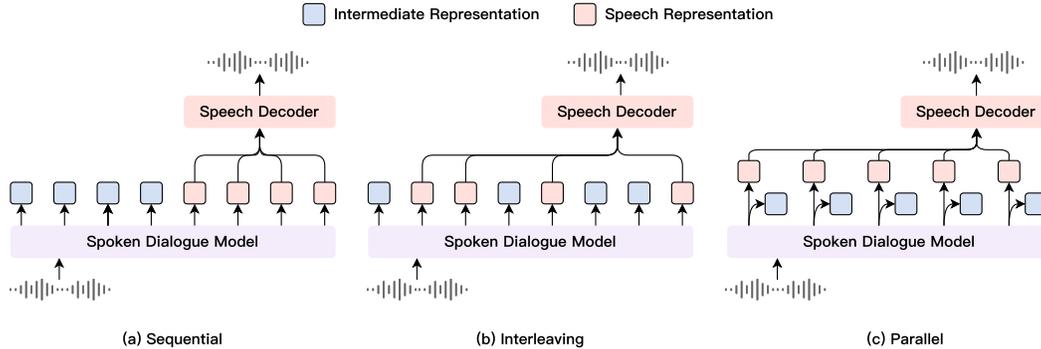


Figure 8: Illustration of different generation patterns in spoken dialogue models (SDMs): (a) Sequential, (b) Interleaved, and (c) Parallel.

Table 5 surveys representative spoken dialogue models (SDMs), including their Intermediate Representations (**IR**), target speech representations (**Speech Rrep.**), and the generation patterns (**Pattern**) that describe how intermediate representations and speech representations are processed during speech response generation. Readers may refer to the spoken language model (SLM) survey paper [1] for a more detailed discussion on speech representation and generation pattern.

**Intermediate Representation (IR).** With the emergence of text-based large language models (LLMs) demonstrating strong reasoning capabilities, modern spoken dialogue models (SDMs) increasingly adopt LLMs to generate speech responses, using text as an intermediate representation for semantic planning.

Text-based IR offers high versatility and can serve multiple purposes, as summarized in Table 5, including style control (+S.), reasoning (+R.), tool calling (+Tool), and direct guidance of the target speech content.

**Pattern.** The intermediate representation and the target speech representations can be generated under several design patterns, each leading to different trade-offs in terms of efficiency, latency, and the degree to which speech generation is conditioned on the intermediate representation. For simplicity, we assume text as the intermediate representation in the following discussion, and provide an illustration in Figure 8.

- **Sequential:** Text is generated first, followed by speech tokens. Chunking strategies can be incorporated to support streaming generation.
- **Parallel:** Text and speech tokens are generated simultaneously. In this setting, the hidden representations of a text LLM are typically used to predict text tokens and speech tokens through separate prediction networks. Frame-level operations can further be introduced to realize delay patterns.
- **Interleaved:** Text and speech tokens are arranged in a single interleaved sequence, typically modeled by a single LLM, allowing speech representations to be conditioned more directly on text representations.

**Speech Representations.** The goal of a spoken dialogue model is to generate an appropriate spoken response, typically represented as a sequence of speech tokens. These tokens can be further synthesized into waveforms using a pre-trained vocoder or an audio codec decoder.

- **Phonetic tokens.** Phonetic tokens are obtained by quantizing speech encoder representations (e.g., via K-means), such as those extracted from self-supervised speech models (e.g., HuBERT) or foundation ASR models (e.g., Whisper encoders). They primarily capture phonetic and linguistic content, while containing relatively limited acoustic information such as speaker identity or environmental characteristics. In prior work, they are also referred to as *semantic tokens* [76].

When *phonetic tokens* are used as the target speech representation, an additional vocoder (e.g., HiFi-GAN or flow-matching decoders) is typically required to incorporate speaker identity and speaking style, as these attributes are not explicitly encoded.

- **Acoustic tokens.** Acoustic tokens [77, 25, 26] are derived from neural speech codec models trained with reconstruction objectives. These models typically employ multiple hierarchical codebooks based on residual vector quantization (RVQ).

When *acoustic tokens* are generated, a pre-trained audio codec decoder can be directly used for waveform synthesis. Recently, there has been a growing trend toward distilling phonetic information into the early layers of acoustic tokens, aiming to preserve phonetic structure while maintaining rich acoustic detail [78, 59].

*(Please find the survey table on the following page.)*

Table 5: Spoken dialogue models (SDMs) with speech input and speech output, ordered by their first public release time. **Date:** First released date. **IR:** Intermediate representation used in the SDM. **Speech Rep.:** Speech representation (prediction target) used by the model. **Pattern:** The pattern of how the intermediate and speech representations are generated.

Model	Date	IR	Speech Rrep.	Pattern	Notes
dGSLM [56]	2022-03	-	Phonetic token	Direct	“Dual-tower” architecture for dual channel full-duplex modeling. Direct modeling of two-channel phonetic tokens.
SpeechGPT [57]	2023-05	Text (+R.)	Phonetic token	Sequential	Using “Chain-of-Modality”. Expands LLM vocabulary with phonetic tokens.
SPIRIT-LM [58]	2024-02	Text (+S.)	Phonetic token	Interleaved	Interleaves text and speech in one stream; expressive version adds pitch and style tokens.
Moshi [59]	2024-07	Text	Acoustic token	Parallel	Dual-channel full-duplex model. Parallel decoding text and acoustic tokens with delay pattern.
LLaMA-Omni [49]	2024-09	Text	Phonetic token	Parallel	CTC speech decoder maps LLM response states to phonetic tokens for streaming speech synthesis.
SyncLLM [60]	2024-09	-	Phonetic token	Direct	Interleaving user speech and model speech for full-duplex dialogue.
Mini-Omni2 [61]	2024-10	Text	Acoustic token	Parallel	Parallel decoding with delay pattern.
Freeze-Omni [62]	2024-11	Text	Acoustic token	Sequential	TDM-based full-duplex interaction; speech decoder conditioned on text tokens and LLM hidden states.
GLM-4-Voice [63]	2024-12	Text	Phonetic token	Interleaved	Single speech codebook paired with flow matching speech decoder.
SLAM-Omni [64]	2024-12	Text	Phonetic token	Parallel	“Semantic group modeling” enables generating multiple phonetic tokens per text token.
VITA-1.5 [65]	2025-01	Text	Acoustic token	Sequential	NAR + AR speech decoder taking LLM embedding as input.
Baichuan-Audio [66]	2025-02	Text	Acoustic token	Interleaved	Text-guided speech generation with an independent audio head.
Qwen2.5-Omni [23]	2025-03	Text	Acoustic token	Sequential	Thinker-Talker architecture. Supports visual modality.
Kimi-Audio [67]	2025-04	Text	Phonetic token	Parallel	Shared LLM with text head and audio head.
LLaMA-Omni 2 [68]	2025-05	Text	Phonetic token	Parallel	Gate fusion of LLM hidden states and text tokens for improved speech quality.
Step-Audio 2 [69]	2025-07	Text (+R., Tool)	Phonetic token	Interleaved	Using “multi-modal RAG” to support grounded response and timbre/style control.
STITCH [21]	2025-07	Text (+R.)	Phonetic token	Interleaved	Backbone: GLM-4-Voice. Various interleaving text, reasoning, speech patterns are discussed in the paper.
Qwen3-Omni [24]	2025-09	Text (+R.)	Acoustic token	Sequential	Thinker-Talker architecture. Supports explicit thinking mode and compatible with tool calling. Supports visual modality.
Moshi-CoT [70]	2025-10	Text (+R.)	Acoustic token	Parallel	CoT-tuned Moshi performs text reasoning in the “text monologue” stream to enable “thinking while listening” paradigm.
SCoT [71]	2025-10	Text	Acoustic token	Interleaved	CoT framework for SDMs. Blockwise streaming full-duplex model.
Streaming RAG [22]	2025-10	Text (+Tool)	Acoustic token	Sequential	Enables the SDM to trigger tool queries in parallel with the user’s speech.
LFM2-Audio [72]	2025-11	Text	Acoustic token	Interleaved	Supports both interleaved and sequential patterns, adapting to different tasks.
Mimo-Audio [73]	2025-12	Text (+R.)	Acoustic token	Interleaved	Interleaving text tokens and “audio patches”, which includes a delay pattern.
PersonaPlex [74]	2026-01	Text	Acoustic token	Parallel	Followed the Moshi architecture. Dual-channel full-duplex model.
MiniCPM-o 4.5 [75]	2026-02	Text (+R.)	Acoustic token	Interleaved	TDM full-duplex model. Supports visual modality.