TiCo: Time-Controllable Spoken Dialogue Model

Anonymous authors
Paper under double-blind review

Abstract

Existing spoken dialogue models (SDMs) can generate high-quality responses, yet they struggle to control response duration. Such control is important for managing service costs, improving user experience, and supporting applications with duration requirements. We introduce **TiCo-Bench**, a benchmark for systematically evaluating duration control across four task categories and response target durations of 10–60 seconds. Evaluations of open-source and commercial models reveal frequent failures to meet explicit duration constraints. To address this gap, we propose **TiCo**, a simple and efficient post-training approach that incorporates duration estimation into semantic planning. TiCo uses **Spoken Time Markers (STMs)**, such as *<10.6 seconds>*, to make estimated elapsed speaking time explicit during autoregressive generation, allowing the model to adjust the remaining content to meet the target duration. Because STMs express speaking time as text, the duration reward is verifiable without synthesizing speech during training, and the training data are self-generated, with no external question-answer pairs. Across three backbones, TiCo reduces duration error by 2.7× – 3.5× while preserving response quality. The best TiCo model halves the duration error of the strongest text-planning cascade system, and the response is more natural than a cascade system that additionally rescales the TTS speaking rate. Demo page: `https://tico-9wu.pages.dev/`

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) (Arora et al.; Cui et al., 2025; Ji et al., 2024), which are gaining increasing attention in real-world applications, such as personal assistants, wearable devices, and healthcare systems (de Barcelos Silva et al., 2020; Adams et al., 2025). In these scenarios, a response must not only be accurate and natural, but often also strictly bounded in duration. For example, a user with one minute before the next meeting wants that minute used: a summary that is as complete as it can be in the time available. In dubbing and simultaneous interpretation, the spoken output has to match the length of the source utterance in order to stay aligned with it. Similarly, in hands-busy or emergency scenarios, a voice assistant may need to deliver brief yet informative instructions before the moment to act on it has passed. What these settings ask for is not a single response length but *controllability*: whoever deploys the system, or the user, states the constraint that applies, and the model meets it[1]. Despite its importance, response-elapsed time controllability remains 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 (Zhang et al., 2025a). Moreover, LLM outputs often exhibit verbosity or length bias, a phenomenon associated with preference-based evaluation and alignment (Dubois et al., 2024; Hu et al., 2025). This tendency weakens instruction-following capability and negatively affects user experience. While benchmarks, prompting, and training strategies have begun to address length controllability in text LLMs (Zhang et al., 2025a; Xie and Lee, 2025; Jie et al., 2023; Song et al., 2025), this research direction remains active and continues to attract attention due to its substantial practical importance.

[1]Qualitative instructions such as "briefly" do steer response length, but they leave the realized duration unspecified and vague, which is not something a provider can plan a time budget around.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

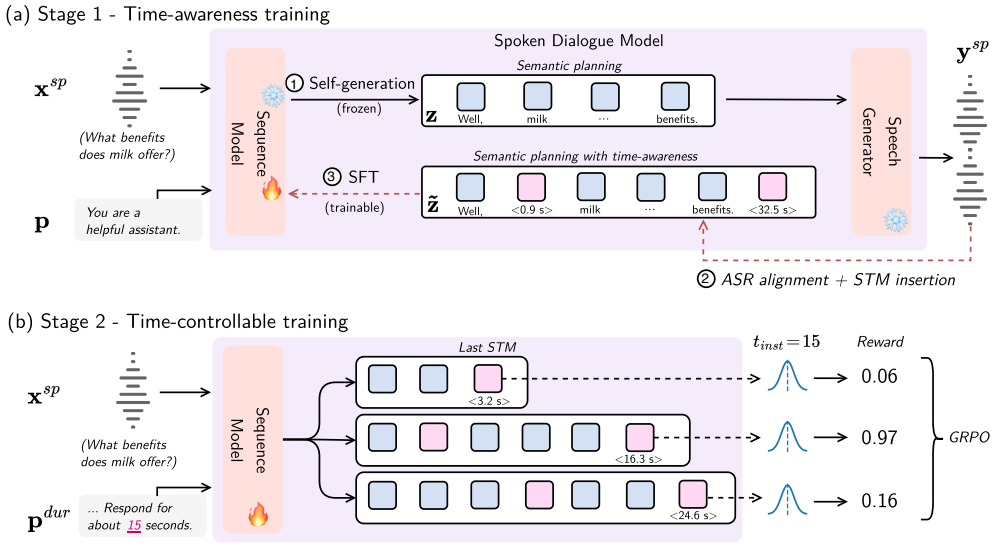


Figure 1: Overview of **TiCo**, a two-stage framework for time-controllable speech generation. **Stage 1 (top):** The model leverages self-generation to produce responses annotated with **Spoken Time Markers (STMs)**, which serve as supervision for learning *time awareness*, i.e., associating intermediate 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.

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 (Klatt, 1976). Moreover, speaking rate may vary across speakers, speaking styles, and communicative conditions, which depends on both speaker and listener (Lindblom, 1990). 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 introduce a simple yet effective post-training approach, **TiCo** (Figure 1), a time-controllable training framework for SDMs to estimate and regulate generated speech duration through **Spoken Time Markers (STMs)**. TiCo follows the standard SFT-RL paradigm and is trained with a two-stage procedure. In the first stage (SFT), 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 (RL), *Reinforcement Learning with Verifiable Rewards (RLVR)* (Lambert et al., 2024; Shao et al., 2024) is applied, where rewards are automatically verified based on the STMs, to further shape the response distribution and improve the instruction following capability. This stage encourages the model to better satisfy target time constraints while preserving the response quality.

The paper is organized as follows. **Sec. 2** reviews related work. **Sec. 3** introduces TiCo-Bench. **Sec. 4** presents TiCo, a simple approach to SDM duration control. **Sec. 5** describes the experimental setup. **Sec. 6** presents our main findings: **(1) Sec. 6.1** shows that TiCo follows duration instructions and can be applied to different SDM architectures. **(2) Sec. 6.2** shows that TiCo generalizes to various instruction types, satisfying upper bound, lower bound, and range constraints. **(3) Sec. 6.3** shows that STM benefits time-aware semantic planning and that TiCo is efficient in its use of training data. Finally, **Sec. 7** analyzes the acoustic and semantic properties of TiCo's responses and presents qualitative examples.

2 RELATED WORKS

2.1 SPOKEN DIALOGUE MODELS

Spoken Dialogue Models (SDMs) (Arora et al.; Cui et al., 2025) 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 (Glass, 1999; Ji et al., 2024).

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[2]. 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 (Chiang et al., 2026), tool calling (Arora et al., 2025a) to leverage external modules, and more direct guidance over spoken content (Xu et al., 2025a;b). 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 (Arora et al.; Guo et al., 2025; Wu et al., 2024)), and subsequently synthesized into a waveform. We provide a survey of representative SDMs and their intermediate representations in Appendix K.

2.2 CONTROLLABILITY AND TIME AWARENESS IN SPEECH GENERATION

Recently, several benchmarks have begun to evaluate SDMs beyond response quality, incorporating dimensions such as speaking style (Yang et al., 2025), interactivity (Lin et al., 2025a;b), controllability (Yang et al., 2025) and time-awareness (Chang et al., 2025). 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 an efficient post-training approach.

It is worth noting that TiCo differs fundamentally from *duration modeling* in TTS systems (Ren et al., 2021; Xie et al., 2025). 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* (Sridhar et al., 2025), 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?") (Wang et al., 2025a). In contrast, TiCo focuses on time awareness in the *generation process*, rather than temporal comprehension of the input.

2.3 LENGTH-CONTROLLED LARGE LANGUAGE MODELS

Length control in text LLMs has been studied along three lines. Training-free or decoding-time methods enforce length via sampling (Gu et al., 2025), zero-shot prompting (Retkowski and Waibel, 2025), or EOS-token reweighting (Belligoli et al., 2025). Instruction-tuning approaches inject length-tracking signals into generation, such as distance-to-target encodings (Butcher et al., 2025), latent tracking tokens (Song et al., 2025), or explicit positional markers (Wang et al., 2024). A third line uses RL or preference optimization to decouple length bias from response quality (Liu et al., 2024; Li et al., 2025a) and to control reasoning length, either by enforcing concise steps (Liu et al., 2026) or by extending the trajectory for harder problems (Liu et al., 2025; Aggarwal and Welleck, 2025; Jie et al., 2023). All these methods operate on text, where word count is a direct proxy for length. TiCo instead targets spoken dialogue, where word count is only a loose proxy for speech duration since the realized timing depends on paralinguistic factors beyond the textual output.

[2]This challenge has been largely explored in prior work such as the *"Textless NLP" paradigm* (Lakhotia et al., 2021; Polyak et al., 2021; Hassid et al., 2023; Chang et al., 2022; 2024; Popuri et al., 2022), where quantized speech representations are treated as "pseudo text" to improve training efficiency and efficacy.

3 TiCo-Bench

Dataset. We introduce TiCo-Bench to evaluate an SDM's ability to follow explicit duration constraints, a capability overlooked by existing benchmarks. TiCo-Bench is organized into four task categories, each drawing queries from one publicly available source: *Question Answering* (**QA**) from InstructS2S (Fang et al., 2025a), *Reasoning* (**REA**) from URO-Bench (Yan et al., 2025), *Creative Generation* (**CRE**) from databricks-dolly-15k (Conover et al., 2023), and *Summarization* (**SUM**) from Extreme Summarization (XSum) (Narayan et al., 2018). InstructS2S and URO-Bench provide native speech queries; Dolly-15k and XSum are text only and synthesized via a TTS pipeline with ASR-based verification. See Appendix B for details.

Each base query is paired with an explicit time-control instruction specifying a target duration t_{inst}, and instantiated under both a **Short** setting (10–30 s) and a **Long** setting (30–60 s), giving 2,000 evaluation samples in total. Targets within each regime are sampled uniformly.

Metrics. We evaluate duration controllability with the Mean Absolute Percentage Error (MAPE, in %), defined as the average of $|d - t_{\text{inst}}|/t_{\text{inst}}$ over evaluation samples, where d is the realized speech response duration and t_{inst} is the instructed target duration. Because MAPE normalizes the error by the target duration, it is comparable across target durations. Lower is better. We additionally report the Mean Absolute Error (MAE, in seconds) in Appendix D.

We further evaluate the quality of the generated responses along two dimensions. For *semantic quality*, we report **GPT-score**: each spoken response is transcribed with ASR, and GPT-5-mini (OpenAI, 2025a) rates the transcript as a response to the query on a 5-point scale. For *acoustic quality*, we report **UTMOS** (Saeki et al., 2022), with a 5-point Mean Opinion Score (MOS) that human listeners would assign for naturalness, as a measure of perceptual quality. We further report a prosodic analysis of speaking rate, pausing, and phone duration in Section 7.

4 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 planning and low-level speech synthesis. Concretely, a **sequence model** p_θ first generates an intermediate representation conditioned on the user input:

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

Depending on the architecture, $\mathbf{z}$ may include text tokens to be spoken, reasoning tokens, tokens for tool calls, and potentially interleaved with speech tokens. The final spoken response is then generated by a **speech generator** q_ϕ:

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

Note that different architectures impose different conditional independence assumptions on Eq. (2). In cascaded systems, the speech synthesis module has no access to the original user speech or instruction, reducing the generation to $q_\phi(\mathbf{y}^{\text{sp}} \mid \mathbf{z})$. In end-to-end models, the generation of $\mathbf{y}^{\text{sp}}$ may additionally depend on $\mathbf{x}^{\text{sp}}$ and $\mathbf{p}$[3]. In this paper, we focus on end-to-end models, which retain access to $\mathbf{x}^{\text{sp}}$ and $\mathbf{p}$ during speech generation and therefore produce more natural and user-adaptive spoken responses.

4.1 TiCo Stage 1: Time-Awareness Training

This stage (Figure 1 (a)) trains the SDM to incorporate *Spoken Time Markers (STMs)* into the semantic planning process. STMs are in plain text form, e.g., `<6.8 seconds>`, and mark the estimated cumulative speaking time up to that point in the intermediate representation $\mathbf{z}$. Interleaving STMs with $\mathbf{z}$ turns the semantic plan into a time-aware one: The model generates not just the content to be spoken. At each STM, it also predicts how much speaking time will have elapsed by that point in the final spoken response $\mathbf{y}^{\text{sp}}$. This prediction is made under the same speech input $\mathbf{x}^{\text{sp}}$ and instruction $\mathbf{p}$ that condition speech generation.

[3]For example, in Qwen-Omni's "Thinker-Talker" design (Xu et al., 2025a;b).

Training data. We obtain the STM supervision by *self-generation*, without the need for question–answer pairs. Let $\mathcal{D} = \{(\mathbf{x}^{\mathrm{sp}}, \mathbf{p})\}$ denote a pool of input speech query–instruction pairs. For each input $(\mathbf{x}^{\mathrm{sp}}, \mathbf{p}) \in \mathcal{D}$, the model freely generates an intermediate representation $\mathbf{z} = [z_1, \ldots, z_N]$ of N tokens and its spoken response $\mathbf{y}^{\mathrm{sp}}$. ASR alignment timestamps each token of $\mathbf{z}$ with the time it is spoken in $\mathbf{y}^{\mathrm{sp}}$. At each clause and sentence boundary of $\mathbf{z}$, we insert a marker t_j carrying the timestamp of that boundary. This gives the time-annotated sequence

$$\tilde{\mathbf{z}} = [\, z_1, \ldots, z_i, t_j, \ldots, z_N, t_M \,], \tag{3}$$

where M is the number of STMs. M scales with the number of sentences rather than the number of text tokens, and is about an order of magnitude smaller than N. Training on $\mathcal{D}_{\mathrm{SFT}} = \{(\mathbf{x}^{\mathrm{sp}}, \mathbf{p}, \tilde{\mathbf{z}})\}$ is then a standard next-token objective over $\tilde{\mathbf{z}}$, with the loss function:

$$\mathcal{L}_{\mathrm{SFT}} = -\,\mathbb{E}_{(\mathbf{x}^{\mathrm{sp}}, \mathbf{p}, \tilde{\mathbf{z}}) \sim \mathcal{D}_{\mathrm{SFT}}} \Big[\textstyle\sum_{n=1}^{|\tilde{\mathbf{z}}|} \log p_\theta\big(\tilde{z}_n \mid \tilde{z}_{<n}, \mathbf{x}^{\mathrm{sp}}, \mathbf{p}\big) \Big]. \tag{4}$$

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 (Lu et al., 2025).

4.2 TiCo Stage 2: Time-Controllable Training

This stage (Figure 1 (b)) further trains the model to follow time-constrained instructions. We augment the textual instruction $\mathbf{p}$ with a target duration t_{inst} and denote the result by $\mathbf{p}^{\mathrm{dur}}$. We apply reinforcement learning to the sequence generator $p_\theta(\tilde{\mathbf{z}} \mid \mathbf{x}^{\mathrm{sp}}, \mathbf{p}^{\mathrm{dur}})$, with GRPO (Shao et al., 2024). For each input $(\mathbf{x}^{\mathrm{sp}}, \mathbf{p}^{\mathrm{dur}})$ we sample a group of G candidate sequences $\tilde{\mathbf{z}}^{(1)}, \ldots, \tilde{\mathbf{z}}^{(G)}$ from the old policy, and score each of them with the reward below.

Reward design. The main reward measures how close the planned duration is to the target:

$$\mathcal{R}^{(g)}_{\mathrm{main}} = F(t_{\mathrm{inst}} - t^{(g)}_{\mathrm{last}}), \tag{5}$$

where $t^{(g)}_{\mathrm{last}}$ is the time indicated by the final STM in $\tilde{\mathbf{z}}^{(g)}$, and $F(\Delta t) = \exp(-(\Delta t)^2/(2\sigma^2))$ is a Gaussian kernel with σ controlling the tolerance to duration errors. This reward is computed directly from the generated text, without synthesizing speech inside the training loop.

A grounding term $\mathcal{R}^{(g)}_{\mathrm{ground}}$ is introduced to prevent reward hacking, keeping the intermediate markers on a valid, increasing timeline rather than copied or repeated values. Its form is given in Appendix A, and the reward ablation in Appendix H. The overall reward is $R^{(g)} = \mathcal{R}^{(g)}_{\mathrm{main}} + \mathcal{R}^{(g)}_{\mathrm{ground}}$. We then optimize p_θ with the standard GRPO objective, using group-relative advantages computed from $\{R^{(g)}\}_{g=1}^{G}$.

5 Experimental Setup

5.1 Baselines

We compare against three categories of baselines in TiCo-Bench: (1) Open-sourced SDMs – Kimi Audio (Ding et al., 2025), LFM Audio (Amini et al., 2025), MiMo-Audio (Zhang et al., 2025b), Step-Audio-2-mini (Wu et al., 2025), Qwen2.5-Omni-7B (Xu et al., 2025a), Qwen3-Omni-30B (Xu et al., 2025b), (2) Commercial models – GPT-audio [4], and (3) Cascaded systems – Cascade(Qwen) and Cascade(GPT). For the strong cascaded 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 (TTS) system to synthesize the corresponding speech. Specifically, we utilize GPT-5.2 (OpenAI, 2025b) as a commercial LLM and Qwen2.5-7B-Instruct (Team, 2024) as a representative open-source language model. IndexTTS-2 (Zhou et al., 2026) is employed to generate high-quality speech from the LLM response. Detailed prompts used for the cascaded system can be found at the Appendix E.1. To ensure the evaluation reflects generation quality, all SDMs are allocated a sufficiently large token budget to cover responses of up to 1 minute of speech.

[4] https://developers.openai.com/api/docs/models/gpt-audio

Table 1: **TiCo-Bench** evaluation on **speech-query** tasks under Short (10–30 s) and Long (30–60 s) settings. Model categories: Cascaded, Commercial, Open-sourced, Proposed. Per task (QA, REA, CRE, and SUM) metric is MAPE, the lower the better. *For Kimi Audio on the SUM subset, audio clips exceeding the model's effective context window are excluded from evaluation.

TiCo-Bench (Speech)	Short (10s–30s)				Long (30s–60s)				Overall		
Model	**QA**	**REA**	**CRE**	**SUM**	**QA**	**REA**	**CRE**	**SUM**	**MAPE ↓**	**GPT-score ↑**	**UTMOS ↑**
Cascade (Qwen)	36.52	51.43	35.93	59.19	46.32	65.32	38.90	24.89	46.17	3.18	3.00
Cascade (GPT)	19.90	28.07	19.97	17.10	24.28	39.55	25.81	16.84	25.17	**3.92**	2.91
GPT-audio	82.53	107.91	127.97	58.37	45.96	48.21	49.51	41.01	69.38	3.45	3.72
Kimi Audio	133.17	166.79	156.45	60.65*	60.42	72.43	59.66	46.18*	101.82	1.69	2.43
LFM Audio	89.13	61.46	73.46	48.13	47.66	74.02	53.99	48.07	65.72	2.58	4.01
MiMo-Audio	74.74	75.06	123.27	68.20	45.80	38.18	40.92	31.31	60.31	2.91	3.31
Interleaved SDM											
Step-Audio-2-mini	73.80	59.51	93.52	152.06	48.31	65.57	42.41	37.61	65.57	2.72	**4.51**
+ TiCo	18.39	23.43	20.55	47.98	21.78	23.80	17.74	19.08	22.40	3.01	**4.51**
Sequential SDM											
Qwen2.5-Omni-7B	49.61	45.97	55.99	79.55	32.94	42.13	29.82	24.93	43.33	3.31	4.09
+ TiCo	15.44	17.81	17.32	48.96	11.66	13.92	13.07	13.51	16.19	3.32	4.04
Qwen3-Omni-30B	51.65	50.16	75.86	32.19	30.00	38.26	25.05	34.53	42.06	3.58	4.44
+ TiCo	**13.07**	**11.91**	**12.78**	**17.08**	**11.22**	**11.65**	**11.32**	**10.27**	**12.18**	3.76	4.44

5.2 Training Details

To verify that TiCo is not tied to a particular architecture, we apply it to three backbones: Qwen-2.5-Omni 7B (Xu et al., 2025a) and Qwen-3-Omni 30B (Xu et al., 2025b), which are *sequential* architectures, and Step-Audio-2-mini (Wu et al., 2025), which is an *interleaved* architecture. The full training details are in Appendix A

We sample 4,000 speech questions (*without* the paired response) from InstructS2S (Fang et al., 2025a) as training data, holding out 400 for validation. The training data do not overlap with the test set in TiCo-Bench. Word-level timestamps are obtained with Whisper medium (Louradour, 2023), and an STM is inserted after each clause- or sentence-level punctuation mark (e.g., commas, periods, exclamation marks), carrying the end time of the preceding word. We place markers at punctuation locations for two reasons. A clause boundary is where the model can still decide how much more to say, whereas a time estimate in the middle of a sentence gives it little to act on. Placing a marker after every word would also make the sequence several times longer and slow both training and decoding. Punctuation-level markers keep the overhead small: each response contains 13.3 markers on average, one every 2.7 seconds. The full marker distribution is shown in Appendix A.

6 Results

6.1 TiCo-Bench

Most off-the-shelf SDMs fail to follow duration-constrained instructions. Table 1 shows TiCo-Bench results on speech queries. Every end-to-end SDM we evaluate exceeds 40% MAPE, from 42.06% (Qwen3-Omni-30B) to 101.82% (Kimi-Audio), meaning a 30s request is typically missed by 12s or more. Scale does not close it either: Qwen3-Omni-30B is roughly $4\times$ larger than Qwen2.5-Omni-7B and gives better responses (GPT-score 3.58 vs. 3.31), yet its duration error is similar (42.06% vs. 43.33% MAPE). This indicates that duration controllability does not emerge from scaling alone. Only cascaded systems with an explicit text-length proxy are competitive, with Cascade (GPT) reaching 25.17%. The performance still relies on word count as an estimation proxy for duration, and the same pipeline with a weaker LLM planner degrades to 46.17% (Cascade (Qwen)). A stronger cascade variant rescales the TTS speaking rate to match the target, which reduces duration error but at the cost of speech naturalness (Appendix E.2). Errors are consistently larger in the Short subsets than in the Long subsets, mirroring the verbosity bias exhibited in text LLMs (Dubois et al., 2024).

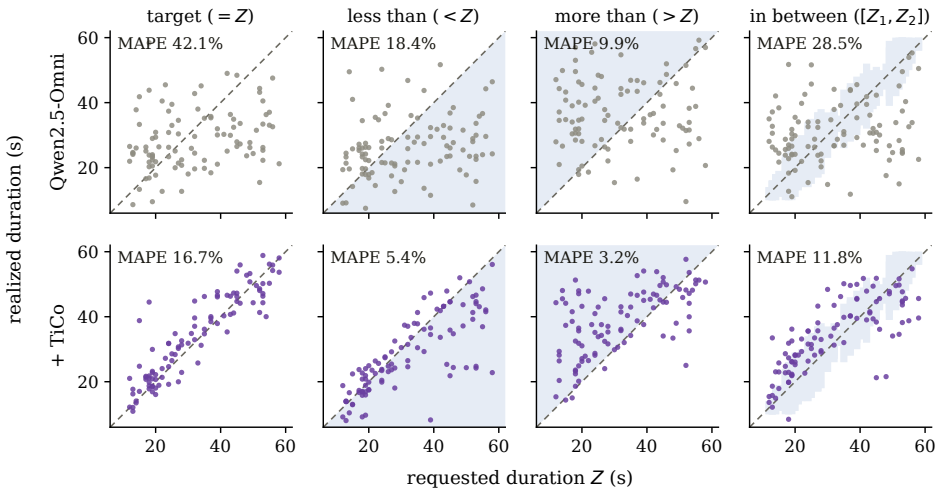


Figure 2: Realized versus requested duration under four instruction types: *target*, *less than*, *more than*, and *in between*. Only *target* is seen during training. Shaded regions mark the admissible range; MAPE is reported per panel and is hinged for the three bounded forms.

With TiCo training, the models can follow duration-constrained instructions. TiCo lowers overall MAPE from 43.33% to 16.19% on Qwen2.5-Omni-7B (2.7×), from 42.06% to 12.18% on Qwen3-Omni-30B (3.5×), and from 65.57% to 22.40% on Step-Audio-2-mini (2.9×) , while response quality holds or improves on every backbone. The improvement holds across all four task categories and both short and long duration subsets, suggesting a generic time-aware planning ability rather than dataset or task-specific adaptation. SUM–Short remains a challenging case. TiCo improves it substantially (79.55% → 48.96% on Qwen2.5-Omni, 152.06% → 47.98% on Step-Audio-2-mini) , yet the error stays above each model's own average. This subset pairs a long spoken input with a short target response (Figure 7b), requiring both long-context understanding and compression. The error is likely bounded by the backbone. With the stronger Qwen3-Omni backbone, TiCo reaches 17.08% on the same subset, the best of all systems.

TiCo preserves backbone models' response quality. Across all three backbones, GPT-score is comparable or higher (3.31 → 3.32 on Qwen2.5-Omni, 3.58 → 3.76 on Qwen3-Omni, 2.72 → 3.01 on Step-Audio-2-mini), and UTMOS is essentially similar (4.09 → 4.04, 4.44 → 4.44, 4.51 → 4.51). This shows that duration control is not achieved by degrading what the model says (semantic) or how it sounds (acoustic). After TiCo training, the *sequence generator* (trained) and the *speech generator* (fixed) also stay in sync, as reflected in the text–speech consistency (Appendix F.2), and the speaking style does not drift from the backbone (Appendix I). For prosody-level analysis and qualitative results, please refer to Section 7.

TiCo applies to different types of SDM architectures. TiCo is not tied to a particular architecture. We show that it applies to both *sequential* SDMs (Qwen-Omni) and *interleaved* ones (Step-Audio). In Step-Audio-2-mini, text and speech are decoded in a single stream, where one text token and four audio tokens are interleaved and generated by the same sequence model. STMs are consequently interleaved with audio tokens rather than planned in a text-only prefix, yet the same TiCo recipe transfers unchanged and cuts MAPE from 65.57% to 22.40%. This 2.9× reduction is larger than the 2.7× TiCo obtains on Qwen2.5-Omni. However, Step-Audio-2-mini + TiCo lags Qwen2.5-Omni + TiCo (22.40% vs. 16.19%), as interleaving multiplies the sequence length per second of speech and the model degrades on long contexts. Due to computational constraints, TiCo refers to the Qwen2.5-Omni-7B backbone in the following experiments unless explicitly specified.

6.2 TiCo generalizes beyond the trained condition

TiCo generalizes to instruction types that are not explicitly trained on. TiCo is trained on matching a specific target duration, where the response should land as close as possible to a requested duration. Here we explicitly test whether it can follow other types of instructions, and we evaluate

four forms: (1) *target*, the trained one, which asks the response to be as close as possible to Z seconds; (2) *less than*, which requires the response to be shorter than Z seconds; (3) *more than*, which requires it to be longer than Z seconds; and (4) *in between*, which requires it to fall inside a window $[Z_1, Z_2]$. We find that TiCo generalizes to all three unseen forms.

We sample 100 audio queries from TiCo-Bench and augment each with three additional instruction types, keeping the original instruction as *target*. All targets range from 10 s to 60 s. We report MAPE for *target* and hinge MAPE for *less than*, *more than*, and *in between*, which is zero inside the admissible range and measures the relative error against the violated bound outside it.

Figure 2 shows the results. TiCo improves on all four instruction types, reducing error by $2.5\times$ on the trained *target* form (42.1% to 16.7%) and by 2.4–$3.4\times$ on the three unseen ones. The gain on the unseen forms is as large as on the trained one. Therefore, Stage 2 training does not only fit the phrasing of the training instruction; it yields a duration-planning ability that transfers across constraint types. This shows that TiCo training has the potential to be applied to various real-world applications for users or model providers. However, we still see a portion of samples that fall away from the expected target duration. Improving the generation stability remains future work.

TiCo generalizes to longer responses and text queries. Due to limited computational resources, TiCo training only covers audio up to $\sim$40 seconds, yet TiCo remains controllable on targets beyond this range: the Long subsets of TiCo-Bench in Table 1 extend to 60 s, and the realized duration in Figure 2 keeps tracking the requested targets up to 60 s. Appendix C.1 breaks the error down by instructed duration: TiCo's MAPE stays between 10% and 20% across the full 10–60 s range, including targets beyond 40 s, whereas the backbone's error rises at both ends. TiCo also transfers across input modalities. Although post-trained only on speech queries, it reaches 18.0% MAPE on text queries, close to its 16.2% on speech queries. See details in Appendix C.2.

6.3 On the efficiency of TiCo and the Spoken Time Marker

Spoken time markers are more data-efficient than direct instruction-following tuning. To isolate the contribution of TiCo and STMs, we compare two supervised *instruction tuning* (IT) variants that differ only in whether their target intermediate representations include STMs. Both variants are first trained on the same 4,000 examples as TiCo, paired with time-constrained instructions. As shown in Figure 3, direct IT does not achieve performance comparable to TiCo. Nevertheless, STMs remain effective within IT. A question naturally arises: *can IT match TiCo given more data?* We therefore scale the training set to 40k examples. However, direct IT with STMs still lags behind TiCo, even with $10\times$ more training data.

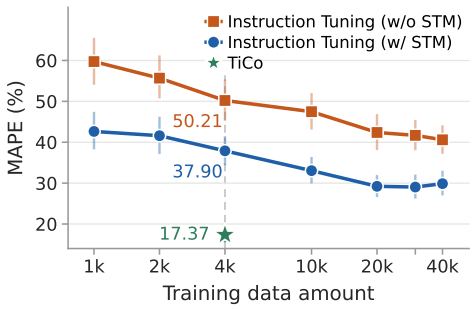


Figure 3: Data efficiency of TiCo. All models are evaluated on the same stratified subset of TiCo-Bench (500 samples).

Spoken time markers provide a denser duration supervision. Although measuring realized speech duration gives the most accurate duration feedback, STMs are more efficient for duration control for two reasons. First, final-duration rewards provide only sequence-level feedback, whereas STMs supply intermediate timing during training and autoregressive inference. We study the importance of intermediate STMs in Appendix G. Second, our profiling shows that the rollout completion time for full speech generation is approximately $20.9\times$ that for intermediate-sequence generation (Appendix A). Consequently, GRPO with realized-duration rewards incurs substantial rollout computation and memory costs while providing only sparse timing supervision. We tried our best but were unable to achieve effective duration control using this approach within our computational budget.

7 Qualitative Analysis and Examples

For acoustic examination, beyond UTMOS, we also study the prosody of TiCo's responses. Table 2 reports prosodic statistics of the generated responses. Most statistics shift by well under one standard deviation (SD) across responses: the silence ratio by 1.2 points (SD 4.0), the mean

Target 10 s → Realized 12.0 s	**Target 30 s → Realized 27.9 s**
Well, ⟨1.5⟩ Mars has resources like water ice, ⟨3.8⟩ which could be used for life support. ⟨5.4⟩ It also has an atmosphere that might be modified to make it more habitable. ⟨9.2⟩ So yeah, ⟨10.0⟩ it's a good candidate.	Well, ⟨1.5⟩ Mars has some things going for it. ⟨4.2⟩ It's closer to Earth than other potential places like Jupiter's moons. ⟨8.6⟩ There might be water ice on its surface, ⟨11.7⟩ which could be used for life support systems. ⟨14.9⟩ Also, ⟨16.3⟩ its gravity is lower compared to Earth's, ⟨19.0⟩ making it easier to build structures there. ⟨21.7⟩ And it has an atmosphere that can potentially be modified over time. ⟨25.5⟩ So all these factors make Mars a good candidate for human colonization. ⟨30.0⟩

Figure 4: TiCo's intermediate representation sequence for query *"Why is Mars considered a candidate for human colonization?"* under a 10 s and a 30 s target. STMs are abbreviated as $\langle t \rangle$.

pause by 0.06 s (SD 0.22 s), and the mean phone duration by 2.8 ms (SD 43.5 ms). The phone-level lengthening is spread across both vowels and consonants rather than concentrated on vowels. Speaking rate shows a bigger shift, dropping by 0.21 words/s (about 8%), which is still within one SD. This shows that TiCo learns to adjust the speaking rate slightly and mainly by adjusting the content. The speech generator itself is frozen, but it is conditioned on the hidden states of the sequence model, which TiCo does train. Delivery can therefore drift slightly even though it is never optimized directly. Direct speaking rate optimization remains our future work. Appendix I.1 provides more analysis on different TiCo backbones and detailed speaking rate distribution. Appendix I.2 further examines the information rate of the generated content.

Figure 4 shows the same query answered under a 10 s and a 30 s target. With the longer target, TiCo keeps the two points of the short answer (water ice and a modifiable atmosphere) and adds more detail through two new points (proximity to Earth and lower gravity), rather than stretching the same content or adding fillers. More examples are in Appendix J.

Table 2: Prosody analysis of generated responses.

Metric	Qwen2.5 Omni-7B	TiCo (Qwen2.5)
VAD silence ratio (%)	10.1 ± 4.0	11.3 ± 5.0
Speaking rate (words/s)	2.74 ± 0.30	2.53 ± 0.30
Pause mean (s)	0.46 ± 0.22	0.52 ± 0.27
Mean phone duration (ms)	80.6 ± 43.5	83.4 ± 46.5
Vowel duration (ms)	87.8 ± 51.2	89.4 ± 52.2
Consonant duration (ms)	76.1 ± 37.2	79.8 ± 42.2

8 Conclusion

Spoken dialogue models (SDMs) generate high-quality responses, yet they struggle to follow explicit duration constraints. We introduced **TiCo-Bench**, which evaluates this ability across four task categories and 10–60 s targets; every end-to-end SDM we evaluated exceeds 40% MAPE, regardless of scale. We then proposed **TiCo**, a simple yet effective post-training approach that requires only self-generated data. TiCo inserts Spoken Time Markers (STMs) into the intermediate representation, expressing elapsed speaking time as text. The markers thus take part in semantic planning, letting the model adjust its remaining content, and make the duration reward verifiable without synthesizing speech. Across three backbones with sequential and interleaved architectures, TiCo reduces duration error by 2.7–3.5× while preserving response quality, and it generalizes to unseen instruction types, targets beyond the trained duration range, and text queries.

Limitations. First, TiCo's reward verifies the final STM, not the audio. This is what makes training data- and compute-efficient, but it also means that control is only as accurate as the markers, which deviate from realized time by 2.7 s (Short) and 3.2 s (Long) on average (Appendix F.1). Grounding a fraction of Stage 2 rollouts in synthesized audio is a natural remedy that we leave to future work. Also, the markers are calibrated to the model's own synthesizer, so TiCo learns the timing of synthesized speech. Pauses, hesitations, and pacing that make dialogue sound natural are not modeled, and reconciling synthesizer-calibrated markers with human timing patterns is an open problem. Second, to end on time the model sometimes closes with a generic offer of further help or an unfinished clause (Appendix J). Our automatic metrics (e.g., GPT-score) measure overall quality but do not explicitly penalize this. Third, our evaluation covers single-turn English responses of 10 to 60 s, with the duration instruction given as text. Spoken instructions, multi-turn time budgets, real-time and full-duplex interaction are left to future work. Finally, TiCo controls duration mostly through semantic planning, by design: the speech generator is frozen. Combining TiCo with speaking-rate control is a direction we have not explored.

AI USE STATEMENT

We used LLMs to polish author-written text, to help locate related work, for brainstorming and as coding assistants for parts of the experimental code, and to paraphrase the instruction templates used in training and benchmarking. We also used a TTS system to synthesize the spoken queries for the text-only sources in our benchmark. LLMs are additionally employed within the study itself, as the automatic response-quality judge and as the planners in our cascaded baselines. The core method does not depend on any LLM beyond the spoken dialogue model itself. The authors reviewed all outputs and take full responsibility for the content of this paper.

REPRODUCIBILITY STATEMENT

TiCo-Bench is built from open-source datasets. The construction pipeline is discussed in Appendix B and their licenses are discussed in Appendix B.1; we will release the benchmark upon acceptance. Detailed training configurations for TiCo are given in Section 5.2 and Appendix A, including the hardware used for training and the associated computational requirements and constraints. Audio samples are available on an anonymized demo page, and we will release our models upon acceptance.

REFERENCES

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*.

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.

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.

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.

Scott J Adams, Julián N Acosta, and Pranav Rajpurkar. How generative ai voice agents will transform medicine. *npj Digital Medicine*, 8(1):353, 2025.

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*, 2025a.

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.

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.

Juncheng Xie and Hung-yi Lee. Prompt-based one-shot exact length-controlled generation with llms. *arXiv preprint arXiv:2508.13805*, 2025.

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.

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.

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.

Björn Lindblom. Explaining phonetic variation: A sketch of the h&h theory. In *Speech production and speech modelling*, pages 403–439. Springer, 1990.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024.

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.

James Glass. Challenges for spoken dialogue systems. In *Proceedings of the 1999 IEEE ASRU Workshop*, volume 696. MIT Laboratory for Computer Science Cambridge, 1999.

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

Adam Polyak, Yossi Adi, Jade Copet, Eugene Kharitonov, Kushal Lakhotia, Wei-Ning Hsu, Abdelrahman Mohamed, and Emmanuel Dupoux. Speech resynthesis from discrete disentangled self-supervised representations. In *Proc. Interspeech 2021*, pages 3615–3619, 2021.

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.

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.

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.

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.

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.

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*, 2025a.

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, 2025a. URL https://arxiv.org/abs/2503.20215.

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*, 2025b.

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.

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.

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.

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*, 2025a.

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*, 2025b.

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.

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.

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.

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.

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*, 2025a.

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.

Fabian Retkowski and Alex Waibel. Zero-shot strategies for length-controllable summarization. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 551–572, 2025.

Zeno Belligoli, Emmanouil Stergiadis, Eran Fainman, and Ilya Gusev. Controlling summarization length through eos token weighting. *arXiv preprint arXiv:2506.05017*, 2025.

Bradley Butcher, Michael O'Keefe, and James Titchener. Precise length control for large language models. *Natural Language Processing Journal*, 11:100143, 2025.

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.

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.

Gengxu Li, Tingyu Xia, Yi Chang, and Yuan Wu. Length-controlled margin-based preference optimization without reference model. *arXiv preprint arXiv:2502.14643*, 2025a.

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.

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.

Pranjal Aggarwal and Sean Welleck. L1: Controlling how long a reasoning model thinks with reinforcement learning. *arXiv preprint arXiv:2503.04697*, 2025.

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*, 2025a.

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/.

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. Free dolly: Introducing the world's first truly open instruction-tuned llm, 2023. URL https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. Don't give me the details, just the summary! Topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium, 2018.

OpenAI. GPT-5 system card. Technical report, OpenAI, 2025a. URL https://arxiv.org/abs/2601.03267.

Takaaki Saeki, Detai Xin, Wataru Nakata, Tomoki Koriyama, Shinnosuke Takamichi, and Hiroshi Saruwatari. UTMOS: UTokyo-SaruLab system for VoiceMOS challenge 2022. In *Proc. Interspeech*, pages 4521–4525, 2022.

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.

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.

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.

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*, 2025b.

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.

OpenAI. Update to gpt-5 system card: Gpt-5.2. Technical report, OpenAI, December 2025b. URL https://cdn.openai.com/pdf/3a4153c8-c748-4b71-8e31-aecbde944f8d/oai_5_2_system-card.pdf.

Qwen Team. Qwen2.5: A party of foundation models, September 2024. URL https://qwenlm.github.io/blog/qwen2.5/.

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.

Jérôme Louradour. whisper-timestamped. https://github.com/linto-ai/whisper-timestamped, 2023.

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.

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.

Zhihao Du, Changfeng Gao, Yuxuan Wang, Fan Yu, Tianyu Zhao, Hao Wang, Xiang Lv, Hui Wang, Chongjia Ni, Xian Shi, et al. Cosyvoice 3: Towards in-the-wild speech generation via scaling-up and post-training. *arXiv preprint arXiv:2505.17589*, 2025.

Heiga Zen, Viet Dang, Rob Clark, Yu Zhang, Ron J Weiss, Ye Jia, Zhifeng Chen, and Yonghui Wu. LibriTTS: A corpus derived from LibriSpeech for text-to-speech. April 2019.

Yunpei Li, Xun Zhou, Jinchao Wang, Lu Wang, Yong Wu, Siyi Zhou, Yiquan Zhou, Yining Wang, Yaogen Yang, Zhetao Hu, Shiyao Duan, Jiacheng Xu, Bin Xia, and Jingchen Shu. Indextts 2.5 technical report, 2026. URL https://arxiv.org/abs/2601.03888.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

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.

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.

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

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.

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.

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.

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.

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.

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

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, 2025b.

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.

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.

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.

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*, 2025b.

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, 2025b.

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.

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*, 2025b.

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.

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.

Appendix

Table of Contents

A TiCo Training Details **17**

B TiCo-Bench Construction Details **18**

B.1 Source Datasets . . . 19

B.2 TTS Pipeline and Quality Control . . . 19

B.3 Time-Control Instruction Sampling . . . 20

C Generalization Across Target Duration and Input Modality **20**

C.1 Generalization Beyond Trained Target Duration . . . 20

C.2 Generalization to Text Queries . . . 21

D Detailed Per-Subset Results on TiCo-Bench **22**

E Cascade System **25**

E.1 Cascaded System Prompt Templates . . . 25

E.2 Speaking-Rate-Controlled Cascade . . . 25

F Analysis on Spoken Time Marker **25**

F.1 Local Alignment of Spoken Time Marker . . . 26

F.2 Effect of Marker Insertion on Speech Generation . . . 26

G Spoken Time Marker Ablations **26**

G.1 Spoken Time Marker Ablation for Stage 1 TiCo . . . 26

G.2 Spoken Time Marker Ablation for Direct Instruction Tuning . . . 27

H TiCo Reward Ablations **27**

I Duration Control Behavior Analysis **27**

I.1 Prosodic Statistics . . . 28

I.2 Information Rate . . . 28

J Qualitative Examples **29**

J.1 How deep is the ocean? (Speech query) . . . 29

J.2 Discuss an event from history (Speech query) . . . 30

J.3 What is quantum mechanics? (Text query) . . . 31

J.4 Why is Mars considered a candidate for human colonization? (Text query) . . . 32

K Spoken Dialogue Model Survey **33**

A TiCo Training Details

We adopted MS-SWIFT (Zhao et al., 2024) [5] to train the model throughout this paper.

Stage 1: Time-Awareness SFT. We fine-tune with LoRA (r=8, α=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 5×10^{-5} and 10% warmup. Maximum sequence length is 2,048 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 (Zhang et al., 2026) to optimize duration controllability. The LoRA configuration uses r=8, α=32. We train for 800 steps with a per-GPU batch size of 1 and gradient accumulation of 8 (effective batch size 32), learning rate 5×10^{-6} with cosine schedule and 10% warmup. Each prompt generates G=4 rollouts with maximum completion length of 512 tokens. The clipping parameter is ε=0.2 and the KL penalty coefficient is β=0.04.

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.

Qwen-3-Omni 30B uses the same budget, and its talker emits codec tokens at the same rate, so the 2,048-token training cap again corresponds to roughly 41 seconds. At inference the thinker is allowed 2,048 tokens and the talker 4,096. We used H100 to train Qwen-3-Omni 30B but with limited access hours. Therefore the main analysis experiments in the paper is conducted with Qwen-2.5-Omni.

Step-Audio-2-mini is configured differently because it interleaves text and audio tokens in a single autoregressive stream, so one budget bounds both. We use 2,048 completion tokens for training and inference alike, without the inference-time increase applied to the Qwen models. Audio tokens are emitted at 25 Hz, so once the text portion of the stream is accounted for the effective speech ceiling is 65.5 seconds, or 1,638 audio tokens.

Sources of speech-rollout overhead. Figure 5 breaks a rollout into its three stages. Generating the intermediate sequence is nearly free, while the Talker and Code2Wav together account for almost all of the wall-clock. The Talker is not slower per token than the Thinker, as the two decode at 57.9 and 53.3 tokens/s respectively; the gap comes from sequence length, since one second of speech requires 50 codec tokens against 3.4 text tokens. The cost also grows with the requested duration, from 9.9$\times$ a Thinker-only rollout at 18 s of realized speech to 32.9$\times$ at 45 s, as the Talker scales linearly with duration while Code2Wav scales faster still. Longer speech outputs therefore increase the cost of evaluating duration rewards based on synthesized speech, because generating the complete audio becomes more expensive. TiCo avoids this overhead by computing duration rewards directly from Spoken Time Markers in the generated intermediate sequence, eliminating the need for speech synthesis during GRPO rollouts.

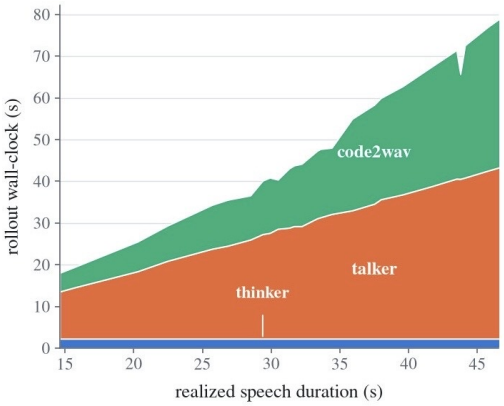


Figure 5: Per-stage rollout cost of Qwen2.5-Omni-7B, measured on one RTX 6000 Ada on TiCo-Bench prompts.

Reward Design. The main reward function is

$$\mathcal{R}_{\text{main}}^{(g)} = F\left(t_{\text{inst}} - t_{\text{last}}^{(g)}\right), \tag{6}$$

[5] https://github.com/modelscope/ms-swift

where t_{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 kernel:

$$F(\Delta t) = \exp\left(-\frac{(\Delta t)^2}{2\sigma^2}\right), \tag{7}$$

where σ 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,

$$\mathcal{R}_{\text{pres}}^{(g)} = \mathbb{I}[\, M_g \geq 1\,], \tag{8}$$

  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:

$$\mathcal{R}_{\text{mono}}^{(g)} = \frac{1}{M_g - 1} \sum_{j=1}^{M_g - 1} \mathbb{I}\Big[\, t_{j+1}^{(g)} > t_j^{(g)} \,\Big]. \tag{9}$$

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

$$\mathcal{R}_{\text{rep}}^{(g)} = -\left(1 - \frac{|\{t_1^{(g)}, \ldots, t_{M_g}^{(g)}\}|}{M_g}\right), \tag{10}$$

  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_{inst}:

$$\mathcal{R}_{\text{copy}}^{(g)} = -\frac{1}{M_g} \sum_{j=1}^{M_g - 1} \mathbb{I}\Big[\, |t_j^{(g)} - t_{\text{inst}}| < \tau \,\Big], \tag{11}$$

  where τ=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 grounding term in Section 4 is the sum of these four terms, $\mathcal{R}_{\text{ground}}^{(g)} = \mathcal{R}_{\text{pres}}^{(g)} + \mathcal{R}_{\text{mono}}^{(g)} + \mathcal{R}_{\text{rep}}^{(g)} + \mathcal{R}_{\text{copy}}^{(g)}$, so the overall reward for the g-th sample is

$$R^{(g)} = \mathcal{R}_{\text{main}}^{(g)} + \mathcal{R}_{\text{pres}}^{(g)} + \mathcal{R}_{\text{mono}}^{(g)} + \mathcal{R}_{\text{rep}}^{(g)} + \mathcal{R}_{\text{copy}}^{(g)}. \tag{12}$$

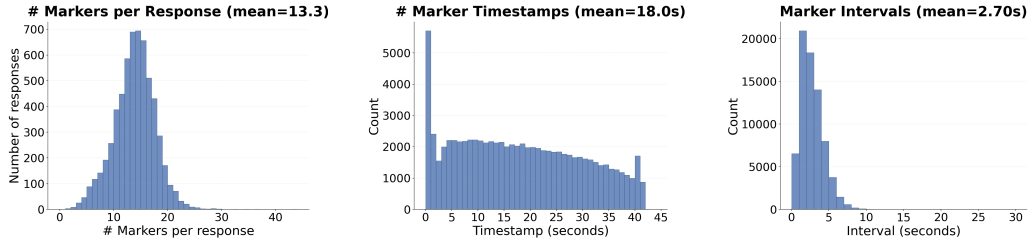
Note that $\mathcal{R}_{\text{rep}}^{(g)}$ and $\mathcal{R}_{\text{copy}}^{(g)}$ are non-positive by construction, so no explicit subtraction is needed.

CHORD. CHORD interleaves SFT updates with GRPO updates using a mixing coefficient μ that decays from μ_{peak}=0.8 to μ_{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.

Training Data Statistics. Figure 6 shows the distribution of Spoken Time Markers in the Stage 1 self-generated training data (Qwen2.5-Omni-7B). Each response contains on average 13.3 markers, with a mean inter-marker interval of 2.70 seconds. The marker timestamps are most densely distributed within the first 20–30 seconds, reflecting the typical length range of self-generated responses.

B TiCo-Bench Construction Details

This appendix expands on the construction of TiCo-Bench summarized in Section 3. We describe the source datasets and their licenses, the TTS pipeline used to obtain spoken queries for the text-only sources, and the time-control instruction sampling protocol.



(a) Markers per Response (b) Marker Timestamps (c) Inter-Marker Intervals

Figure 6: Distribution of Spoken Time Markers in the First stage training data.

B.1 Source Datasets

Figure 7 summarizes the resulting composition. TiCo-Bench draws speech queries from four publicly available sources, one per task category. We describe each below, including the rationale for assigning it to its category and the license under which the source dataset is distributed.

Question Answering (QA): InstructS2S (Fang et al., 2025a). InstructS2S is the spoken instruction-following corpus released alongside LLaMA-Omni, in which information-seeking questions are paired with single-voice speech recordings. Its prompts are predominantly factual question-answering, which directly matches the QA category. We sample 500 unique English queries that do not overlap with our training subset. The dataset is released under the CC BY-NC 4.0 license.

Reasoning (REA): URO-Bench (Yan et al., 2025). URO-Bench is a comprehensive evaluation suite for end-to-end spoken dialogue models, organized into capability-specific subsets. We restrict our sampling to its reasoning-oriented subsets, which together cover narrative reasoning, truthfulness, mathematical reasoning, multi-domain knowledge, and open-ended multi-turn reasoning. Drawing from these subsets ensures that the REA category in TiCo-Bench targets reasoning ability rather than generic question answering. We sample 300 queries in total. URO-Bench is released under the MIT license.

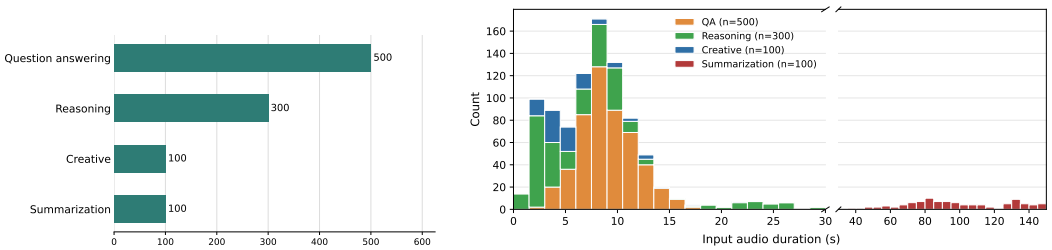
Creative Generation (CRE): databricks-dolly-15k Creative Writing (Conover et al., 2023). databricks-dolly-15k is an open-source instruction-tuning dataset organized into eight task categories. We use only its `creative_writing` subset, in which prompts ask the respondent to produce open-ended creative content such as a short rant, an imaginary monologue, or a stylized rewrite. The open-ended nature of these prompts makes the natural response length flexible, which is the property required of the CRE category. The dataset is released under the CC BY-SA 3.0 license.

Summarization (SUM): Extreme Summarization (XSum) (Narayan et al., 2018). XSum pairs each BBC news article with a single professionally written summary sentence. We use the article side as the input and prompt the model to produce a spoken summary under a target duration. Articles are substantially longer than the queries in the other three categories, which provides a natural source for the long-input regime that distinguishes the SUM category from the others. The dataset is released under the MIT license.

All four source datasets are released under licenses that permit redistribution and adaptation for non-commercial research, allowing us to include their queries as part of TiCo-Bench.

B.2 TTS Pipeline and Quality Control

For the Dolly-15k and XSum sources, which are text only, we obtain spoken queries through a TTS-then-verify pipeline. Each original instruction is synthesized with CosyVoice 3 (Du et al., 2025), using reference audio randomly sampled from LibriTTS (Zen et al., 2019) as the prompt voice and applying no paralinguistic conditioning. Synthesized waveforms are produced at 24 kHz mono.



(a) Per-task sample counts.

(b) Input-audio duration distribution per task.

Figure 7: Composition of **TiCo-Bench**. (a) The benchmark contains 1,000 base speech queries spanning four task categories: Question Answering, Reasoning, Creative, and Summarization. (b) Most queries fall within a short duration range, while only Summarization extends to substantially longer durations.

The two text sources are verified with different procedures, reflecting their distinct failure modes. For XSum, each synthesized utterance is transcribed using Whisper with the large-v3 checkpoint and compared against the source article using the `jiwer` library, after applying the Whisper English text normalizer to both sides. An utterance is retained only if its word error rate is below 0.1 and its duration falls within $[30, 150]$ seconds; the duration filter is what gives the SUM category its long-input regime. For Dolly, ASR-based verification is unreliable because creative-writing prompts contain rare proper nouns and unconventional phrasing that the ASR can mistranscribe even when the synthesized speech is intelligible. We therefore manually review every synthesized utterance and retain only those that are intelligible and faithful to the original prompt.

B.3 Time-Control Instruction Sampling

For each base query we sample a target duration t_{inst} uniformly from $[10, 30]$ seconds in the Short setting and from $(30, 60]$ seconds in the Long setting. Targets are quantized to one-second granularity, and the two regimes use disjoint target ranges. The two settings therefore give $1{,}000 \times 2 = 2{,}000$ evaluation samples in total.

The textual time-control instruction is instantiated from a pool of semantically equivalent templates, with one template sampled uniformly per query. The instruction is delivered to the model as a textual turn alongside the speech query, following the conversation format of each model under evaluation.

C Generalization Across Target Duration and Input Modality

TiCo is post-trained under a narrow set of conditions: the targets sampled in Stage 2 reach only 40 s, and every training query is speech. This appendix asks whether duration control survives outside those conditions. Appendix C.1 bins the speech-query results by instructed duration to see how control behaves as the target grows, in particular beyond the trained range. Appendix C.2 replaces the speech query with text and evaluates on two text benchmarks. In both settings we also check whether the Spoken Time Markers remain a faithful account of the realized duration.

C.1 Generalization Beyond Trained Target Duration

Section 6.1 reports errors aggregated over the Short and Long regimes. To examine how control holds up as the target grows, in particular beyond the 40 s range sampled in Stage 2, we bin the speech-query results by instructed duration. Figure 8 shows the MAE and MAPE of Qwen2.5-Omni-7B and TiCo on the QA and REA subsets of TiCo-Bench. The backbone fails at both ends: its MAPE is highest for the shortest targets, reaches a minimum around 25–30 s, and rises again for long targets, where its responses fall short. TiCo's MAPE stays between 10% and 20% across the whole range, indicating that duration control extends beyond the targets seen in training rather than being memorized within that range.

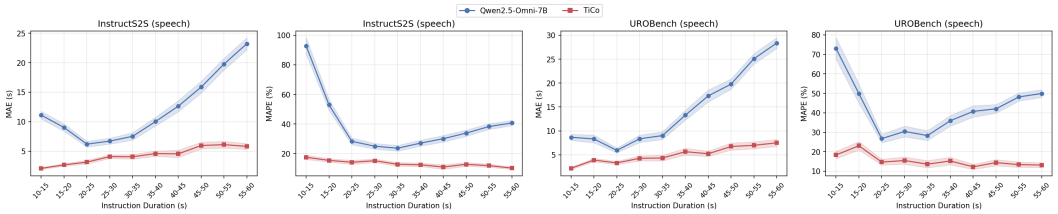


Figure 8: Duration MAE and MAPE of Qwen2.5-Omni-7B and TiCo across instructed-duration bins on the QA (InstructS2S) and REA (URO-Bench) subsets. The backbone's relative error is high at both ends of the range, while TiCo stays flat throughout.

Figure 9 compares, for TiCo, the error measured against the instructed duration t_{inst} with the error measured against its own final Spoken Time Marker t_{last}. The two curves nearly coincide, so TiCo's plan almost always ends at the instructed duration, and the remaining error comes from the gap between the planned and the realized duration.

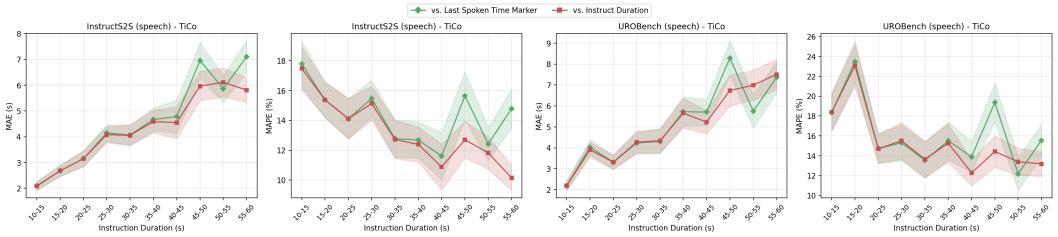


Figure 9: Duration error of TiCo across instructed-duration bins, comparing two reference signals: the instructed duration t_{inst} and the final Spoken Time Marker t_{last}. The close alignment indicates that the final time marker accurately estimates realized speech duration.

C.2 Generalization to Text Queries

During both training stages of TiCo, the model always receives speech queries as input. Here we evaluate whether the trained model generalizes to text queries on two benchmarks.

The first benchmark is **LIFEBench** (Zhang et al., 2025a), a length-instruction benchmark for text LLMs in which each query is paired with a word-count target. We extract its English queries and replace the original word-count instruction with a time-control instruction under the same Short (10–30 s) and Long (30–60 s) regimes as TiCo-Bench.

The second benchmark uses the textual transcriptions provided by **URO-Bench** (Yan et al., 2025). URO-Bench is a speech benchmark that pairs each speech query with its textual transcription. We use the transcriptions as text input, drawing from the same reasoning subsets as the REA category in TiCo-Bench, though with a smaller sample size. Targets follow the same Short and Long protocol as in TiCo-Bench.

Table 3 reports the time-control performance on text queries. TiCo achieves the lowest MAE and MAPE on all four benchmark-regime cells, with an overall MAPE of 18.0%, a 1.6× reduction over the strongest baseline Cascade (GPT) (28.1%) and a 2.7× reduction over the backbone Qwen2.5-Omni-7B (48.9%). Notably, this text-query performance is comparable to TiCo's speech-query MAPE on TiCo-Bench (16.2%, Table 1), despite the model being trained exclusively on speech queries. As shown in Figure 10, TiCo maintains consistently low error across all instructed-duration bins on both LIFEBench and UROBench-text, while the backbone's error grows substantially for longer targets. The GPT-score (2.76 vs. backbone 2.67) further confirms that response quality is preserved under the modality shift.

Figure 11 verifies that the marker-based planning mechanism remains faithful on text queries. The error of TiCo's outputs against the instructed duration tracks closely the error against the final Spoken

Time Marker on both benchmarks, with a small and roughly constant gap, indicating that the markers continue to track realized speaking time even when inputs are textual rather than spoken.

Table 3: **Text-query** evaluation on LIFEBench and UROBench-text under Short (10–30 s) and Long (30–60 s) settings. Results are reported as MAE (seconds) / MAPE (%), with the rightmost column showing GPT-score averaged across both benchmarks. Lower is better for MAE/MAPE, and higher is better for GPT-score. Model categories are indicated by color: Cascaded, Commercial, Open-sourced, and Proposed. **Bold** marks the best result per column.

	Short (10s–30s)		Long (30s–60s)		Overall	
Model	**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
+ TiCo	**4.46 / 22.7%**	**4.08 / 21.0%**	**6.62 / 14.8%**	**6.25 / 13.5%**	**5.35 / 18.0%**	2.76

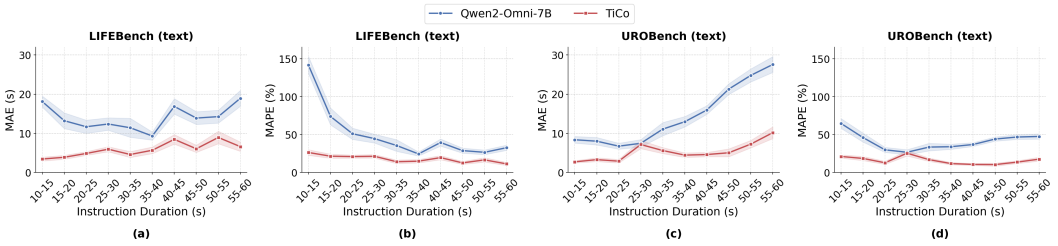


Figure 10: Text benchmarks: duration error of Qwen2.5-Omni-7B vs. TiCo measured against instructed duration. From left to right: LIFEBench MAE (s), LIFEBench MAPE (%), UROBench MAE (s), UROBench MAPE (%). Shaded regions indicate ±1 SEM.

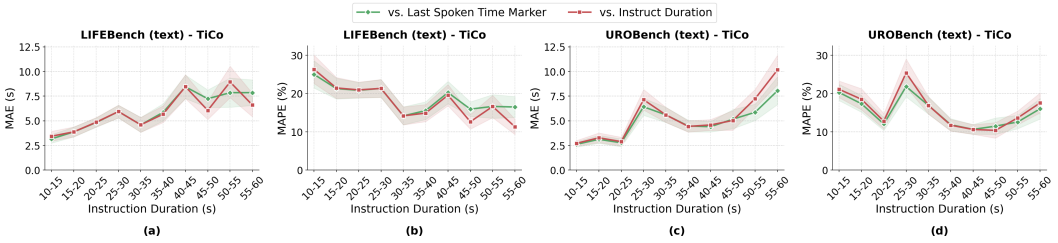


Figure 11: Text benchmarks (TiCo): duration error measured against instructed duration vs. last time marker. From left to right: LIFEBench MAE (s), LIFEBench MAPE (%), UROBench MAE (s), UROBench MAPE (%). Shaded regions indicate ±1 SEM.

D Detailed Per-Subset Results on TiCo-Bench

Table 4: Per-task **MAE** (seconds) on TiCo-Bench. Lower is better. Model categories indicated by color: Cascaded, Commercial, Open-sourced, and Proposed. **Bold** marks the best result in each column. *For Kimi Audio on the SUM subset, audio clips longer than 120 s exceed the model's effective context window and are excluded from evaluation.

TiCo-Bench (Speech)	Short (10s–30s)				Long (30s–60s)				Overall
Model	**QA**	**REA**	**CRE**	**SUM**	**QA**	**REA**	**CRE**	**SUM**	**MAE ↓**
Cascade (Qwen)	7.59	10.45	7.32	10.35	21.48	30.27	17.92	11.47	14.61
Cascade (GPT)	3.96	5.70	3.80	3.26	11.33	18.47	12.19	7.91	8.46
GPT-audio	16.74	20.72	23.71	11.99	19.90	20.63	21.60	17.80	19.36
Kimi Audio	24.03	29.26	28.01	10.49*	26.77	31.79	26.62	21.86*	26.50
LFM Audio	15.61	12.37	12.00	7.85	22.41	34.36	24.78	22.66	19.88
MiMo-Audio	13.77	12.50	21.60	11.51	20.42	17.53	17.96	15.29	16.37
Interleaved SDM									
Step-Audio-2-mini	13.60	11.61	16.27	25.29	21.93	30.29	18.96	16.38	19.01
+ TiCo	3.70	4.49	3.82	8.63	9.98	10.97	8.08	8.55	7.19
Sequential SDM									
Qwen2.5-Omni-7B	8.32	7.91	9.60	13.05	15.58	19.96	14.18	11.48	12.57
+ TiCo	3.00	3.37	3.42	8.10	5.30	6.38	5.86	5.94	4.70
Qwen3-Omni-30B	9.10	8.87	12.05	5.26	13.98	17.72	11.58	16.30	12.02
+ TiCo	**2.37**	**2.20**	**2.27**	**2.91**	**5.42**	**5.69**	**5.59**	**4.89**	**3.91**

Table 5: Per-task **GPT-score** (1–5 scale) on TiCo-Bench. Higher is better. Model categories indicated by color: Cascaded, Commercial, Open-sourced, and Proposed. **Bold** marks the best result in each column. *For Kimi Audio on the SUM subset, audio clips longer than 120 s exceed the model's effective context window and are excluded from evaluation.

TiCo-Bench (Speech)	Short (10s–30s)				Long (30s–60s)				Overall
Model	**QA**	**REA**	**CRE**	**SUM**	**QA**	**REA**	**CRE**	**SUM**	**GPT-score ↑**
Cascade (Qwen)	3.34	3.27	3.56	2.64	3.21	3.29	3.17	2.48	3.18
Cascade (GPT)	**4.07**	**4.11**	**4.13**	**2.88**	**4.11**	**4.19**	**3.99**	**2.86**	**3.92**
GPT-audio	3.74	3.74	3.38	2.78	3.56	3.33	3.06	2.50	3.45
Kimi Audio	1.93	1.56	2.03	1.10	1.72	1.46	1.92	1.10	1.69
LFM Audio	3.09	2.22	2.91	1.67	2.85	2.18	2.73	1.49	2.58
MiMo-Audio	2.48	3.19	3.20	2.07	3.24	3.22	3.15	1.96	2.91
Interleaved SDM									
Step-Audio-2-mini	2.93	2.61	2.93	2.18	2.85	2.54	2.86	2.15	2.72
+ TiCo	3.04	3.14	3.01	2.13	3.13	3.19	3.08	2.06	3.01
Sequential SDM									
Qwen2.5-Omni-7B	3.55	3.48	3.59	2.38	3.29	3.37	3.07	2.37	3.31
+ TiCo	3.31	3.50	3.40	2.34	3.39	3.67	3.31	2.36	3.32
Qwen3-Omni-30B	3.89	3.68	3.63	2.20	3.79	3.57	3.75	1.87	3.58
+ TiCo	3.82	3.83	3.77	2.59	3.98	3.93	3.87	2.71	3.76

Table 6: Per-task **UTMOS** (1–5 scale) on TiCo-Bench. Higher is better. The first row reports the ground-truth query speech as a reference and is excluded from the bold comparison. Model categories indicated by color: Cascaded, Commercial, Open-sourced, and Proposed. **Bold** marks the best result in each column. *For Kimi Audio on the SUM subset, audio clips longer than 120 s exceed the model's effective context window and are excluded from evaluation.

TiCo-Bench	Short (10s–30s)				Long (30s–60s)				Overall
Model	**QA**	**REA**	**CRE**	**SUM**	**QA**	**REA**	**CRE**	**SUM**	**UTMOS ↑**
GT (Speech Query)	4.28	3.79	4.16	3.90	4.28	3.79	4.16	3.90	4.08
Cascade (Qwen)	3.03	3.17	3.03	2.88	2.92	3.07	2.86	2.75	3.00
Cascade (GPT)	3.00	3.07	3.03	2.93	2.78	2.89	2.80	2.69	2.91
GPT-audio	3.89	3.82	3.73	3.86	3.65	3.54	3.55	3.50	3.72
Kimi Audio	2.47	2.56	2.30	2.59*	2.31	2.49	2.25	2.48*	2.43
LFM Audio	3.96	4.14	3.98	4.03	3.96	4.11	3.90	3.99	4.01
MiMo-Audio	3.06	3.47	3.49	3.55	3.26	3.47	3.25	3.41	3.31
Interleaved SDM									
Step-Audio-2-mini	4.51	4.51	**4.51**	**4.52**	**4.51**	**4.52**	**4.51**	**4.51**	**4.51**
+ TiCo	**4.52**	**4.52**	**4.51**	4.48	4.50	4.51	4.50	4.48	**4.51**
Sequential SDM									
Qwen2.5-Omni-7B	4.11	4.12	4.07	4.07	4.09	4.10	4.03	4.03	4.09
+ TiCo	4.10	4.11	4.07	4.12	3.98	3.99	3.97	4.03	4.04
Qwen3-Omni-30B	4.44	4.45	4.42	4.45	4.44	4.45	4.43	4.44	4.44
+ TiCo	4.44	4.45	4.44	4.45	4.43	4.44	4.42	4.43	4.44

E CASCADE SYSTEM

E.1 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.

E.2 SPEAKING-RATE-CONTROLLED CASCADE

The cascaded baselines in Table 1 control duration purely through text. Specifically, the language model decides how much content to produce, and the TTS stage synthesizes it at its default speaking rate, so whatever gap remains between the intended and realized duration goes uncorrected. Recent TTS systems like IndexTTS-2.5 (Li et al., 2026) expose explicit control of speaking rate, which offers a second opportunity to close that gap after the content has been produced. We therefore evaluate a stronger cascade built on IndexTTS-2.5.

We use a two-pass procedure. The response is first synthesized with the default duration factor of 1.0, yielding a speech response of duration t_0. IndexTTS-2.5 exposes a duration factor that scales synthesized duration relative to this default, so we set it to

$$\alpha = \mathrm{clip}\left(\frac{t_{\text{inst}}}{t_0},\, 0.5,\, 2.0\right), \tag{13}$$

where $\alpha > 1$ lengthens the speech and $\alpha < 1$ shortens it, and synthesize the same text a second time. The clipping range is the range IndexTTS-2.5 supports, so a response whose natural duration is more than twice or less than half the target cannot be fully corrected. We report both passes separately: *Natural*, the first-pass output, which measures how far the raw content sits from the target without any duration control, and *Adjusted*, the second-pass output, in which the duration has been rescaled toward the target.

Table 7: Speaking-rate-controlled cascade on TiCo-Bench. MAPE (%), GPT-score, and UTMOS are averaged over all eight subsets. Both IndexTTS-2.5 rows use Qwen2.5-7B-Instruct as the language model, matching TiCo's backbone family.

Model	MAPE ↓	GPT-Score ↑	UTMOS ↑
IndexTTS-2.5			
Natural	46.36	3.09	3.12
Adjusted	17.67	3.07	2.66
Qwen2.5-Omni-7B	43.33	3.31	**4.09**
+ TiCo	**16.19**	**3.32**	4.04

F ANALYSIS ON SPOKEN TIME MARKER

Spoken Time Markers are the mechanism through which TiCo exposes realized speaking time inside the generation loop, and the main results in Section 6.1 rest on them. This appendix examines the mechanism itself along two lines. Appendix F.1 asks whether the markers faithfully track the realized speaking time at every position, rather than only at the end of the response. Appendix F.2 asks whether inserting markers into the intermediate representation perturbs the downstream speech generation.

F.1 Local Alignment of Spoken Time Marker

This appendix provides the per-task breakdown of the local-accuracy analysis. To assess marker accuracy at every position rather than only at the end of the response, we align each Spoken Time Marker in the generated text with the word-level timestamp obtained from a Whisper-based ASR pass over the synthesized speech. For each matched position we compute the absolute error in seconds between the predicted marker time and the ASR-aligned word timestamp, then average across positions and samples within each task and duration regime. Table 8 reports the resulting local marker errors.

Table 8: Local alignment quality of Spoken Time Markers in TiCo. Each cell reports the mean absolute error in seconds between the predicted Spoken Time Marker and the corresponding ASR-aligned word timestamp.

Task	Short (10s–30s)	Long (30s–60s)
QA	1.90	3.09
Reasoning	2.08	3.32
Creative	2.05	3.60
Summarization	7.45	3.62
Overall	2.65	3.21

F.2 Effect of Marker Insertion on Speech Generation

At inference time, Spoken Time Markers are stripped from the intermediate representation by a regular expression before it is passed to the speech generator, so the markers never reach the Talker directly. However, they do alter the token sequence produced by the Thinker, and could therefore perturb speech generation indirectly. Hence, we quantify this as text–speech consistency. We transcribe the generated speech with ASR and compute the word error rate against the marker-stripped intermediate representation that was fed to the Talker. A low WER indicates that the speech faithfully realizes the intended text, so any drift introduced by the markers would surface as an elevated error rate.

Table 9: Effect of marker insertion on speech generation.

Model	WER (%)
Qwen2.5-Omni-7B	2.7 ± 4.8
+ TiCo	2.4 ± 4.5

G Spoken Time Marker Ablations

In this section, we ablate the design of STMs under three settings: (1) keeping only the last STM at the end of the response; (2) keeping all STMs but randomly shuffling their values; and (3) the default STM setting used in TiCo. We conduct the STM ablations in two scenarios: (i) Stage 1 training of TiCo, and (ii) direct instruction tuning (IT) with SFT.

G.1 Spoken Time Marker Ablation for Stage 1 TiCo

STM ablation for Stage 1 TiCo. Table 10 shows two findings. First, shuffled STMs keep the same number of markers but break the link between each marker value and the realized speaking time. The grounding reward in Stage 2 partly compensates for this (28.32%), but the remaining gap to TiCo (16.19%) shows that the gain comes from marker values grounded in measured speaking time, not only from the presence of marker tokens. Second, training Stage 1 with only the last STM gives the highest error (60.87%), worse than the backbone (43.33%). A single marker per response provides only a sparse signal, and the grounding reward, which scores intermediate markers, has nothing to act on in this setting. This makes the model challenging to learn to do semantic planning grounded on intermediate elapsed time.

Table 10: STM ablation for Stage-1 TiCo, on the full TiCo-Bench. #: average number of markers per training example in Stage 1. All values are MAPE (%) after Stage-2 training; lower is better. The shuffled and proposed rows carry an identical number of markers and identical response text, differing only in whether the marker values are in their measured order.

		Short (10–30 s)				Long (30–60 s)				
TiCo (Stage-1 STM format)	#	QA	REA	CRE	SUM	QA	REA	CRE	SUM	Overall
TiCo (last STM only)	1.0	102.83	68.99	113.95	94.45	27.06	32.85	24.40	29.72	60.87
TiCo (shuffled STM)	12.8	28.04	28.33	31.26	61.55	21.90	33.41	20.05	18.72	28.32
TiCo (proposed)	12.8	**15.40**	**17.77**	**17.33**	**49.03**	**11.74**	**13.92**	**13.11**	**13.45**	**16.19**

G.2 Spoken Time Marker Ablation for Direct Instruction Tuning

STM ablation for direction instruction tuning. Table 11 shows that in the direct IT scenario, the full STM setting achieves the lowest error. The last-only, shuffled, and no-STM variants perform similarly. We hypothesize that a single final STM gives too sparse a signal, and shuffled values give no usable one, for the model to learn duration control. Note that in Figure 3, increasing the data amount does improve the performance of direct instruction tuning, even without any STM. But it's less efficient than TiCo.

Table 11: Ablation of STMs for direction instruction tuning (IT). Values are MAPE (%) on TiCo-Bench; lower is better. Note that Stage-1-only TiCo do not see instructions before, so perfroms worse. Direct IT: direct instruction tuning with SFT.

	Short (10–30 s)				Long (30–60 s)				
Model	QA	REA	CRE	SUM	QA	REA	CRE	SUM	Overall
Direct IT (w/o STM)	94.12	68.74	102.68	108.71	23.44	32.28	22.68	21.67	57.33
Direct IT (last STM only)	95.03	68.16	101.36	95.88	25.75	33.81	25.18	27.50	57.99
Direct IT (shuffled STM)	89.60	71.91	89.05	124.61	24.90	33.32	23.07	28.64	57.68
Direct IT (default STM)	47.84	45.48	50.05	87.52	17.30	25.98	14.50	15.78	**35.40**
TiCo (proposed)	**15.40**	**17.77**	**17.33**	**49.03**	**11.74**	**13.92**	**13.11**	**13.45**	**16.19**

H TiCo Reward Ablations

Table 12 ablates the reward components. Models are evaluated on a 720-sample subset of TiCo-Bench. With only the main duration reward ($\mathcal{R}^{(g)}_{\text{main}}$), GRPO already outperforms the backbone model but exhibits reward hacking. Adding $\mathcal{R}^{(g)}_{\text{pres}}$ alone hurts performance, but combining it with $\mathcal{R}^{(g)}_{\text{mono}}$ recovers the baseline and improves MAE to 7.66s, showing that monotonicity is critical. Substituting presence with $\mathcal{R}^{(g)}_{\text{copy}}$ further reduces MAE to 5.30s, and adding $\mathcal{R}^{(g)}_{\text{rep}}$ brings it down to 4.71s. The *Full* configuration, integrating all components, achieves the best performance (MAE 4.55s, MAPE 15.38%), demonstrating that the reward components are complementary and jointly necessary. Overall, the main reward alone is sufficient to improve time-controllability over the base model, while the auxiliary rewards further prevent reward hacking and yield the best performance when combined.

I Duration Control Behavior Analysis

A model could in principle reduce duration error without genuinely adapting its semantic plan. One route is acoustic, for example extending the silence between phrases or lengthening individual phones. The other is lexical, such as padding the response with fillers and repeated content. Either route would satisfy the target duration while leaving the underlying content essentially unchanged. This appendix tests whether TiCo relies on such behaviors. We analyze the generated speech along two

Table 12: Ablation of reward components in second-stage training on a TiCo-Bench subset.

Variant	$\mathcal{R}^{(g)}_{\text{main}}$	$\mathcal{R}^{(g)}_{\text{pres}}$	$\mathcal{R}^{(g)}_{\text{mono}}$	$\mathcal{R}^{(g)}_{\text{copy}}$	$\mathcal{R}^{(g)}_{\text{rep}}$	**MAE (s)**	**MAPE (%)**
Main only	✓					8.39	32.21
+ Presence	✓	✓				10.18	38.89
+ Presence + Mono.	✓	✓	✓			7.66	28.10
+ Mono. + Copy	✓		✓	✓		5.30	19.07
+ Mono. + Copy + Rep.	✓		✓	✓	✓	4.71	17.16
Full	✓	✓	✓	✓	✓	**4.55**	**15.38**

axes: prosodic and acoustic realization and information density. Throughout, we compare TiCo against its backbones, computed over the full TiCo-Bench.

I.1 Prosodic Statistics

Table 13 reports seven prosodic statistics of the generated speech, grouped into utterance timing and segment-level duration. The utterance-level statistics are derived from Silero VAD[6], except for speaking rate, which is computed from Whisper transcription. VAD silence ratio is the fraction of the response duration falling outside detected speech regions, and captures how much of the time budget is spent not speaking. Pause mean is the average duration of the detected non-speech intervals. Speaking rate is the number of transcribed words divided by the response duration, and reflects how fast the model delivers.

The segment-level statistics are obtained from forced alignment with the Montreal Forced Aligner (MFA)[7]. Mean phone duration is the average realized duration of an aligned phone, with vowel duration and consonant duration giving the same quantity restricted to each class. Vowel-consonant ratio is the ratio between the two, and indicates whether strengthening is distributed evenly across phone classes or concentrated on vowels.

Table 13: Acoustic statistics of generated responses on TiCo-Bench, before and after TiCo training.

	Step-Audio-2-mini		Qwen2.5-Omni-7B		Qwen3-Omni-30B	
Metric	Base	**+ TiCo**	Base	**+ TiCo**	Base	**+ TiCo**
VAD silence ratio (%)	12.8±10.9	14.1±17.1	10.1±4.0	11.3±5.0	7.1±3.8	6.4±3.3
Speaking rate (words/s)	2.24±0.46	2.23±0.50	2.74±0.30	2.53±0.30	2.48±0.37	2.89±0.33
Pause mean (ms)	414±661	395±237	460±225	517±265	323±131	328±128
Mean phone duration (ms)	84.4±43.9	86.3±43.9	80.6±43.5	83.4±46.5	76.2±39.4	74.7±39.2
Vowel duration (ms)	89.7±52.3	93.4±52.8	87.8±51.2	89.4±52.2	85.0±46.9	84.0±48.1
Consonant duration (ms)	81.1±37.3	82.0±36.7	76.1±37.2	79.8±42.2	70.9±32.8	69.0±31.3
Vowel:consonant ratio	1.11	1.14	1.15	1.12	1.20	1.22

I.2 Information Rate

The statistics in Table 2 present how the response is realized acoustically, but not what it contains. To detect lexical padding, we quantify how much information the generated content carries, following the surprisal-based formula[8]. We report two quantities. Mean surprisal measures information per token in bits and reflects how predictable the generated content is. Information rate measures information per second and reflects how much the response delivers per unit of the time budget. Fillers and repeated content are highly predictable given their context, so such padding would depress both quantities. To compute these quantities, we transcribe with CrisperWhisper[9], a Whisper variant trained for verbatim transcription that preserves disfluencies such as filled pauses and repetitions. Subsequently,

[6] https://github.com/snakers4/silero-vad

[7] https://github.com/MontrealCorpusTools/Montreal-Forced-Aligner

[8] https://github.com/aalok-sathe/surprisal/

[9] https://github.com/nyrahealth/CrisperWhisper

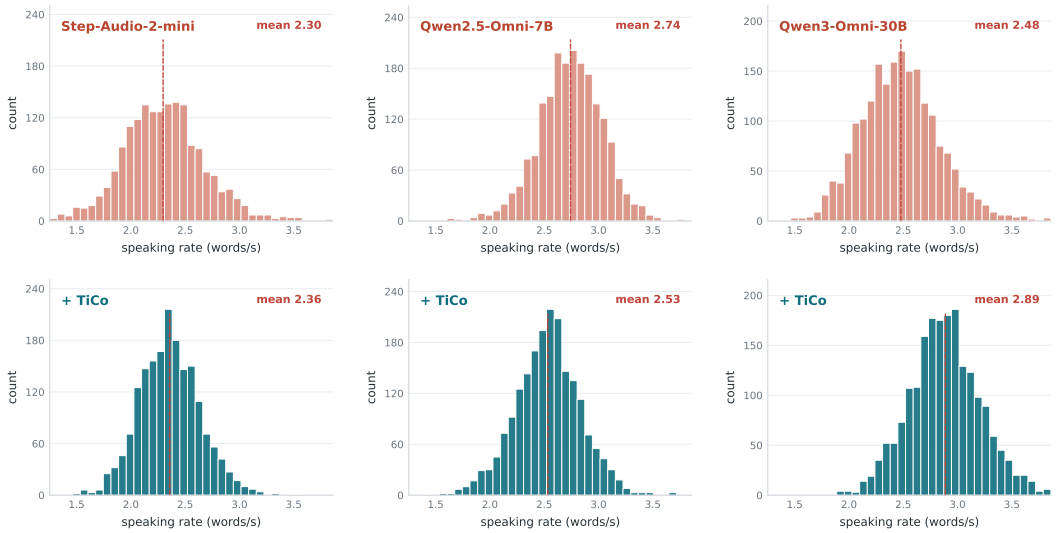


Figure 12: Distribution of speaking rate for each backbone (top) and after TiCo training (bottom). TiCo shifts the mean while the shape of the distribution stays largely unchanged.

we compute scores of the resulting transcription with GPT-2 (Radford et al., 2019), which serves as a fixed reference distribution. Results are shown in Table 14.

Table 14: Surprisal and information rate of generated responses.

Metric	Cascade (GPT)	Qwen2.5-Omni-7B	TiCo (Qwen2.5)
Mean surprisal (bits/token)	4.28	3.27	3.38
Information rate (bits/sec)	13.77	10.65	10.15

J Qualitative Examples

J.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*

J.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: *Discuss an event from history. 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*

J.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*

J.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*

K SPOKEN DIALOGUE MODEL SURVEY

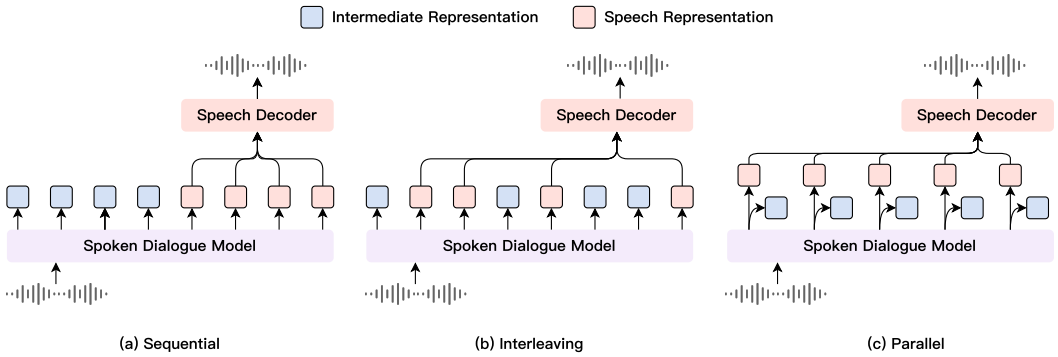


Figure 13: Illustration of different generation patterns in spoken dialogue models (SDMs): (a) Sequential, (b) Interleaved, and (c) Parallel.

Table 15 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 (Arora et al.) 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 15, 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 13.

- **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* (Borsos et al., 2023).

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 (Mousavi et al., 2025; Guo et al., 2025; Wu et al., 2024) 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 (Zhang et al.; Défossez et al., 2024).

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

Table 15: 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 Rrep.**: 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 Nguyen et al. (2023)	2022-03	-	Phonetic token	Direct	"Dual-tower" architecture for dual channel full-duplex modeling. Direct modeling of two-channel phonetic tokens.
SpeechGPT Zhang et al. (2023)	2023-05	Text (+R.)	Phonetic token	Sequential	Using "Chain-of-Modality". Expands LLM vocabulary with phonetic tokens.
SPIRIT-LM Nguyen et al. (2025)	2024-02	Text (+S.)	Phonetic token	Interleaved	Interleaves text and speech in one stream; expressive version adds pitch and style tokens.
Moshi Défossez et al. (2024)	2024-07	Text	Acoustic token	Parallel	Dual-channel full-duplex model. Parallel decoding text and acoustic tokens with delay pattern.
LLaMA-Omni Fang et al. (2025a)	2024-09	Text	Phonetic token	Parallel	CTC speech decoder maps LLM response states to phonetic tokens for streaming speech synthesis.
SyncLLM Veluri et al. (2024)	2024-09	-	Phonetic token	Direct	Interleaving user speech and model speech for full-duplex dialogue.
Mini-Omni2 Xie and Wu (2024)	2024-10	Text	Acoustic token	Parallel	Parallel decoding with delay pattern.
Freeze-Omni Wang et al. (2025b)	2024-11	Text	Acoustic token	Sequential	TDM-based full-duplex interaction; speech decoder conditioned on text tokens and LLM hidden states.
GLM-4-Voice Zeng et al. (2024)	2024-12	Text	Phonetic token	Interleaved	Single speech codebook paired with flow matching speech decoder.
SLAM-Omni Chen et al. (2025)	2024-12	Text	Phonetic token	Parallel	"Semantic group modeling" enables generating multiple phonetic tokens per text token.
VITA-1.5 Fu et al. (2025)	2025-01	Text	Acoustic token	Sequential	NAR + AR speech decoder taking LLM embedding as input.
Baichuan-Audio Li et al. (2025b)	2025-02	Text	Acoustic token	Interleaved	Text-guided speech generation with an independent audio head.
Qwen2.5-Omni Xu et al. (2025a)	2025-03	Text	Acoustic token	Sequential	Thinker-Talker architecture. Supports visual modality.
Kimi-Audio Ding et al. (2025)	2025-04	Text	Phonetic token	Parallel	Shared LLM with text head and audio head.
LLaMA-Omni 2 Fang et al. (2025b)	2025-05	Text	Phonetic token	Parallel	Gate fusion of LLM hidden states and text tokens for improved speech quality.
Step-Audio 2 Wu et al. (2025)	2025-07	Text (+R., Tool)	Phonetic token	Interleaved	Using "multi-modal RAG" to support grounded response and timbre/style control.
STITCH Chiang et al. (2026)	2025-07	Text (+R.)	Phonetic token	Interleaved	Backbone: GLM-4-Voice. Various interleaving text, reasoning, speech patterns are discussed in the paper.
Qwen3-Omni Xu et al. (2025b)	2025-09	Text (+R.)	Acoustic token	Sequential	Thinker-Talker architecture. Supports explicit thinking mode and compatible with tool calling. Supports visual modality.
Moshi-CoT Shih et al. (2025)	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 Arora et al. (2025b)	2025-10	Text	Acoustic token	Interleaved	CoT framework for SDMs. Blockwise streaming full-duplex model.
Streaming RAG Arora et al. (2025a)	2025-10	Text (+Tool)	Acoustic token	Sequential	Enables the SDM to trigger tool queries in parallel with the user's speech.
LFM2-Audio Amini et al. (2025)	2025-11	Text	Acoustic token	Interleaved	Supports both interleaved and sequential patterns, adapting to different tasks.
Mimo-Audio Zhang et al. (2025b)	2025-12	Text (+R.)	Acoustic token	Interleaved	Interleaving text tokens and "audio patches", which includes a delay pattern.
PersonaPlex Roy et al. (2026)	2026-01	Text	Acoustic token	Parallel	Followed the Moshi architecture. Dual-channel full-duplex model.
MiniCPM-o 4.5 OpenBMB (2026)	2026-02	Text (+R.)	Acoustic token	Interleaved	TDM full-duplex model. Supports visual modality.