# What Makes a Good Full-Duplex SLM? A Systematic Study of Architecture, Objectives, and Data

**Anonymous authors**
Paper under double-blind review

## Abstract

Full-Duplex Speech Language Models (FD-SLMs) model user and assistant speech as separate channels, allowing a spoken assistant to listen while speaking and handle overlapping speech. However, existing systems combine different architectures, backbones, objectives, and data recipes, which makes their design choices difficult to compare. We conduct controlled experiments on four axes—dual-channel sequence layout and speech granularity, duplex training objectives, text guidance granularity, and real versus synthetic duplex data—using a shared pipeline and two open SLM backbones. Interleaved training improves mean dialogue quality and interruption recovery over our step-aligned implementation across both backbones and all tested speech chunk sizes. Joint user–assistant prediction with equal loss weights increases assistant-speech cross-entropy and usually reduces post-interruption latency, but its effects on dialogue quality and post-interruption recovery rate depend on the backbone and chunk size. Turn-level text guidance favors mean dialogue quality and faster post-interruption responses, whereas chunk-level guidance makes models easier to interrupt. Synthetic dialogue improves spoken QA on GLM-4-Voice and post-interruption latency and response rating on both backbones, but reduces post-interruption recovery rate; we also observe text-channel collapse in the synthetic-trained Kimi-Audio checkpoint. These findings provide actionable insights into selecting architectures, training objectives, and dialogue data to balance response quality, generation stability, and turn-taking behavior.

## 1 Introduction

Turn-based spoken assistants usually wait for the user to finish speaking before they respond (Cui et al., 2025; 2026a). Real conversations are different: people give short feedback while listening, start responding before an utterance ends, interrupt, and yield the floor. Overlapping speech is therefore an important part of natural conversation. To model such overlap, dialogue audio must be represented as two synchronized channels, one for the user and one for the assistant, so that both speakers can be active at the same time. Full-Duplex Speech Language Models (Full Duplex SLMs, FD-SLMs) are designed for this two-channel setting. They process the user channel while generating the assistant channel, allowing them to listen while speaking and support more natural real-time interaction. Recent systems have shown the potential of this approach (Nguyen et al., 2022; Zhang et al., 2026b; Ma et al., 2024; Xu et al., 2024; Zhang et al., 2025).

However, it remains unclear how different design choices affect full-duplex learning. Existing systems often differ in their base model architecture, pretraining data, audio representation, duplex training objective, duplex training data, etc. When many components change at the same time, it is difficult to tell which one causes a performance difference or to draw general lessons across systems. In addition, full-duplex modeling is harder than turn-based modeling because the model must process two concurrent audio streams while generating its response. Poor architecture or data choices can reduce the model's general capability or make the interaction less natural. Controlled comparisons are therefore needed to understand how to build FD-SLMs that remain capable and interact naturally.

We conduct controlled experiments on four common choices in FD-SLM training, using the same training and evaluation pipeline within each comparison. We run the central comparisons on two different open SLM backbones, GLM-4-Voice (Zeng et al., 2024) and Kimi-Audio (KimiTeam et al.,



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.

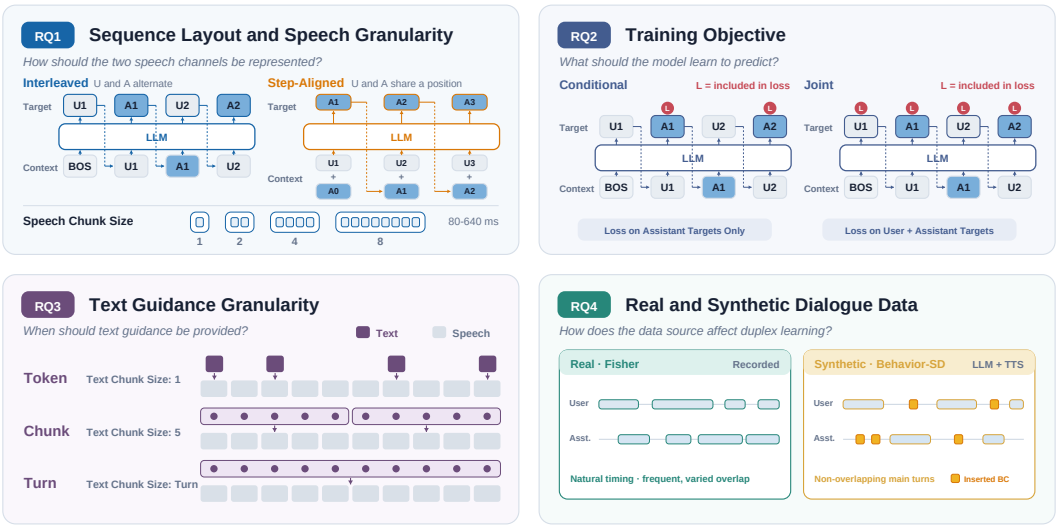



Figure 1: Overview of the four controlled comparisons. RQ1 studies dual-channel sequence layout and speech granularity; RQ2 compares conditional and joint training objectives; RQ3 varies text-guidance granularity; and RQ4 contrasts real and synthetic dialogue data.

2025), to test whether the conclusions hold across model families. We study the following research questions (summarized in Figure 1):

- **RQ1: Dual-Channel Sequence Layout and Granularity.** Should user inputs and assistant outputs use an interleaved or step-aligned layout, and what speech chunk size $s_c$ should each autoregressive step represent?
- **RQ2: Duplex Training Objective.** Should the model predict only assistant audio conditioned on the user, or jointly model both speakers?
- **RQ3: Text Guidance Granularity.** At what granularity should text tokens be scheduled to guide speech generation?
- **RQ4: Duplex Data Source.** How does training on real versus synthetic duplex speech affect model capabilities and interaction behavior?

Our experiments provide several practical findings. First, interleaved training improves mean dialogue quality and interruption recovery over our step-aligned implementation across both backbones and all tested speech chunk sizes. Second, joint user–assistant prediction increases assistant-speech CE, while its effects on dialogue quality and turn-taking depend on the backbone and speech chunk size. Third, turn-level text guidance achieves higher mean dialogue quality and faster post-interruption responses, whereas chunk-level guidance makes the model easier to interrupt. Finally, synthetic dialogue improves spoken QA on GLM-4-Voice and post-interruption latency and response rating on both backbones, but reduces post-interruption recovery rate; the synthetic-trained Kimi-Audio checkpoint also exhibits text-channel collapse. Together, these findings offer actionable insights into choosing architectures, training objectives, and dialogue data to balance response quality, generation stability, and turn-taking behavior in FD-SLMs.

## 2 Related Work

**Full-Duplex Spoken Dialogue Modelling.** FD-SLMs jointly process user and assistant speech to support overlapping interaction. Existing architectures use separate Transformer towers with cross-attention, as in dGSLM (Nguyen et al., 2022), interleave user and assistant time slices (Zhang et al., 2026b), decode both channels in parallel (Xu et al., 2024), or serialize speech and text into one sequence (Zhang et al., 2025). NTPP (Wang et al., 2025) uses step-aligned prediction of paired speech tokens. Systems further differ in user-speech fusion (Ma et al., 2024; Lu et al., 2026a), text

guidance (Cui et al., 2026b), and interaction control (Roy et al., 2026; Chang et al., 2026). These choices are coupled with different backbones and training recipes, complicating direct comparisons.

**Data and Evaluation.** Fisher (Cieri et al., 2004) provides real conversations with natural timing and overlap, whereas synthetic corpora offer greater control over dialogue content and interaction behaviors (Lee et al., 2025; Roy et al., 2026). Evaluation covers both conversational timing, as in Full-Duplex-Bench (Lin et al., 2025), and multi-round dialogue quality, as in MTR-DuplexBench (Zhang et al., 2026a). LLM-based assessments also depend on the judge and evaluation protocol (Zheng et al., 2023).

**Investigating FD-SLM Design Choices.** Recent studies examine individual design choices: TurnGuide (Cui et al., 2026b) studies chunk-level text guidance, NTPP (Wang et al., 2025) jointly predicts paired user and assistant tokens, and State Inertia (Chang et al., 2026) investigates shifts in predictive focus between speakers. The closest controlled study compares user-channel fusion and cross-attention routing under a shared training pipeline (Lu et al., 2026a). These studies focus on individual design axes or model settings, while surveys summarize the broader design space (Lu et al., 2026b). We instead compare sequence layout and granularity, training objectives, text guidance, and data source through controlled experiments across two backbones.

## 3 Experimental Design

**Model Backbone.** We use two pretrained SLMs as backbones: GLM-4-Voice and Kimi-Audio. GLM-4-Voice uses one output vocabulary for both text and discrete speech tokens. Text and speech token predictions occur in separate autoregressive steps. Kimi-Audio takes both continuous speech features and discrete speech tokens as input, and uses separate output layers for text and speech. Text and speech tokens are predicted simultaneously. Because the two backbones predict over different output spaces, cross-entropy (CE) values are meaningful only when comparing configurations within the same backbone. Lower CE means that the model assigns a higher probability to the correct target tokens. Both backbones represent speech at 12.5 Hz.

**Training Data.** We primarily train on the Fisher English corpus (Cieri et al., 2004), which contains real two-channel telephone conversations and preserves naturally occurring turn timing and overlapping speech. We pre-process Fisher audio into 45,532 non-overlapping 120-second segments for training and 404 separate segments for validation. Each training segment is presented once with the original user and assistant channels and once with the channels swapped, preventing either physical channel from always having the same role. To study the effect of real and synthetic training speech in RQ4, we additionally use Behavior-SD (Lee et al., 2025), a synthetic dialogue corpus created with language-model-generated conversations and text-to-speech synthesis.

**Training Recipe.** For each configuration, we train for one epoch on its target training data. We use a linear warmup and decay learning-rate schedule with a peak learning rate of $4 \times 10^{-6}$. During the first 3% of training updates, the learning rate increases linearly to this peak and then gradually decreases. Training uses eight H20 GPUs and ZeRO Stage 2 (Rajbhandari et al., 2020) to distribute optimizer memory across the GPUs. Table 1 summarizes the four comparisons. Appendix F provides the training, decoding, and automated-judge settings and full judge prompts.

**Evaluation Benchmarks.** We utilize the Dialogue Quality subset of the MTR-DuplexBench (MTR) (Zhang et al., 2026a) to measure the semantic interaction capability. It runs each model with full-duplex user input and uses DeepSeek-V4 Pro to rate the dialogue quality from 0 to 5. OpenAudioBench (OAB) (KimiTeam et al., 2025) evaluates spoken question answering; DeepSeek-V4 Pro (Xu et al., 2026) judges whether each answer is correct, and we report accuracy. Full-Duplex-Bench (FDB) (Lin et al., 2025) evaluates conversational timing and turn-taking behaviors, including pause handling, backchanneling, and interruptions.

### 3.1 RQ1: Sequence Layout and Speech Granularity

Existing FD-SLMs mainly organize the two synchronized audio channels in two ways. SyncLLM (Veluri et al., 2024) and TurnGuide (Cui et al., 2026b) use an interleaved layout, in which aligned

Table 1: Overview of the four controlled comparisons. Each row lists the design choice being varied, the settings compared, the training data, and the base configuration used for the comparison.

| | Design Choice | Compared Settings | Training Data | Base Setting |
|---|---|---|---|---|
| RQ1 | Sequence layout and speech granularity | Interleaved vs. step-aligned; $s_c \in \{1, 2, 4, 8\}$ | Fisher | Audio-only; assistant-only objective |
| RQ2 | Duplex training objective | Assistant-only vs. joint user–assistant prediction; $s_c \in \{1, 4, 8\}$ | Fisher | Interleaved; audio-only |
| RQ3 | Text guidance granularity | $t_c \in \{1, 5, \text{turn}\}$ | Fisher | $s_c = 1$; one-step text lead |
| RQ4 | Duplex data source | Real Fisher vs. synthetic Behavior-SD | Fisher or Behavior-SD | Assistant-only; interleaved; turn-level text; $s_c = 1$ |

user and assistant speech chunks are serialized as separate blocks of autoregressive positions. Moshi (Défossez et al., 2024), LSLM (Ma et al., 2024), and F-Actor (Züfle et al., 2026) use a step-aligned layout: at each autoregressive step, the model receives the user-speech representation for a time interval and predicts the assistant-speech token or tokens aligned with that same interval. These two layouts make different trade-offs between sequence length and the autoregressive context available within each stream. **RQ1 compares interleaved and step-aligned layouts across several speech chunk sizes.**

To describe the two layouts, let $U = (u_1, \ldots, u_T)$ and $A = (a_1, \ldots, a_T)$ be the user and assistant speech-token streams, where $u_t$ and $a_t$ cover the same time interval. The speech chunk size $s_c$ is the number of consecutive tokens from each channel in one temporal chunk. We use $s_c = 1, 2, 4, 8$, corresponding to 80, 160, 320, and 640 ms per chunk. Let $B = \lfloor T/s_c \rfloor$ be the number of complete chunks, and define $U_b = (u_{b,1}, \ldots, u_{b,s_c})$ and $A_b = (a_{b,1}, \ldots, a_{b,s_c})$, where $u_{b,r} = u_{(b-1)s_c+r}$ and $a_{b,r} = a_{(b-1)s_c+r}$.

In the interleaved layout, aligned user and assistant chunks alternate in the autoregressive sequence:

$$S_{\text{interleave}} = (U_1, A_1, U_2, A_2, \ldots, U_B, A_B), \tag{1}$$

Each chunk is expanded into its $s_c$ speech tokens, and every token occupies its own Transformer position. The interleaved sequence has approximately $2T$ Transformer positions, and changing $s_c$ changes speech modeling granularity for each channel.

In the step-aligned layout, when $s_c = 1$, each Transformer position receives the current user speech feature or token together with the previous assistant token and predicts the assistant token aligned with the current user input. For $s_c > 1$, one position represents a complete chunk: the embeddings of the $s_c$ current user tokens and the $s_c$ previous assistant tokens are summed, and $s_c$ parallel output heads predict the aligned assistant chunk. Formally, at chunk $b$,

$$x_b = \sum_{r=1}^{s_c} \left[ E_U(u_{b,r}) + E_A(a_{b-1,r}) \right], \qquad (\hat{a}_{b,1}, \ldots, \hat{a}_{b,s_c}) = (H_1(h_b), \ldots, H_{s_c}(h_b)). \tag{2}$$

Here, $x_b$ is the input embedding at step $b$, and $h_b$ is the corresponding Transformer hidden state. $E_U$ and $E_A$ denote the user and assistant input representations, respectively; $H_r$ is the $r$-th output head, and $\hat{a}_{b,r}$ is its predicted assistant token. We choose this formulation because summing multiple aligned embeddings and predicting multiple speech tokens through parallel heads are well-established modelling choices in SLAM-Omni (Chen et al., 2024), SALM-Duplex (Hu et al., 2025), and F-Actor (Züfle et al., 2026), although these systems group tokens along different dimensions.

**Remark 1** *We adopt a simple, general step-aligned formulation that can be applied to both backbones. For $s_c > 1$, we sum input representations without adding explicit within-chunk positional encodings or using a learned fusion module, such as a small attention network. On the output side, parallel heads predict all tokens in a chunk from the same hidden state, without sequential conditioning on earlier predictions within that chunk. Our comparison therefore evaluates this specific formulation rather than the full range of step-aligned architectures; richer input fusion and sequential multi-token prediction remain unexplored.*

### 3.2 RQ2: Conditional Versus Joint Modelling

An FD-SLM can be trained only to generate the assistant stream, or it can also learn to predict the user stream. Predicting both streams may help the model learn the behavior of both speakers, but the additional objective may compete with assistant generation. **RQ2 compares assistant-only and joint user–assistant training at $s_c = 1, 4, 8$.**

In conditional training, the user speech is provided as context and the model is trained only to predict the assistant speech. Its loss is $\mathcal{L}_{\text{cond}} = \mathcal{L}_A = -\log p(A \mid U)$. In joint training, the model is also trained to predict the user speech:

$$\mathcal{L}_{\text{joint}} = \mathcal{L}_A + \lambda \mathcal{L}_U, \qquad \lambda = 1, \tag{3}$$

where $\mathcal{L}_U$ is the user-speech loss and $\lambda = 1$ gives the user and assistant losses equal weight.

### 3.3 RQ3: Text Guidance Granularity and Layout

Assistant text can provide a semantic plan that guides speech generation when it is placed near the speech that expresses the same content. However, the appropriate temporal granularity for adding text to an FD-SLM remains unclear. For example, Moshi (Défossez et al., 2024) aligns individual text tokens with speech frames, TurnGuide (Cui et al., 2026b) aligns five-token text chunks with corresponding speech chunks, while OmniFlatten (Zhang et al., 2025) aligns the complete text of a turn to the beginning of its speech and then interleaves the text and speech streams. Motivated by these different choices, **RQ3 compares token-, chunk-, and turn-level text scheduling within the same full-duplex training pipeline.** At every speech decoding step, the model predicts one speech token. If the schedule assigns a text token to that step, the model also predicts one text token; otherwise, it predicts speech only. We use $t_c$ for the number of text tokens scheduled as one group: token-level guidance uses $t_c = 1$, chunk-level guidance uses $t_c = 5$, and turn-level guidance groups all text tokens in an assistant turn, denoted by $t_c =$ turn. We compare the three granularities at $s_c = 1$. This is because when using larger speech chunk sizes, the token-level schedule runs out of available positions to insert text tokens. Figure 5 illustrates the three text schedules. Appendix D describes the exact alignment and backbone-specific implementations.

### 3.4 RQ4: Real and Synthetic Speech

Real and synthetic duplex speech provide different training signals. Real conversations preserve natural turn timing and overlapping speech, while synthetic conversations provide greater control over dialogue content but may not fully reproduce these interaction patterns. Appendix E shows that Fisher and Behavior-SD have similar average speech and text density but differ substantially in overlap frequency and interaction-timing diversity. **RQ4 compares training on real Fisher speech with training on synthetic Behavior-SD speech.** For each backbone, we train one model on each corpus using the assistant-only objective, the interleaved layout, turn-level text guidance, and $s_c = 1$. Behavior-SD contains slightly more audio than Fisher, so we subsample it to match Fisher's total duration of 1,517.7 hours, giving the two data-source conditions the same amount of training audio.

## 4 Results

This section presents the results for all four RQs. We note that we only show the metrics most relevant to each finding in the main text due to space limits and provide detailed results in Appendix G. For RQ1–3, we omit the OpenAudioBench results because models trained only on casual Fisher conversations lack sufficient instruction-following ability for meaningful spoken-QA evaluation. MTR, FDB, and OAB accuracy results in the main text and figures are arithmetic means over four inference runs with seeds from 0 to 3. Appendix G reports means and sample standard deviations ($n = 4$, ddof $= 1$), with incomplete extra controls explicitly marked. CE is evaluated on fixed trained checkpoints. Text-emission and decoding-intervention results retain their original single evaluation passes.

### 4.1 RQ1: Sequence Layout and Speech Granularity

To address RQ1, we compare the interleaved and step-aligned sequence layouts on both GLM-4-Voice and Kimi-Audio at four speech chunk sizes, $s_c \in \{1, 2, 4, 8\}$. Figure 2 summarizes the

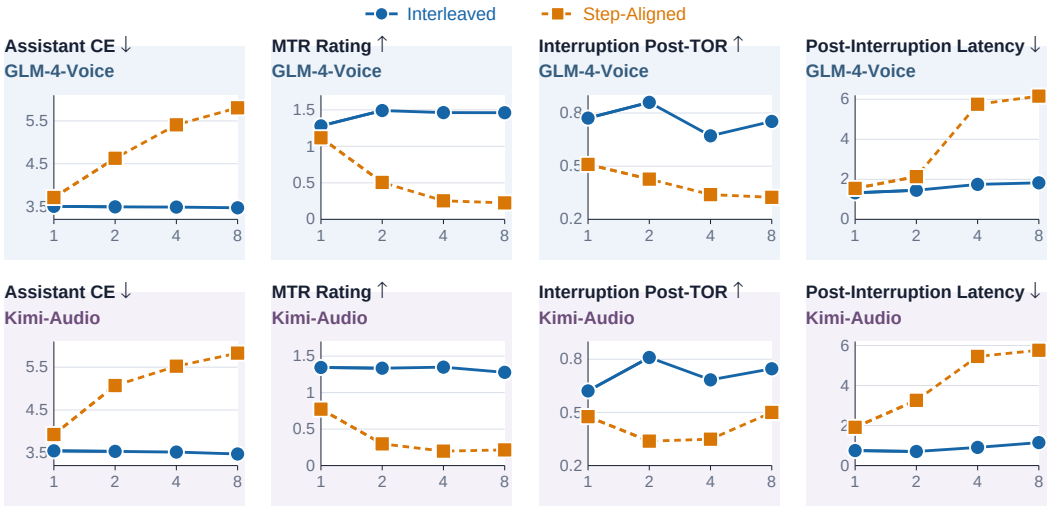



Figure 2: **RQ1 Results Across Sequence Layouts and Speech Granularities.** Rows correspond to GLM-4-Voice and Kimi-Audio; columns report held-out assistant-speech CE, MTR dialogue quality, Post-TOR, and interruption latency.

results using four metrics: assistant-speech CE represents the final validation loss, MTR dialogue quality measures semantic interaction quality, post-interruption take-over rate (Post-TOR) measures whether the model resumes speaking after an interruption, and post-interruption latency measures how quickly it does so. Lower CE and latency and higher MTR and Post-TOR indicate better performance. We highlight the following key observations.

**Interleaved layout outperforms our step-aligned implementation across backbones and speech granularities.** Interleaved obtains a higher MTR dialogue-quality score in all eight matched comparisons. Its advantage over step-aligned ranges from 0.16 to 1.24 points on GLM-4-Voice and from 0.57 to 1.15 points on Kimi-Audio. Interleaved also achieves lower CE than step-aligned at every measured point. At $s_c = 1$, interleaved lowers CE by 0.21 on GLM-4-Voice and by 0.38 on Kimi-Audio; at $s_c = 8$, these gaps increase to 2.34 and 2.36, respectively.

**Step-aligned prediction degrades as speech chunk size increases.** On GLM-4-Voice, step-aligned MTR decreases monotonically from 1.12 at $s_c = 1$ to 0.23 at $s_c = 8$. On Kimi-Audio, it drops from 0.77 at $s_c = 1$ to 0.30 at $s_c = 2$ and remains near the floor at 0.20–0.21 for $s_c \in \{4, 8\}$. The same degradation is evident in the frequency of silent outputs. Among the six configurations with $s_c \geq 2$, step-aligned models produce no response in an average of 633.50 of 1,200 MTR conversations (52.79%), whereas interleaved models do so in an average of only 143.75 (11.98%). Coarser step-aligned prediction therefore produces a failure during autoregressive generation, not only a higher CE.

**The effect of speech chunk size within the interleaved layout is backbone-dependent.** On GLM-4-Voice, the MTR scores at $s_c \in \{2, 4, 8\}$ are 1.49, 1.46, and 1.46, respectively, all higher than the 1.28 obtained at $s_c = 1$, indicating that larger speech chunks generally perform better. Kimi-Audio is less sensitive over $s_c \in \{1, 2, 4\}$, with mean MTR scores of 1.34, 1.33, and 1.35, respectively, while $s_c = 8$ obtains 1.28. These results do not identify a clear preference among the three smaller chunk sizes and support tuning speech granularity separately for each backbone.

**Post-TOR shows no shared granularity trend, whereas interruption latency tends to increase with speech chunk size.** Post-TOR does not change monotonically with $s_c$ across both layouts and backbones. For example, the Post-TOR of interleaved models remains within 0.67–0.86 on GLM-4-Voice and 0.62–0.81 on Kimi-Audio. By contrast, each layout–backbone combination has higher interruption latency at $s_c = 8$ than at $s_c = 1$: it increases from 1.33 to 1.82 for interleaved GLM-4-Voice, from 1.55 to 6.16 for step-aligned GLM-4-Voice, from 0.75 to 1.15 for interleaved Kimi-Audio, and from 1.91 to 5.76 for step-aligned Kimi-Audio. Figure 2 also shows that interleaved has lower interruption latency and higher Post-TOR in all eight matched comparisons.

Table 2: **Effect of Joint User–Assistant Training.** Each cell reports Conditional → Joint. Parenthetical changes are computed from unrounded values and reported as relative percentages for CE, MTR, and Post-TOR, and as absolute differences for latency.

| Backbone | $s_c$ | Assistant CE ↓ | MTR ↑ | Post-TOR ↑ | Post-Interruption Latency ↓ |
|---|---|---|---|---|---|
| GLM-4-Voice | 1 | 3.50 → 3.55 (+1.25%) | 1.28 → 1.28 (−0.39%) | 0.77 → 0.70 (−8.91%) | 1.33 → 1.31 (−0.01) |
| | 4 | 3.49 → 3.51 (+0.58%) | 1.46 → 1.51 (+3.38%) | 0.67 → 0.66 (−1.12%) | 1.75 → 1.52 (−0.22) |
| | 8 | 3.47 → 3.48 (+0.33%) | 1.46 → 1.58 (+8.17%) | 0.75 → 0.76 (+1.50%) | 1.82 → 1.44 (−0.38) |
| Kimi-Audio | 1 | 3.54 → 3.89 (+9.71%) | 1.34 → 1.01 (−24.92%) | 0.62 → 0.76 (+22.13%) | 0.75 → 0.98 (+0.23) |
| | 4 | 3.52 → 3.64 (+3.45%) | 1.35 → 1.10 (−18.49%) | 0.68 → 0.65 (−4.94%) | 0.91 → 0.46 (−0.45) |
| | 8 | 3.47 → 3.57 (+2.70%) | 1.28 → 1.19 (−6.70%) | 0.75 → 0.88 (+17.42%) | 1.15 → 0.37 (−0.78) |

### 4.2 RQ2: Conditional Versus Joint User Modelling

To address RQ2, we compare conditional assistant-only training with joint user–assistant training on both GLM-4-Voice and Kimi-Audio at $s_c \in \{1, 4, 8\}$, using equal loss weights ($\lambda = 1$). Table 2 summarizes assistant CE, MTR dialogue quality, Post-TOR, and post-interruption latency. We highlight the following key observations.

**Joint user modelling consistently increases assistant CE, with a much larger cost on Kimi-Audio.** Joint training improves user-speech prediction: at $s_c = 1$, the user CE of GLM-4-Voice decreases from 8.71 to 3.52. However, assistant CE increases in all six configurations. The relative increases are 1.25%, 0.58%, and 0.33% on GLM-4-Voice, compared with 9.71%, 3.45%, and 2.70% on Kimi-Audio at $s_c \in \{1, 4, 8\}$, respectively. For both backbones, this assistant-side cost decreases as $s_c$ grows.

**The effect on dialogue quality is strongly backbone- and granularity-dependent.** On GLM-4-Voice, mean MTR is nearly unchanged at $s_c = 1$ and increases by 0.05 and 0.12 at $s_c = 4$ and $s_c = 8$, respectively. Kimi-Audio instead has lower MTR in all three comparisons, decreasing from 1.34 to 1.01 at $s_c = 1$, from 1.35 to 1.10 at $s_c = 4$, and from 1.28 to 1.19 at $s_c = 8$. The degradation is therefore largest for Kimi-Audio at small speech chunk sizes and becomes much smaller as $s_c$ grows.

**Joint modelling usually reduces interruption latency, but its effect on Post-TOR varies.** Joint training lowers mean post-interruption latency in five of the six configurations, although the change for GLM-4-Voice at $s_c = 1$ is negligible. Its effect on Post-TOR is mixed, with increases in three configurations and decreases in three. Joint modelling therefore offers gains in selected interaction metrics rather than a uniform improvement in interruption handling.

### 4.3 RQ3: Text Guidance Granularity and Layout

To address RQ3, we compare token-, chunk-, and turn-level text guidance at $s_c = 1$ using both GLM-4-Voice and Kimi-Audio. For GLM-4-Voice, we evaluate two text layouts. GLM Default follows the backbone's original streaming-thought sequence: scheduled text tokens are inserted as autoregressive positions before the corresponding speech, and speech decoding continues without additional text positions after the text is exhausted. GLM Padded instead reserves a text position at every speech step and predicts a text-padding token whenever no scheduled text token remains. We include this padded variant because frame-aligned text padding is common in full-duplex models such as Moshi (Défossez et al., 2024), and because it provides a closer comparison with Kimi-Audio, which also emits a text-padding token after the scheduled text runs out. Figure 3 summarizes assistant-speech CE, MTR dialogue quality, Pre-TOR, Post-TOR, and post-interruption latency. Here, Pre-TOR measures how often the model fails to be interrupted, with lower values indicating that it yields more readily to the interrupting user (see Appendix B). We highlight the following key observations.

**Turn-level text guidance achieves the highest mean dialogue quality.** In the primary layout of each backbone, moving from token- to turn-level guidance lowers assistant-speech CE from 3.19 to 3.13 on GLM-4-Voice and from 3.18 to 3.07 on Kimi-Audio. Turn-level guidance obtains the highest mean MTR score in all three implementations, with a larger gain over chunk-level guidance on Kimi-Audio than on either GLM-4-Voice layout.

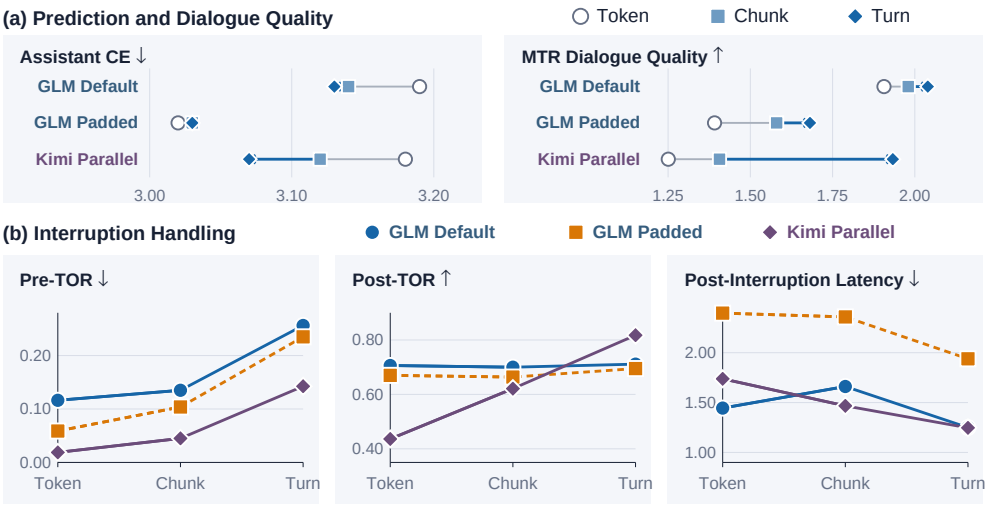



Figure 3: **Effects of Text-Guidance Granularity including token-, chunk-, and turn-level text guidance.** Top: assistant-speech CE and MTR dialogue quality. Bottom: Pre-TOR, Post-TOR, and post-interruption latency.

Table 3: **Effects of Real Versus Synthetic Dialogue Training.** OAB and FDB values are four-run means; text emission retains the original single-pass result. Best values within each backbone are bold. Following the official FDB implementation, post-interruption rating and latency are averaged only over cases in which the model takes over after the interruption (Post-TOR = 1). They should therefore be interpreted together with Post-TOR.

| | | OAB Accuracy (%; Text/Speech) ↑ | | | | | Pause (FDB) | Post-Interruption (FDB) | |
|---|---|---|---|---|---|---|---|---|---|
| Backbone | Data | Llama Q. | TriviaQA | Web Q. | Avg. | Text Emit. (%) ↑ | Avg. TOR ↓ | Latency (s) ↓ | Rating ↑ |
| GLM-4-Voice | Real | 5.17/5.17 | 1.28/1.23 | 3.18/3.18 | 3.21/3.19 | **91.30** | **0.46** | 1.26 | 1.05 |
| | Synthetic | **31.42/28.92** | **10.30/9.83** | **11.68/11.05** | **17.80/16.60** | 74.30 | 0.78 | **0.51** | **2.49** |
| Kimi-Audio | Real | **6.92/6.67** | **2.98/2.75** | **5.05/4.95** | **4.98/4.79** | **91.70** | **0.29** | 1.38 | 1.51 |
| | Synthetic | 1.84/2.25 | 0.75/0.63 | 0.18/0.08 | 0.92/0.98 | 0.00 | 0.73 | **0.78** | **2.66** |

**Turn-level guidance trades easier interruption for faster post-interruption responses.** Across all three implementations, moving from chunk- to turn-level guidance increases mean Pre-TOR and reduces mean post-interruption latency. Turn-level guidance therefore makes models harder to interrupt but faster to respond afterward. The gain in Post-TOR is backbone-dependent: on Kimi-Audio, Post-TOR increases from 0.62 to 0.82, whereas the increases are smaller for GLM Default (0.70 to 0.71) and GLM Padded (0.66 to 0.70).

**Lower CE does not necessarily translate into better dialogue quality.** GLM Padded achieves lower CE than GLM Default but lower MTR at all three text-guidance granularities. At turn level, for example, padding reduces CE from 3.13 to 3.03 while also reducing MTR from 2.04 to 1.68. This contrast suggests that CE alone is insufficient for selecting a text-channel layout.

### 4.4 RQ4: Real Versus Synthetic Dialogue Data

To address RQ4, we compare training on real Fisher conversations with training on synthetic Behavior-SD dialogues, using the interleaved layout with turn-level text guidance at $s_c = 1$ on both backbones. Table 3 summarizes QA accuracy on the three OAB subsets together with text-channel emission and interruption behavior. Each OAB entry reports text/speech accuracy: the first value is judged from the model's text stream, while the second is judged from the Parakeet transcription of its generated speech. Text emission is the percentage of outputs containing at least one non-text-padding token on the 300-question Llama Questions subset. We highlight the following key observations.

**Real dialogue improves pause handling.** Averaged equally across the Candor and synthetic pause benchmarks, real-data training reduces pause-handling TOR from 0.78 to 0.46 on GLM-4-Voice and from 0.73 to 0.29 on Kimi-Audio, indicating fewer premature take-overs during user pauses. We hypothesize that the more varied timing patterns in real conversations help models distinguish pauses within an unfinished turn from opportunities to respond.

**Synthetic dialogue substantially improves QA performance on GLM-4-Voice.** Training on synthetic rather than real dialogue improves both text- and speech-based QA accuracy on all three OAB subsets. Averaged across the subsets, text accuracy increases from 3.21% to 17.80%, while speech accuracy increases from 3.19% to 16.60%.

**Kimi-Audio exhibits text-channel collapse after training on synthetic dialogue.** Kimi-Audio instead has lower QA accuracy with synthetic training on all three subsets: averaged text accuracy decreases from 4.98% to 0.92%, and speech accuracy decreases from 4.79% to 0.98%. This result coincides with a failure to emit normal text tokens: on the Llama Questions subset, the fraction of outputs containing such tokens falls from 91.70% with real training to 0.00% with synthetic training. Both corpora nevertheless present a similarly padding-dominated text objective, with text-padding targets accounting for 84.67% and 85.36% under real and synthetic training. We hypothesize that this severe imbalance makes the text channel vulnerable to collapse, while Behavior-SD's less diverse interaction timing further exacerbates this vulnerability. Appendix E shows that inference interventions can force both checkpoints to emit text, but the synthetic-data checkpoint remains substantially weaker and the real-data checkpoint also improves when premature padding is suppressed.

**Synthetic dialogue improves post-interruption latency and response rating but reduces Post-TOR.** On GLM-4-Voice, synthetic training reduces post-interruption latency from 1.26 to 0.51 seconds and increases the interruption response rating from 1.05 to 2.49. Kimi-Audio shows the same pattern: latency decreases from 1.38 to 0.78 seconds, while rating increases from 1.51 to 2.66. However, mean Post-TOR decreases from 0.71 to 0.52 on GLM-4-Voice and from 0.71 to 0.26 on Kimi-Audio. Synthetic training therefore improves response latency and rating while reducing Post-TOR.

### 4.5 Discussion

**Architecture and Objective Choices.** Interleaved training is preferable when dialogue quality and interaction performance take priority, but uses approximately $2s_c$ times as many Transformer positions as our step-aligned formulation; this conclusion is limited to our summed-input, parallel-head implementation. Joint user–assistant modelling offers no consistent overall gain: it increases assistant CE and has backbone-dependent effects on MTR and Post-TOR. Assistant-only training is therefore a reasonable default for the tested equal-weight objective, although other user-loss weights may behave differently.

**Text Guidance and Dialogue Data.** Turn-level text guidance favors dialogue quality and lower response latency, whereas chunk-level guidance makes models easier to interrupt; lower teacher-forced CE alone is insufficient for selecting the schedule. Synthetic dialogue creates a similar trade-off: it improves spoken QA on GLM-4-Voice and post-interruption latency and rating on both backbones, but reduces Post-TOR and causes text-channel collapse on Kimi-Audio, while real dialogue better handles pauses. We believe that the padding-dominated text objective and reduced timing diversity likely contribute to this failure.

## 5 Conclusion

We present a controlled study of four FD-SLM training choices across two open backbones. Using a shared pipeline, we compare sequence layouts and granularities, duplex objectives, text-guidance schedules, and real versus synthetic dialogue data. Our results favor interleaving and reveal trade-offs among dialogue quality, interruptibility, response latency, and text-channel stability; notably, synthetic data improves some capabilities but can reduce Post-TOR and cause text-channel collapse. By isolating these factors, our work provides practical guidance for building capable and natural full-duplex speech systems.

## AI Use Statement

We use DeepSeek-V4 Pro as an automated evaluator for dialogue quality, spoken question answering, and post-interruption response rating, as described in Section 3. We also use the publicly released Behavior-SD dataset, whose dialogues were generated using language models and text-to-speech synthesis. We use GPT-5.6 Sol and Codex to develop code for this work; all such code was reviewed by the authors. Codex also assisted with figure preparation, manuscript editing, and refinement of the wording of result interpretations. The authors launched the experiments and performed the final analysis of the results. All AI-assisted outputs used in this work were reviewed by the authors, who take responsibility for the code and final content.

## References

Cheng-Kuang Chang, Kai-Wei Chang, Alexander H. Liu, and James Glass. Overcoming state inertia in full-duplex spoken language models via activation steering, 2026. URL https://arxiv.org/abs/2606.11386.

Wenxi Chen, Ziyang Ma, Ruiqi Yan, Yuzhe Liang, Xiquan Li, Ruiyang Xu, Zhikang Niu, Yanqiao Zhu, Yifan Yang, Zhanxun Liu, Kai Yu, Yuxuan Hu, Jinyu Li, Yan Lu, Shujie Liu, and Xie Chen. SLAM-Omni: Timbre-controllable voice interaction system with single-stage training, 2024. URL https://arxiv.org/abs/2412.15649.

Christopher Cieri, David Miller, and Kevin Walker. The fisher corpus: a resource for the next generations of speech-to-text. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, 2004. URL https://aclanthology.org/L04-1500/.

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)*, pp. 13943–13970, 2025.

Wenqian Cui, Xiao-Hui Li, Daxin Tan, Qiyong Zheng, and Irwin King. Minimizing modality gap from the input side: Your speech llm can be a prosody-aware text llm. *arXiv preprint arXiv:2605.05927*, 2026a.

Wenqian Cui, Lei Zhu, Xiaohui Li, Zhihan Guo, Haoli Bai, Lu Hou, and Irwin King. Turnguide: Enhancing meaningful full duplex spoken interactions via dynamic turn-level text-speech interleaving, 2026b. URL https://arxiv.org/abs/2508.07375.

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, 2024. URL https://arxiv.org/abs/2410.00037.

Ke Hu, Ehsan Hosseini-Asl, Chen Chen, Edresson Casanova, Subhankar Ghosh, Piotr Żelasko, Zhehuai Chen, Jason Li, Jagadeesh Balam, and Boris Ginsburg. SALM-Duplex: Efficient and direct duplex modeling for speech-to-speech language model, 2025. URL https://arxiv.org/abs/2505.15670.

KimiTeam, Ding Ding, Zeqian Ju, Yichong Leng, Songxiang Liu, Tong Liu, Zeyu Shang, Kai Shen, Wei Song, Xu Tan, Heyi Tang, Zhengtao Wang, Chu Wei, Yifei Xin, Xinran Xu, Jianwei Yu, Yutao Zhang, Xinyu Zhou, Y. Charles, Jun Chen, Yanru Chen, Yulun Du, Weiran He, Zhenxing Hu, Guokun Lai, Qingcheng Li, Yangyang Liu, Weidong Sun, Jianzhou Wang, Yuzhi Wang, Yuefeng Wu, Yuxin Wu, Dongchao Yang, Hao Yang, Ying Yang, Zhilin Yang, Aoxiong Yin, Ruibin Yuan, Yutong Zhang, and Zaida Zhou. Kimi-audio technical report, 2025. URL https://arxiv.org/abs/2504.18425.

Sehun Lee, Kang-wook Kim, and Gunhee Kim. Behavior-SD: Behaviorally aware spoken dialogue generation with large 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 1: Long Papers)*, pp. 9574–9593, 2025. doi: 10.18653/v1/2025.naacl-long.484. URL https://aclanthology.org/2025.naacl-long.484/.

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, 2025. URL https://arxiv.org/abs/2503.04721.

Hui Lu, Xueyuan Chen, Huimeng Wang, Shuhai Peng, Shiyin Kang, Xixin Wu, and Zhiyong Wu. How should llms listen while speaking? a study of user-stream routing in full-duplex spoken dialogue, 2026a. URL https://arxiv.org/abs/2605.10199.

Jingyu Lu, Yuhan Wang, Jianming Luo, Yifu Chen, Tianle Liang, Shengpeng Ji, Ziyue Jiang, Xiaoda Yang, Yu Zhang, Xize Cheng, Chenyuhao Wen, Changhao Pan, Haoxiao Wang, Chen Ye, Jian Wu, Xiaoxi Jiang, Guanjun Jiang, and Zhou Zhao. A survey of full-duplex spoken dialogue systems: Architectural hierarchy, interaction ontology, and decision state machine, 2026b. URL https://arxiv.org/abs/2606.19453.

Ziyang Ma, Yakun Song, Chenpeng Du, Jian Cong, Zhuo Chen, Yuping Wang, Yuxuan Wang, and Xie Chen. Language model can listen while speaking, 2024. URL https://arxiv.org/abs/2408.02622.

Tu Anh Nguyen, Eugene Kharitonov, Jade Copet, Yossi Adi, Wei-Ning Hsu, Ali Elkahky, Paden Tomasello, Robin Algayres, Benoit Sagot, Abdelrahman Mohamed, and Emmanuel Dupoux. Generative spoken dialogue language modeling, 2022. URL https://arxiv.org/abs/2203.16502.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. Zero: Memory optimizations toward training trillion parameter models, 2020. URL https://arxiv.org/abs/1910.02054.

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, 2026. URL https://arxiv.org/abs/2602.06053.

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*, pp. 21390–21402, 2024. doi: 10.18653/v1/2024.emnlp-main.1192. URL https://aclanthology.org/2024.emnlp-main.1192/.

Qichao Wang, Ziqiao Meng, Wenqian Cui, Yifei Zhang, Pengcheng Wu, Bingzhe Wu, Irwin King, Liang Chen, and Peilin Zhao. Ntpp: Generative speech language modeling for dual-channel spoken dialogue via next-token-pair prediction, 2025. URL https://arxiv.org/abs/2506.00975.

Anyi Xu, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chenchen Ling, et al. Deepseek-v4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*, 2026.

Wang Xu, Shuo Wang, Weilin Zhao, Xu Han, Yukun Yan, Yudi Zhang, Zhe Tao, Zhiyuan Liu, and Wanxiang Che. Enabling real-time conversations with minimal training costs, 2024. URL https://arxiv.org/abs/2409.11727.

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, 2024. URL https://arxiv.org/abs/2412.02612.

He Zhang, Wenqian Cui, Haoning Xu, Xiaohui Li, Lei Zhu, Haoli Bai, Shaohua Ma, and Irwin King. Mtr-duplexbench: Towards a comprehensive evaluation of multi-round conversations for full-duplex speech language models, 2026a. URL https://arxiv.org/abs/2511.10262.

Qinglin Zhang, Luyao Cheng, Chong Deng, Qian Chen, Wen Wang, Siqi Zheng, Jiaqing Liu, Hai Yu, Chaohong Tan, Zhihao Du, and Shiliang Zhang. Omniflatten: An end-to-end gpt model for seamless voice conversation, 2025. URL https://arxiv.org/abs/2410.17799.

Xinrong Zhang, Yingfa Chen, Shengding Hu, Xu Han, Zihang Xu, Yuanwei Xu, Weilin Zhao, Maosong Sun, and Zhiyuan Liu. Beyond the turn-based game: Enabling real-time conversations with duplex models, 2026b. URL https://arxiv.org/abs/2406.15718.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena, 2023. URL https://arxiv.org/abs/2306.05685.

Maike Züfle, Ondrej Klejch, Nicholas Sanders, Jan Niehues, Alexandra Birch, and Tsz Kin Lam. F-Actor: Controllable conversational behaviour in full-duplex models, 2026. URL https://arxiv.org/abs/2601.11329.

## A Limitations

Although our controlled study covers four central design choices across two open backbones, our work still contains several limitations.

1) We focus on pretrained backbones of similar size and English dialogue, so the findings may not transfer directly to models trained from scratch, other languages, or speech tokenizers with different frame rates.

2) We also do not cover Thinker–Talker architectures such as Qwen2.5-Omni, partly because suitable backbones with accessible fine-tuning support for both thinker and talker were difficult to find for our study. Their intermediate text stream also complicates RQ1–3: Thinker text tokens and Talker speech tokens do not share a native frame-level alignment, and the Thinker does not directly predict speech tokens, making our aligned dual-channel speech comparisons difficult to implement without additional architectural changes.

3) We use Fisher and Behavior-SD to represent real and synthetic dialogue data, respectively, but each captures only part of its broader data category. Fisher and Behavior-SD differ in speakers, audio production, topics, turn construction, and segment length; Appendix E quantifies their timing differences, but RQ4 does not independently manipulate these corpus properties. Additionally, RQ4 matches training audio duration rather than optimizer steps: shorter synthetic segments lead to more updates, with 700 steps for real-data training versus 1,212 for synthetic-data training on Kimi-Audio. Data-source effects therefore cannot be fully separated from differences in the optimization process, including their potential contribution to text-channel collapse.

4) MTR, OAB, and parts of FDB rely on ASR and LLM-based judges, while some automatic turn-taking metrics are sensitive to how often a model speaks.

These considerations constrain the scope and causal interpretation of our conclusions, while the shared evaluation pipeline and comparisons across two backbones help assess consistency within the architectures and training settings studied.

## B Interruption Metric Details

We follow the Full-Duplex-Bench (FDB) (Lin et al., 2025) user-interruption protocol and its binary Take-Over Rate (TOR) metric, which treats substantive speech other than silence or a backchannel as a take-over. However, the original FDB metrics do not directly measure whether the model can be successfully interrupted while the user barges in. We therefore introduce Pre-TOR to capture this behavior. Let $t_i^{\text{start}}$ and $t_i^{\text{end}}$ denote the start and end times of the interrupting user utterance in example $i$, and let $\text{TO}(y_i[a : b])$ indicate whether the model takes over within interval $[a, b]$. We compute Pre-TOR by applying this criterion from one second after the interruption starts until the interrupting utterance ends:

$$\text{Pre-TOR} = \frac{1}{N} \sum_{i=1}^{N} \text{TO}\big(y_i[t_i^{\text{start}} + 1\,\text{s} : t_i^{\text{end}}]\big)\,. \tag{4}$$

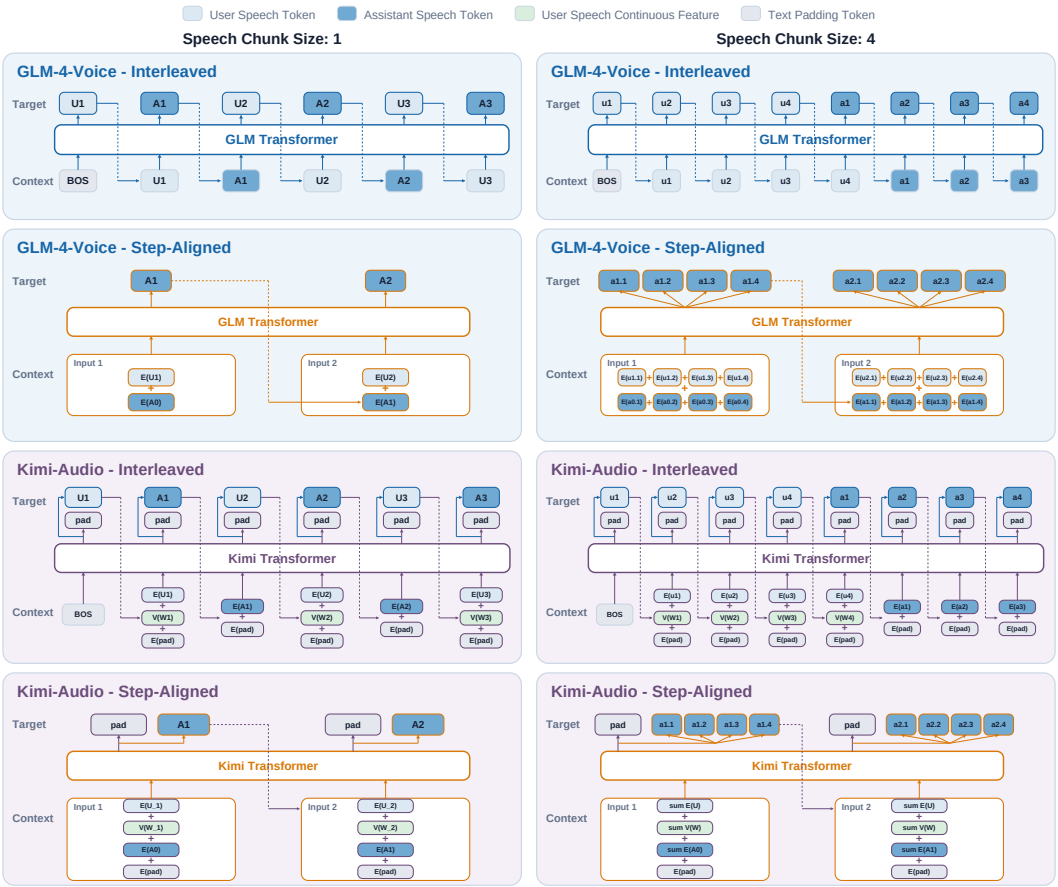



Figure 4: **Sequence Organization and Autoregressive Prediction for RQ1.** Rows show interleaved and step-aligned training for GLM-4-Voice and Kimi-Audio; columns show $s_c \in \{1, 4\}$. Interleaved training uses one speech token per autoregressive position, whereas step-aligned training combines the current user and preceding assistant inputs and predicts assistant chunks through parallel heads. The text-padding token is fixed at every position in Kimi-Audio's input stream rather than generated by the text head.

The first second is excluded to give the model time to yield after detecting the interruption. A lower Pre-TOR therefore indicates that the model is interrupted more successfully. Post-TOR is the original TOR from the FDB user-interruption task, computed after the interrupting utterance ends; we add the prefix "Post" only to distinguish it from Pre-TOR. All other interruption metrics retain their original FDB definitions.

## C RQ1 Backbone-Specific Training Architectures

This section details the backbone-specific sequence construction, input representations, parallel output heads, and training losses used in RQ1. Let $u_t$ and $a_t$ denote the user and assistant speech tokens at frame $t$, and divide each stream into $N$ chunks of $s_c$ tokens, $U_b = (u_{b,1}, \ldots, u_{b,s_c})$ and $A_b = (a_{b,1}, \ldots, a_{b,s_c})$. Here, $b \in \{1, \ldots, N\}$ indexes speech chunks and $r \in \{1, \ldots, s_c\}$ indexes token positions within a chunk. $A_0$ is a chunk of beginning-of-audio tokens. Figure 4 summarizes how the two backbones implement the interleaved and step-aligned layouts.

**Interleaved Training.** Both backbones arrange the speech targets as

$$(U_1, A_1, U_2, A_2, \ldots, U_N, A_N), \tag{5}$$

apply a standard one-position causal shift, and compute loss only on assistant targets. GLM-4-Voice directly feeds the discrete speech-token embeddings into its Transformer and predicts assistant tokens

with its pretrained unified output head. Kimi-Audio retains its native hybrid (continuous + discrete) user representation. Because RQ1 uses speech-only training sequences without assistant text targets, Kimi-Audio fills its parallel text stream with a text-padding token at every autoregressive position. Let $E$ be its shared embedding table for audio and text tokens, $V$ its pretrained continuous-speech adaptor, $w_t$ the aligned continuous user-speech feature, and $x_{\text{pad}}$ the text-padding token. Following Kimi-Audio's native input construction, the combined discrete and continuous user-audio representation is scaled by $\sqrt{2}$. The fused input vectors supplied to the shared Transformer at user and assistant positions are

$$z_t^{\text{user}} = \sqrt{2}\big(E(u_t) + V(w_t)\big) + E(x_{\text{pad}}), \qquad z_t^{\text{asst}} = E(a_t) + E(x_{\text{pad}}). \tag{6}$$

The shared Transformer is followed by Kimi-Audio's pretrained MIMO speech branch and audio output head. The interleaved objective for either backbone is the mean cross-entropy over the $Ns_c$ assistant speech targets.

**Step-Aligned Training.** Step-aligned training aligns the current user input and assistant target at the same autoregressive position. When $s_c = 1$, position $b$ receives the current user token $u_{b,1}$ together with the preceding assistant token $a_{b-1,1}$ and predicts the aligned assistant token $a_{b,1}$. GLM-4-Voice adds the two discrete speech-token embeddings, whereas Kimi-Audio adds its hybrid user representation, the preceding assistant-token embedding, and one text-padding embedding. The resulting causal Transformer state $h_b$ is passed to the backbone's original pretrained speech head $H_1$: the unified output layer of GLM-4-Voice or the MIMO audio output layer of Kimi-Audio.

When $s_c > 1$, we extend both the input and output sides from individual tokens to complete chunks. On the input side, all $s_c$ current-user embeddings and all $s_c$ preceding-assistant embeddings are summed into one autoregressive position. GLM-4-Voice forms

$$z_b^{\text{GLM}} = \sum_{r=1}^{s_c} E(u_{b,r}) + \sum_{r=1}^{s_c} E(a_{b-1,r}), \tag{7}$$

where $E$ is the pretrained GLM-4-Voice speech embedding. Kimi-Audio instead forms

$$z_b^{\text{Kimi}} = \sqrt{2}\sum_{r=1}^{s_c}\big(E(u_{b,r}) + V(w_{b,r})\big) + \sum_{r=1}^{s_c} E(a_{b-1,r}) + E(x_{\text{pad}}). \tag{8}$$

These equations reduce to the single-token inputs described above when $s_c = 1$. On the output side, the same hidden state $h_b$ feeds $s_c$ parallel speech heads $(H_1, \ldots, H_{s_c})$, which jointly predict the assistant chunk $A_b$. The original pretrained head is retained as $H_1$, and every additional head $H_r$, for $r = 2, \ldots, s_c$, is initialized by directly copying the parameters of $H_1$. All heads are then jointly optimized, with $H_r$ predicting the $r$-th token $a_{b,r}$ of each assistant chunk. The per-head loss and final step-aligned objective are

$$\ell_r = \frac{1}{N}\sum_{b=1}^{N} \text{CE}\big(H_r(h_b), a_{b,r}\big), \qquad \mathcal{L}_{\text{step}} = \frac{1}{s_c}\sum_{r=1}^{s_c} \ell_r. \tag{9}$$

Thus, the final loss averages equally over all token positions, and no additional output head is introduced when $s_c = 1$.

## D RQ3 Training Architectures and Text-Alignment Preprocessing

This section details the text-guidance schedules, backbone-specific sequence construction, output heads, training losses, and text-alignment preprocessing used in RQ3. All experiments use $s_c = 1$, so each speech token represents one 80-ms step. Let $u_t$ and $a_t$ denote the user and assistant speech tokens at step $t$, and let $X = (x_1, \ldots, x_M)$ denote the text tokens of an assistant turn. Figure 5 summarizes how the two backbones incorporate these text tokens.

**Text-Guidance Schedule.** We divide $X$ into consecutive groups $G_g = (x_{g,1}, \ldots, x_{g,m_g})$, where $m_g$ is the number of tokens in group $g$. Using the text–speech alignment, we first assign each group to the speech step $\tau_g$ at which its corresponding content begins to be spoken. We then schedule the

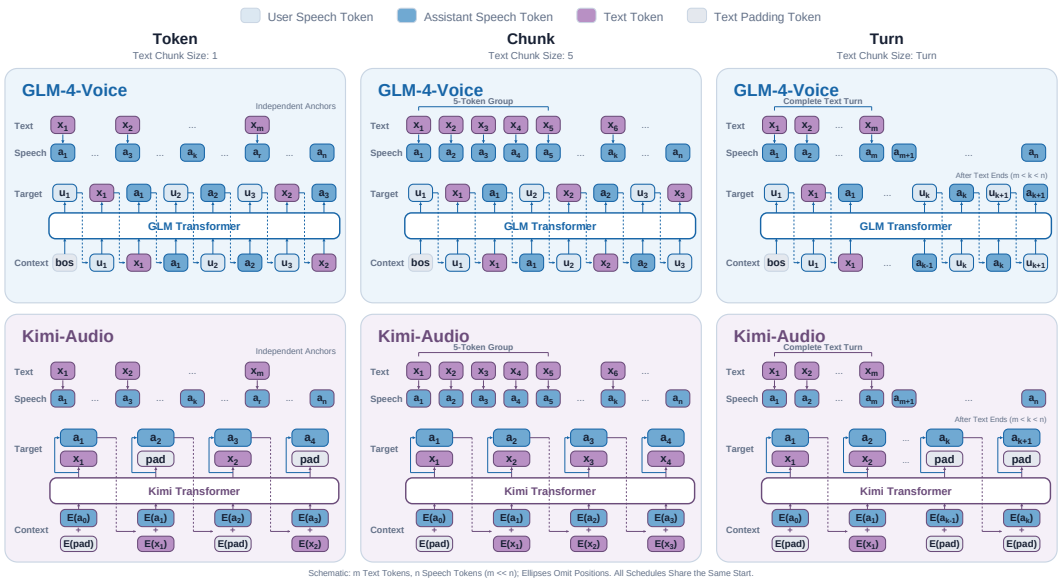



Figure 5: **Text Insertion and Autoregressive Prediction for RQ3.** Columns show the three text-guidance granularities, and rows show GLM-4-Voice and Kimi-Audio. GLM-4-Voice inserts text tokens as additional autoregressive positions, whereas Kimi-Audio predicts text and speech in parallel at each assistant position.

group one speech step earlier so that its text is available before the corresponding speech is generated for better guidance: token $x_{g,r}$ is assigned to step $\tau_g + r - 2$. With token-level guidance ($t_c = 1$), every text token forms its own group and is matched to its own sounding moment. With chunk-level guidance ($t_c = 5$), each group contains up to five consecutive text tokens and is matched to the sounding moment of the group. With turn-level guidance ($t_c = \text{turn}$), the entire assistant turn forms one group whose sounding moment is the turn start. Thus, increasing $t_c$ exposes a longer portion of the text plan earlier in the corresponding speech. We do not combine this comparison with larger speech chunks because the token-level schedule lacks enough autoregressive positions for all text targets: 3.6% cannot be placed at $s_c = 4$, and 29% cannot be placed at $s_c = 8$.

Let $q_t$ denote the assistant text token scheduled at speech step $t$, when one exists. How a step without $q_t$ is represented depends on the model and layout: GLM Default omits the text position entirely, whereas GLM Padded and Kimi-Audio use a text-padding token as described below.

**GLM-4-Voice Training.** GLM-4-Voice uses its pretrained shared vocabulary, embedding table, Transformer, and unified output head for both text and speech. The two modalities occupy separate autoregressive positions. At speech step $t$, GLM Default serializes $[u_t, q_t, a_t]$ when $q_t$ exists and $[u_t, a_t]$ otherwise; it does not insert a text-padding position. GLM Padded instead reserves one text position at every speech step. Let $x_{\text{pad}}$ denote the text-padding token and define

$$\widetilde{q}_t = \begin{cases} q_t, & \text{if a text token is scheduled at step } t, \\ x_{\text{pad}}, & \text{otherwise.} \end{cases} \tag{10}$$

GLM Padded always serializes $[u_t, \widetilde{q}_t, a_t]$. Both layouts apply a standard one-position causal shift. User targets are masked, while assistant-speech tokens and all text positions present in the selected layout are supervised. If $\mathcal{M}_a$ and $\mathcal{M}_x$ denote the assistant-speech and supervised text positions, respectively, the loss is the token-level mean over their union:

$$\mathcal{L}_{\text{GLM}} = \frac{\sum_{i \in \mathcal{M}_a} \text{CE}_i + \sum_{i \in \mathcal{M}_x} \text{CE}_i}{|\mathcal{M}_a| + |\mathcal{M}_x|}. \tag{11}$$

RQ3 therefore introduces no new embedding table or output head for GLM-4-Voice.

**Kimi-Audio Training.** Kimi-Audio retains its native parallel text and speech streams. Unlike GLM Default, its text stream has an output at every assistant step: the target is $q_t$ when text is scheduled

and $x_{\text{pad}}$ otherwise, following the definition of $\widetilde{q}_t$ above. At user positions, it uses the same hybrid discrete–continuous input as in RQ1,

$$z_t^{\text{user}} = \sqrt{2}\big(E(u_t) + V(w_t)\big) + E(x_{\text{pad}}), \tag{12}$$

where $E$ is the shared audio–text embedding table, $V$ is the pretrained continuous-speech adaptor, and $w_t$ is the aligned continuous user-speech feature. After an assistant position, the generated speech token and its parallel text output are embedded and added before entering the next autoregressive position:

$$z_t^{\text{asst}} = E(a_t) + E(\widetilde{q}_t). \tag{13}$$

The causal Transformer state at each assistant prediction position feeds the pretrained text head directly and the pretrained MIMO speech branch followed by the audio head. These heads predict $\widetilde{q}_t$ and $a_t$ in parallel. Thus, unlike GLM Default, Kimi-Audio predicts $x_{\text{pad}}$ whenever no text token is scheduled. Let $\mathcal{M}_a$ and $\mathcal{M}_x$ denote the supervised assistant-speech and assistant-text positions. The two channels are averaged separately and then summed:

$$\mathcal{L}_a = \frac{1}{|\mathcal{M}_a|} \sum_{t \in \mathcal{M}_a} \text{CE}_{a,t}, \qquad \mathcal{L}_x = \frac{1}{|\mathcal{M}_x|} \sum_{t \in \mathcal{M}_x} \text{CE}_{x,t}, \qquad \mathcal{L}_{\text{Kimi}} = \mathcal{L}_a + \mathcal{L}_x. \tag{14}$$

Both output heads retain their pretrained initialization; no new or copied output head is introduced in RQ3.

**Text-Alignment Preprocessing.** We follow the TurnGuide (Cui et al., 2026b) preprocessing pipeline to align each text turn to the corresponding speech sounding moment, but replace its ASR timestamp source with `nvidia/parakeet-tdt-0.6b-v2` because it not only runs significantly faster but also provides more accurate timestamps.

## E RQ4 Corpus Statistics and Kimi-Audio Text-Channel Collapse

This section first compares the real and synthetic training corpora and then analyzes Kimi-Audio's text-channel collapse. Behavior-SD uses the same downstream tokenization, feature extraction, and sequence-construction pipeline as Fisher, but its provided transcript and turn boundaries replace Fisher's ASR, VAD, and turn-construction stages.

**Corpus Measurement Setup.** We analyze the duration-matched RQ4 training subsets before channel swapping. The full subsets contain 45,532 Fisher examples and 77,624 Behavior-SD examples, totaling 1,517.70 and 1,517.80 hours, respectively. For timing statistics, we sample 3,000 examples from each corpus with a fixed random seed and rerun the same `Silero VAD v4` detector on both channels of both corpora. We follow the dGSLM (Nguyen et al., 2022) convention: an inter-pausal unit is bounded by more than 200 ms of silence, and adjacent units from the same speaker separated by a within-speaker pause are regrouped into a turn. Table 4 reports medians with p10–p90 ranges unless otherwise noted.

**Similar Average Activity but Different Timing Diversity.** Fisher and Behavior-SD have similar speech occupancy and turn rates, so their primary difference is not the overall amount of speech. Fisher instead contains approximately twice as much simultaneous speech, more overlap events, and substantially longer overlap. Its timing distributions are also more heterogeneous: the median within-dialogue coefficient of variation of turn duration is 1.08, compared with 0.75 for Behavior-SD. The p99 turn, overlap, gap, and pause durations in Fisher are each 1.70–3.20 times those in Behavior-SD; their observed maxima are 4.30, 8.50, 12.40, and 10.00 times larger, respectively. The direction of the overlap differences is unchanged when the inter-pausal-unit threshold is increased from 200 to 500 ms.

**Inference Interventions Confirm Text-Channel Collapse.** We next re-decode the 300-example Llama Questions subset under four inference settings. Default sampling applies no intervention. Insert START writes the backbone's pretrained assistant-message start token 0.4 s into the 10-second response window. Suppress Padding instead prevents the text head from selecting the text-padding token for the next three steps, optionally using greedy decoding during those steps. These are diagnostic inference-time interventions; they do not change the trained checkpoints.

Table 4: Central tendencies and timing diversity of the duration-matched RQ4 training corpora. Distributional statistics use 3,000 unswapped examples per corpus. The timing-diversity rows report p99 followed by the observed min–max range; bold marks greater overlap or variation, not a preferred outcome.

| Category | Statistic | Fisher (Real) | Behavior-SD (Synthetic) |
|---|---|---|---|
| Scale | Training examples | 45,532 | 77,624 |
| Scale | Example duration (s) | 120.00 (120.00–120.00) | 68.50 (39.90–104.70) |
| Central Tendency | Speech occupancy | 0.82 (0.70–0.90) | 0.79 (0.72–0.85) |
| Central Tendency | Turns per minute | 14.50 (8.00–22.00) | 14.82 (7.87–23.91) |
| Overlap Amount | Simultaneous-speech ratio | **0.06 (0.02–0.15)** | 0.03 (0.01–0.07) |
| Overlap Amount | Events per minute | **6.50 (2.00–12.50)** | 4.24 (1.00–8.97) |
| Overlap Amount | Duration, median (p90), s | **0.44 (1.12)** | 0.35 (0.57) |
| Timing Diversity | Turn-duration CV, median / range | **1.08 / 0.52–2.69** | 0.75 / 0.07–1.46 |
| Timing Diversity | Simultaneous speech, p99 / range | **0.47 / 0–0.71** | 0.11 / 0–0.17 |
| Timing Diversity | Turn duration, p99 / range (s) | **26.40 / 0.29–119.74** | 15.42 / 0.29–28.03 |
| Timing Diversity | Overlap duration, p99 / range (s) | **2.97 / 0.03–16.22** | 0.92 / 0.03–1.92 |
| Timing Diversity | Gap duration, p99 / range (s) | **2.47 / 0.00–42.82** | 1.38 / 0.00–3.46 |
| Timing Diversity | Pause duration, p99 / range (s) | **2.56 / 0.23–36.04** | 1.38 / 0.23–3.59 |

Table 5: Kimi-Audio decoding interventions on the 300-example Llama Questions subset. Each setting uses one generation pass.

| Training Data | Decoding Setting | Text Emission (%) | Text Acc. (%) | Speech Acc. (%) |
|---|---|---|---|---|
| Real | A: Default Sampling | 91.70 | 13.33 | 10.33 |
| Real | B: Insert START | 100.00 | 23.49 | 20.00 |
| Real | C: Suppress Padding | 100.00 | 26.00 | 17.67 |
| Real | D: Suppress Padding + Greedy | 100.00 | **32.11** | **22.00** |
| Synthetic | A: Default Sampling | 0.00 | 0.00 | 2.33 |
| Synthetic | B: Insert START | 100.00 | 7.33 | 2.67 |
| Synthetic | C: Suppress Padding | 100.00 | **9.67** | **3.00** |
| Synthetic | D: Suppress Padding + Greedy | 100.00 | 5.33 | 1.67 |

All three interventions make both checkpoints emit normal text on every example, but they expose a large difference in recoverable capability. For the real-data checkpoint, suppressing padding with greedy decoding raises text accuracy from 13.33% to 32.11% and speech accuracy from 10.33% to 22.00%. Thus, even the real-data checkpoint tends to terminate its text stream prematurely: under default sampling, its median normal-text step ratio—the fraction of generated steps containing a normal text token—is only 0.01 (one normal text-token step), far below the 0.15 ratio in its training labels. Forcing continued text generation reveals substantially higher QA accuracy. The synthetic-data checkpoint can likewise be made to emit text, but its best text and speech accuracies reach only 9.67% and 3.00%. Under the same Suppress Padding intervention, its text accuracy remains 2.70 times lower than the real-data checkpoint (9.67% versus 26.00%). This partial recovery shows that default decoding understates the synthetic checkpoint's QA capability, while its large residual gap supports a more severe text-channel collapse rather than a sampling-only failure. The default rows are a separate seed-0 generation pass from Table 3 and differ slightly because of stochastic decoding; each setting contains 300 examples, with four judge failures across all 2,400 outputs.

**Severe but Similar Text-Target Imbalance Across Corpora.** To measure the labels actually seen by Kimi-Audio, we replay the exact turn-level text scheduler used for training and truncate every sequence to the example's actual length $T$, exactly as the dataset does. We analyze 2,500 samples from each RQ4 training manifest. User positions carry the ignore label and do not contribute to the text loss; at each assistant position, either a normal text token or a text-padding token is supervised. Behavior-SD examples are not padded to 120 seconds, and no trailing text-padding region contributes to the loss. Table 6 reports the resulting target statistics.

Table 6: Kimi-Audio training-side text targets computed at each example's actual sequence length over 2,500 samples per corpus.

| Metric | Fisher (Real) | Behavior-SD (Synthetic) |
|---|---|---|
| Actual length $T$, median (p10–p90) | 1,500 (1,500–1,500) | 872 (503–1,336) |
| Normal text positions, median (p10–p90) | 230 (118–354) | 123 (61–211) |
| Normal-text step ratio | 0.15 | 0.15 |
| Text-padding targets (%) | 84.67 | 85.36 |
| Text segments, median (p10–p90) | 18 (10–26) | 10 (6–16) |
| Padding-to-text transitions per 1,000 steps | 12.00 (6.67–17.33) | 12.06 (6.69–19.20) |

The lower per-example text counts for Behavior-SD track its shorter sequences rather than lower text density: normal-text step ratio, padding share, and text-onset frequency are nearly identical across the two corpora. Nevertheless, approximately 85% of the supervised text targets are padding in both conditions, creating a severely imbalanced objective. We hypothesize that this imbalance makes Kimi-Audio's parallel text channel vulnerable to collapse. Behavior-SD's more homogeneous interaction timing may further exacerbate this vulnerability, but the current experiments do not isolate its contribution from other corpus and training differences.

**The Inference Text-Generation Limit Does Not Explain the Collapse.** The real- and synthetic-trained Kimi-Audio checkpoints use the same turn-level rolling text-emission limits, detailed in Appendix F. The value 0.152 recorded by the decoder denotes the reference training text density, not a hard cap on the generated text ratio. The generation logs contain no refusals triggered by the rolling limits. This inference-side constraint therefore does not account for the observed difference.

# F Training, Decoding, and Judge Configuration

## F.1 Training Configuration

We use AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, weight decay $0.01$), gradient clipping at norm $1.0$, and bfloat16 with ZeRO Stage 2 on eight GPUs. The learning rate peaks at $4 \times 10^{-6}$ after 3% linear warmup and then decays linearly. Training updates only the LLMs; the speech encoders and decoders remain frozen. Head initialization and loss definitions follow Appendices C and D.

The effective batch size is 128 examples. Per-GPU batch size $\times$ gradient accumulation is $4 \times 4$ for GLM speech-only training, $2 \times 8$ for GLM default text guidance, and $1 \times 16$ for GLM padded text and all Kimi runs. Interleaved and step-aligned models therefore receive equal audio batches. A full Fisher example contains 120 seconds (1,500 speech tokens per channel); incomplete trailing speech chunks are discarded. RQ4 also uses fixed example-count batches, so shorter synthetic examples yield less audio per update despite matched total training duration. Recorded schedules include 711 updates for GLM RQ1, 700 for GLM RQ3, and 700/1,212 for Kimi RQ4 real/synthetic training.

Training uses seed 42, shuffled examples, both channel assignments as augmentation, and one epoch; evaluation uses the final checkpoint. Losses average unmasked target tokens within each microbatch, with gradient accumulation combining these microbatch means rather than globally weighting by token count.

## F.2 Decoding Configuration

All checkpoints use temperature $1.0$, top-$p = 0.9$, disabled top-$k$, and no repetition penalty. Generation follows the user timeline at 12.5 Hz: 120 seconds for MTR, the question plus a 10-second response window for OAB, and the benchmark clip duration for FDB. MTR, FDB, and OAB accuracy report means over inference seeds $\{0, 1, 2, 3\}$ at fixed checkpoints, with sample standard deviations in Appendix G. Text-emission and intervention results retain their single evaluation passes.

### F.3 Transcription and Automated Judging

We follow the benchmark evaluation protocols of MTR-DuplexBench (Zhang et al., 2026a), OpenAudioBench (KimiTeam et al., 2025), and FDB (Lin et al., 2025). Our pipeline uses `nvidia/parakeet-tdt-0.6b-v2` for ASR (NeMo defaults, with word timestamps) and DeepSeek-V4 Pro to judge transcripts. MTR/OAB set judge temperature to 0 and maximum completion lengths to 16/256 tokens, without an explicit judge seed. FDB sets judge seed 0 and leaves temperature and completion length at endpoint defaults. Each example receives one judgment. Following FDB, post-interruption rating and latency are conditional on Post-TOR $= 1$ and should be interpreted together with Post-TOR.

### F.4 Full Judge Prompts

The following blocks reproduce the prompt wording, with whitespace normalized for display. Braced names denote values inserted for each example. For the JSON example, the displayed braces are literal braces in the submitted prompt.

**MTR User Message.**

```
{transcript}

Please evaluate the following two-speaker dialogue transcript for how
meaningful the speech is (based on its content), focusing on the model
channel's speech across the whole conversation. Use the following scale:

0: Completely meaningless; no coherent sentences, random words, or
unintelligible.

0.5: Almost no meaning; isolated words or phrases, but no understandable
ideas.

1: Extremely low meaning; rare, vague fragments of ideas, but mostly
incoherent or off-topic.

1.5: Very little meaning; a few short, unclear ideas, but mostly
disjointed or confusing.

2: Low meaning; some recognizable ideas or topics, but mostly unclear,
incomplete, or off-topic.

2.5: Somewhat low meaning; a few coherent points, but overall lacks
clarity or logical flow.

3: Moderate meaning; general topic is understandable, but there are gaps,
unclear parts, or weak connections.

3.5: Fairly meaningful; mostly coherent and relevant, but with some
confusion, repetition, or lack of detail.

4: Meaningful; clear and logical, with relevant and connected ideas,
though may lack depth or detail.

4.5: Very meaningful; almost fully coherent, with well-developed,
relevant, and connected ideas.

5: Extremely meaningful; highly coherent, clear, and detailed, with all
ideas well connected and relevant.

Only output the final score (0, 0.5, 1, 1.5, ..., 5)

**ONLY** according to the above rubric. Do not output anything else.
```

**OAB Llama Questions User Message.**

```
You are a professional QA evaluation expert. You need to assess whether
the model's answer is correct based on the standard answer.
```

```
Provide the reasoning for your score, then generate the result in "[]"
format and make sure it contains "the score is [Correct]" or "the score
is [Incorrect]", for example:

The answer is correct and equivalent to the standard answer, the score is
[Correct]

The answer is incorrect and does not match the standard answer, the score
is [Incorrect]

# Question

{question}

# Standard Answer

{reference}

# Model Answer

{answer}
```

**OAB Other Subsets User Message.**

```
Your will be given a question, the reference answers to that question,
and an answer to be judged. Your tasks is to judge whether the answer to
be judged is correct, given the question and reference answers. An answer
considered correct expresses or contains the same meaning as at least
**one of** the reference answers. The format and the tone of the response
does not matter.

You should respond in JSON format. First provide a one-sentence concise
analysis for the judgement in field 'analysis', then your judgment in field
'judgment'. For example,

'''json

{"analysis": "<a one-sentence concise analysis for the judgement>",
"judgment": < your final judgment, "correct" or "incorrect">}

'''

# Question

{question}

# Reference Answer

{reference}

# Answer To Be Judged

{answer}
```

**FDB Interruption System Message.**

```
The scenario is that the user and AI are talking in the spoken
conversation.

The user first speaks, then the AI responds. But when AI is speaking, the
user interrupts the AI's turn.

Your task is to rate the quality of AI's response after the user
interrupt the turn.

Below is the rating guideline (from 0 to 5, 0 is the worst and 5 is the
best):

- 0: The AI's response is totally unrelated to the user's interrupting
turn.

- 1: The AI's response is not related to the user's interrupting turn.
```

```
- 2: The AI's response is slightly related to the user's interrupting
turn.
- 3: The AI's response is related to the user's interrupting turn.
- 4: The AI's response is highly related to the user's interrupting turn.
- 5: The AI's response is perfectly related to the user's interrupting
turn.

Firstly, briefly analyze the user's interrupting turn and the AI's
response
Then, you must return the overall output as the following format:
Analysis: [Your analysis].
I would rate the AI's response as [Rating].
```

**FDB Interruption User Message.**

```
- Contextual user turn: {in_before_interrupt_text}
- User interrupting turn: {in_interrupt_text}
- AI's response: {out_after_interrupt_text}
```

## G Detailed Main Evaluation Tables

Tables 7–18 report prediction, dialogue quality, and FDB timing metrics for each RQ. OAB results are omitted for RQ1–3 for the reasons discussed in the main text. CE denotes assistant-speech cross-entropy; comparisons are meaningful only within a backbone and on the same validation data. A dash denotes an unavailable result, not zero.

FDB columns follow the task grouping of the original benchmark. Latencies are in seconds; TOR and frequency are reported as fractions. Rating uses our LLM judge. Backchannel frequency has no directional arrow because a higher frequency is not necessarily better. Pre-TOR is defined in Appendix B.

MTR, FDB, and OAB accuracy entries show mean ($\pm$ sample standard deviation) across four inference runs (original and seeds 1–3; $n = 4$, ddof $= 1$). CE is fixed-checkpoint validation. Entries marked † retain the original control result because four-run coverage is unavailable; no standard deviation is estimated for these entries.

Table 7: RQ1 prediction and dialogue-quality results.

| Backbone | Layout | $s_c$ | CE ↓ | MTR ↑ |
|---|---|---|---|---|
| GLM-4-Voice | Interleaved | 1 | 3.50 | 1.28 (±0.15) |
| GLM-4-Voice | Interleaved | 2 | 3.49 | 1.49 (±0.19) |
| GLM-4-Voice | Interleaved | 4 | 3.49 | 1.46 (±0.17) |
| GLM-4-Voice | Interleaved | 8 | 3.47 | 1.46 (±0.06) |
| GLM-4-Voice | Step-Aligned | 1 | 3.71 | 1.12 (±0.11) |
| GLM-4-Voice | Step-Aligned | 2 | 4.63 | 0.51 (±0.02) |
| GLM-4-Voice | Step-Aligned | 4 | 5.41 | 0.25 (±0.07) |
| GLM-4-Voice | Step-Aligned | 8 | 5.81 | 0.23 (±0.05) |
| Kimi-Audio | Interleaved | 1 | 3.54 | 1.34 (±0.27) |
| Kimi-Audio | Interleaved | 2 | 3.53 | 1.33 (±0.07) |
| Kimi-Audio | Interleaved | 4 | 3.52 | 1.35 (±0.07) |
| Kimi-Audio | Interleaved | 8 | 3.47 | 1.28 (±0.05) |
| Kimi-Audio | Step-Aligned | 1 | 3.93 | 0.77 (±0.09) |
| Kimi-Audio | Step-Aligned | 2 | 5.07 | 0.30 (±0.05) |
| Kimi-Audio | Step-Aligned | 4 | 5.52 | 0.20 (±0.07) |
| Kimi-Audio | Step-Aligned | 8 | 5.83 | 0.21 (±0.05) |

Table 8: RQ1 FDB results: pause handling and backchanneling. Entries show mean (± sample standard deviation) across four inference runs unless marked.

| Backbone | Setting | $s_c$ | Pause Synth. TOR ↓ | Pause Candor TOR ↓ | ICC TOR ↓ | ICC Frequency | ICC JSD ↓ |
|---|---|---|---|---|---|---|---|
| GLM | Interleaved | 1 | 0.22 (±0.17) | 0.36 (±0.17) | 0.27 (±0.13) | 0.08 (±0.02) | 0.77 (±0.03) |
| GLM | Interleaved | 2 | 0.22 (±0.15) | 0.37 (±0.14) | 0.26 (±0.07) | 0.10 (±0.03) | 0.76 (±0.03) |
| GLM | Interleaved | 4 | 0.18 (±0.09) | 0.33 (±0.11) | 0.21 (±0.05) | 0.08 (±0.01) | 0.77 (±0.02) |
| GLM | Interleaved | 8 | 0.17 (±0.06) | 0.33 (±0.11) | 0.21 (±0.14) | 0.09 (±0.01) | 0.76 (±0.01) |
| GLM | Step-Aligned | 1 | 0.40 (±0.24) | 0.41 (±0.19) | 0.60 (±0.17) | 0.12 (±0.03) | 0.74 (±0.04) |
| GLM | Step-Aligned | 2 | 0.40 (±0.08) | 0.39 (±0.11) | 0.66 (±0.14) | 0.18 (±0.02) | 0.68 (±0.02) |
| GLM | Step-Aligned | 4 | 0.29 (±0.16) | 0.31 (±0.06) | 0.43 (±0.07) | 0.20 (±0.02) | 0.69 (±0.02) |
| GLM | Step-Aligned | 8 | 0.13 (±0.07) | 0.16 (±0.09) | 0.17 (±0.07) | 0.12 (±0.04) | 0.76 (±0.04) |
| Kimi | Interleaved | 1 | 0.43 (±0.32) | 0.56 (±0.34) | 0.37 (±0.21) | 0.06 (±0.03) | 0.81 (±0.06) |
| Kimi | Interleaved | 2 | 0.61 (±0.28) | 0.54 (±0.11) | 0.25 (±0.06) | 0.05 (±0.02) | 0.85 (±0.03) |
| Kimi | Interleaved | 4 | 0.31 (±0.13) | 0.48 (±0.06) | 0.53 (±0.13) | 0.07 (±0.01) | 0.77 (±0.01) |
| Kimi | Interleaved | 8 | 0.35 (±0.18) | 0.44 (±0.11) | 0.44 (±0.12) | 0.08 (±0.01) | 0.77 (±0.01) |
| Kimi | Step-Aligned | 1 | 0.27 (±0.10) | 0.28 (±0.06) | 0.27 (±0.06) | 0.05 (±0.01) | 0.82 (±0.02) |
| Kimi | Step-Aligned | 2 | 0.08 (±0.04) | 0.10 (±0.02) | 0.40 (±0.05) | 0.06 (±0.01) | 0.80 (±0.02) |
| Kimi | Step-Aligned | 4 | 0.05 (±0.03) | 0.08 (±0.01) | 0.50 (±0.07) | 0.07 (±0.00) | 0.78 (±0.01) |
| Kimi | Step-Aligned | 8 | 0.25 (±0.09) | 0.14 (±0.05) | 0.36 (±0.06) | 0.07 (±0.01) | 0.78 (±0.01) |

Table 9: RQ1 FDB results: smooth turn taking and user interruption. Entries show mean (± sample standard deviation) across four inference runs unless marked.

| Backbone | Setting | $s_c$ | Candor TT TOR ↑ | Candor TT Latency ↓ | Interrupt Pre-TOR ↓ | Interrupt Post-TOR ↑ | Interrupt Rating ↑ | Interrupt Latency ↓ |
|---|---|---|---|---|---|---|---|---|
| GLM | Interleaved | 1 | 0.78 (±0.10) | 0.52 (±0.18) | 0.07 (±0.02) | 0.77 (±0.04) | 1.18 (±0.14) | 1.33 (±0.22) |
| GLM | Interleaved | 2 | 0.78 (±0.07) | 0.43 (±0.30) | 0.06 (±0.02) | 0.86 (±0.03) | 1.20 (±0.10) | 1.45 (±0.36) |
| GLM | Interleaved | 4 | 0.69 (±0.06) | 0.46 (±0.21) | 0.05 (±0.03) | 0.67 (±0.02) | 1.15 (±0.13) | 1.75 (±0.50) |
| GLM | Interleaved | 8 | 0.72 (±0.07) | 0.46 (±0.26) | 0.06 (±0.02) | 0.75 (±0.06) | 1.16 (±0.06) | 1.82 (±0.56) |
| GLM | Step-Aligned | 1 | 0.74 (±0.09) | 0.53 (±0.37) | 0.28 (±0.15) | 0.51 (±0.12) | 0.81 (±0.08) | 1.55 (±0.53) |
| GLM | Step-Aligned | 2 | 0.69 (±0.06) | 0.61 (±0.11) | 0.19 (±0.04) | 0.43 (±0.08) | 0.13 (±0.06) | 2.14 (±0.26) |
| GLM | Step-Aligned | 4 | 0.52 (±0.07) | 0.90 (±0.52) | 0.05 (±0.02) | 0.34 (±0.09) | 0.14 (±0.08) | 5.75 (±1.53) |
| GLM | Step-Aligned | 8 | 0.47 (±0.14) | 1.35 (±0.56) | 0.01 (±0.01) | 0.32 (±0.12) | 0.19 (±0.09) | 6.16 (±2.30) |
| Kimi | Interleaved | 1 | 0.82 (±0.13) | 0.18 (±0.20) | 0.17 (±0.09) | 0.62 (±0.17) | 0.85 (±0.08) | 0.75 (±0.33) |
| Kimi | Interleaved | 2 | 0.81 (±0.06) | 0.14 (±0.08) | 0.22 (±0.05) | 0.81 (±0.04) | 0.94 (±0.06) | 0.70 (±0.26) |
| Kimi | Interleaved | 4 | 0.78 (±0.06) | 0.25 (±0.11) | 0.11 (±0.04) | 0.68 (±0.05) | 1.03 (±0.14) | 0.91 (±0.19) |
| Kimi | Interleaved | 8 | 0.84 (±0.01) | 0.30 (±0.10) | 0.15 (±0.03) | 0.75 (±0.06) | 0.99 (±0.08) | 1.15 (±0.32) |
| Kimi | Step-Aligned | 1 | 0.58 (±0.07) | 0.40 (±0.18) | 0.05 (±0.02) | 0.48 (±0.06) | 0.21 (±0.06) | 1.91 (±1.02) |
| Kimi | Step-Aligned | 2 | 0.44 (±0.10) | 0.77 (±0.16) | 0.01 (±0.01) | 0.34 (±0.01) | 0.10 (±0.05) | 3.26 (±0.43) |
| Kimi | Step-Aligned | 4 | 0.43 (±0.07) | 0.86 (±0.26) | 0.00 (±0.00) | 0.35 (±0.05) | 0.05 (±0.03) | 5.46 (±0.70) |
| Kimi | Step-Aligned | 8 | 0.44 (±0.03) | 1.04 (±0.33) | 0.06 (±0.04) | 0.50 (±0.15) | 0.12 (±0.05) | 5.76 (±2.28) |

Table 10: RQ2 prediction and dialogue-quality results.

| Backbone | Objective | $s_c$ | CE ↓ | MTR ↑ |
|---|---|---|---|---|
| GLM-4-Voice | Conditional | 1 | 3.50 | 1.28 (±0.15) |
| GLM-4-Voice | Conditional | 4 | 3.49 | 1.46 (±0.17) |
| GLM-4-Voice | Conditional | 8 | 3.47 | 1.46 (±0.06) |
| GLM-4-Voice | Joint | 1 | 3.55 | 1.28 (±0.15) |
| GLM-4-Voice | Joint | 4 | 3.51 | 1.51 (±0.14) |
| GLM-4-Voice | Joint | 8 | 3.48 | 1.58 (±0.10) |
| Kimi-Audio | Conditional | 1 | 3.54 | 1.34 (±0.27) |
| Kimi-Audio | Conditional | 4 | 3.52 | 1.35 (±0.07) |
| Kimi-Audio | Conditional | 8 | 3.47 | 1.28 (±0.05) |
| Kimi-Audio | Joint | 1 | 3.89 | 1.01 (±0.21) |
| Kimi-Audio | Joint | 4 | 3.64 | 1.10 (±0.04) |
| Kimi-Audio | Joint | 8 | 3.57 | 1.19 (±0.04) |

Table 11: RQ2 FDB results: pause handling and backchanneling. Entries show mean (± sample standard deviation) across four inference runs unless marked.

| Backbone | Setting | $s_c$ | Pause Synth. TOR ↓ | Pause Candor TOR ↓ | ICC TOR ↓ | ICC Frequency | ICC JSD ↓ |
|---|---|---|---|---|---|---|---|
| GLM | Conditional | 1 | 0.22 (±0.17) | 0.36 (±0.17) | 0.27 (±0.13) | 0.08 (±0.02) | 0.77 (±0.03) |
| GLM | Conditional | 4 | 0.18 (±0.09) | 0.33 (±0.11) | 0.21 (±0.05) | 0.08 (±0.01) | 0.77 (±0.02) |
| GLM | Conditional | 8 | 0.17 (±0.06) | 0.33 (±0.11) | 0.21 (±0.14) | 0.09 (±0.01) | 0.76 (±0.01) |
| GLM | Joint | 1 | 0.29 (±0.20) | 0.27 (±0.14) | 0.24 (±0.13) | 0.07 (±0.02) | 0.78 (±0.04) |
| GLM | Joint | 4 | 0.22 (±0.09) | 0.33 (±0.09) | 0.20 (±0.01) | 0.09 (±0.01) | 0.77 (±0.01) |
| GLM | Joint | 8 | 0.20 (±0.08) | 0.36 (±0.10) | 0.29 (±0.06) | 0.10 (±0.01) | 0.76 (±0.02) |
| Kimi | Conditional | 1 | 0.43 (±0.32) | 0.56 (±0.34) | 0.37 (±0.21) | 0.06 (±0.03) | 0.81 (±0.06) |
| Kimi | Conditional | 4 | 0.31 (±0.13) | 0.48 (±0.06) | 0.53 (±0.13) | 0.07 (±0.01) | 0.77 (±0.01) |
| Kimi | Conditional | 8 | 0.35 (±0.18) | 0.44 (±0.11) | 0.44 (±0.12) | 0.08 (±0.01) | 0.77 (±0.01) |
| Kimi | Joint | 1 | 0.34 (±0.36) | 0.39 (±0.28) | 0.41 (±0.18) | 0.05 (±0.01) | 0.84 (±0.05) |
| Kimi | Joint | 4 | 0.77 (±0.05) | 0.57 (±0.03) | 0.52 (±0.11) | 0.06 (±0.00) | 0.79 (±0.01) |
| Kimi | Joint | 8 | 0.76 (±0.04) | 0.51 (±0.02) | 0.47 (±0.11) | 0.07 (±0.00) | 0.79 (±0.01) |

Table 12: RQ2 FDB results: smooth turn taking and user interruption. Entries show mean (± sample standard deviation) across four inference runs unless marked.

| Backbone | Setting | $s_c$ | Candor TT TOR ↑ | Candor TT Latency ↓ | Interrupt Pre-TOR ↓ | Interrupt Post-TOR ↑ | Interrupt Rating ↑ | Interrupt Latency ↓ |
|---|---|---|---|---|---|---|---|---|
| GLM | Conditional | 1 | 0.78 (±0.10) | 0.52 (±0.18) | 0.07 (±0.02) | 0.77 (±0.04) | 1.18 (±0.14) | 1.33 (±0.22) |
| GLM | Conditional | 4 | 0.69 (±0.06) | 0.46 (±0.21) | 0.05 (±0.03) | 0.67 (±0.02) | 1.15 (±0.13) | 1.75 (±0.50) |
| GLM | Conditional | 8 | 0.72 (±0.07) | 0.46 (±0.26) | 0.06 (±0.02) | 0.75 (±0.06) | 1.16 (±0.06) | 1.82 (±0.56) |
| GLM | Joint | 1 | 0.71 (±0.06) | 0.54 (±0.21) | 0.05 (±0.02) | 0.70 (±0.11) | 1.21 (±0.19) | 1.31 (±0.14) |
| GLM | Joint | 4 | 0.71 (±0.05) | 0.40 (±0.14) | 0.05 (±0.01) | 0.66 (±0.06) | 1.18 (±0.11) | 1.52 (±0.27) |
| GLM | Joint | 8 | 0.79 (±0.04) | 0.48 (±0.18) | 0.06 (±0.01) | 0.76 (±0.05) | 1.24 (±0.10) | 1.44 (±0.19) |
| Kimi | Conditional | 1 | 0.82 (±0.13) | 0.18 (±0.20) | 0.17 (±0.09) | 0.62 (±0.17) | 0.85 (±0.08) | 0.75 (±0.33) |
| Kimi | Conditional | 4 | 0.78 (±0.06) | 0.25 (±0.11) | 0.11 (±0.04) | 0.68 (±0.05) | 1.03 (±0.14) | 0.91 (±0.19) |
| Kimi | Conditional | 8 | 0.84 (±0.01) | 0.30 (±0.10) | 0.15 (±0.03) | 0.75 (±0.06) | 0.99 (±0.08) | 1.15 (±0.32) |
| Kimi | Joint | 1 | 0.80 (±0.10) | 0.33 (±0.16) | 0.08 (±0.07) | 0.76 (±0.22) | 0.16 (±0.02) | 0.98 (±0.22) |
| Kimi | Joint | 4 | 0.78 (±0.04) | 0.11 (±0.06) | 0.13 (±0.03) | 0.65 (±0.07) | 0.65 (±0.10) | 0.46 (±0.07) |
| Kimi | Joint | 8 | 0.89 (±0.03) | 0.15 (±0.10) | 0.21 (±0.03) | 0.88 (±0.05) | 0.89 (±0.08) | 0.37 (±0.02) |

Table 13: RQ3 prediction and dialogue-quality results. A text-free control is also included.

| Backbone | Layout / Guidance | $s_c$ | CE ↓ | MTR ↑ |
|---|---|---|---|---|
| GLM-4-Voice | Default / Token | 1 | 3.19 | 1.91 (±0.12) |
| GLM-4-Voice | Default / Chunk | 1 | 3.14 | 1.98 (±0.14) |
| GLM-4-Voice | Default / Turn | 1 | 3.13 | 2.04 (±0.13) |
| GLM-4-Voice | Padded / Token | 1 | 3.02 | 1.39 (±0.09) |
| GLM-4-Voice | Padded / Chunk | 1 | 3.03 | 1.58 (±0.12) |
| GLM-4-Voice | Padded / Turn | 1 | 3.03 | 1.68 (±0.13) |
| Kimi-Audio | Parallel / Token | 1 | 3.18 | 1.25 (±0.04) |
| Kimi-Audio | Parallel / Chunk | 1 | 3.12 | 1.41 (±0.10) |
| Kimi-Audio | Parallel / Turn | 1 | 3.07 | 1.93 (±0.15) |
| GLM-4-Voice | Padded / Text-Free | 1 | 3.42 | 0.87† |

Table 14: RQ3 FDB results: pause handling and backchanneling. Entries show mean (± sample standard deviation) across four inference runs unless marked.

| Backbone | Setting | $s_c$ | Pause Synth. TOR ↓ | Pause Candor TOR ↓ | ICC TOR ↓ | ICC Frequency | ICC JSD ↓ |
|---|---|---|---|---|---|---|---|
| GLM | Default / Token | 1 | 0.47 (±0.25) | 0.45 (±0.22) | 0.35 (±0.09) | 0.09 (±0.03) | 0.76 (±0.02) |
| GLM | Default / Chunk | 1 | 0.36 (±0.23) | 0.45 (±0.27) | 0.35 (±0.11) | 0.09 (±0.01) | 0.76 (±0.02) |
| GLM | Default / Turn | 1 | 0.41 (±0.24) | 0.47 (±0.25) | 0.51 (±0.08) | 0.08 (±0.02) | 0.77 (±0.04) |
| GLM | Padded / Token | 1 | 0.74 (±0.17) | 0.68 (±0.10) | 0.32 (±0.13) | 0.11 (±0.02) | 0.73 (±0.01) |
| GLM | Padded / Chunk | 1 | 0.79 (±0.11) | 0.73 (±0.11) | 0.43 (±0.14) | 0.10 (±0.02) | 0.75 (±0.03) |
| GLM | Padded / Turn | 1 | 0.74 (±0.17) | 0.68 (±0.16) | 0.39 (±0.11) | 0.10 (±0.02) | 0.76 (±0.02) |
| Kimi | Parallel / Token | 1 | 0.29 (±0.24) | 0.51 (±0.23) | 0.41 (±0.26) | 0.05 (±0.02) | 0.81 (±0.05) |
| Kimi | Parallel / Chunk | 1 | 0.35 (±0.27) | 0.55 (±0.27) | 0.33 (±0.13) | 0.06 (±0.03) | 0.81 (±0.06) |
| Kimi | Parallel / Turn | 1 | 0.16 (±0.14) | 0.40 (±0.20) | 0.34 (±0.27) | 0.05 (±0.02) | 0.82 (±0.05) |
| GLM | Padded / Text-Free | 1 | 0.04† | 0.15† | 0.18† | 0.10† | 0.75† |

Table 15: RQ3 FDB results: smooth turn taking and user interruption. Entries show mean (± sample standard deviation) across four inference runs unless marked.

| Backbone | Setting | $s_c$ | Candor TT TOR ↑ | Candor TT Latency ↓ | Interrupt Pre-TOR ↓ | Interrupt Post-TOR ↑ | Interrupt Rating ↑ | Interrupt Latency ↓ |
|---|---|---|---|---|---|---|---|---|
| GLM | Default / Token | 1 | 0.77 (±0.11) | 0.54 (±0.39) | 0.12 (±0.06) | 0.71 (±0.05) | 1.19 (±0.25) | 1.44 (±0.23) |
| GLM | Default / Chunk | 1 | 0.79 (±0.13) | 0.44 (±0.32) | 0.14 (±0.05) | 0.70 (±0.05) | 1.25 (±0.08) | 1.66 (±0.32) |
| GLM | Default / Turn | 1 | 0.78 (±0.12) | 0.48 (±0.36) | 0.26 (±0.04) | 0.71 (±0.04) | 1.14 (±0.33) | 1.25 (±0.20) |
| GLM | Padded / Token | 1 | 0.82 (±0.07) | 0.11 (±0.08) | 0.06 (±0.01) | 0.67 (±0.07) | 0.23 (±0.04) | 2.40 (±0.25) |
| GLM | Padded / Chunk | 1 | 0.87 (±0.04) | 0.04 (±0.03) | 0.10 (±0.02) | 0.66 (±0.02) | 0.32 (±0.07) | 2.36 (±0.27) |
| GLM | Padded / Turn | 1 | 0.79 (±0.08) | 0.08 (±0.05) | 0.24 (±0.03) | 0.70 (±0.05) | 0.23 (±0.08) | 1.94 (±0.28) |
| Kimi | Parallel / Token | 1 | 0.82 (±0.06) | 0.15 (±0.15) | 0.02 (±0.01) | 0.44 (±0.05) | 1.10 (±0.22) | 1.74 (±0.31) |
| Kimi | Parallel / Chunk | 1 | 0.88 (±0.06) | 0.17 (±0.11) | 0.05 (±0.02) | 0.62 (±0.04) | 1.10 (±0.17) | 1.47 (±0.07) |
| Kimi | Parallel / Turn | 1 | 0.91 (±0.06) | 0.27 (±0.13) | 0.14 (±0.03) | 0.82 (±0.03) | 1.49 (±0.06) | 1.25 (±0.20) |
| GLM | Padded / Text-Free | 1 | 0.49† | 1.23† | 0.03† | 0.64† | 0.24† | 2.37† |

Table 16: RQ4 prediction and dialogue-quality results. All models use turn-level text guidance; CE measures assistant-speech prediction. OAB reports text/speech accuracy (%) as the mean (± sample standard deviation) across four inference runs. Avg. is computed across the three subsets within each run before aggregation. MTR likewise reports four-run means and sample standard deviations.

| | | | | OAB Accuracy (%) ↑ | | | |
|---|---|---|---|---|---|---|---|
| Backbone | Data | CE ↓ | MTR ↑ | Llama Q. | TriviaQA | Web Q. | Avg. |
| GLM-4-Voice | Real | 3.13 | 2.13 (±0.28) | 5.17 (±2.08)/5.17 (±1.92) | 1.28 (±0.60)/1.23 (±0.63) | 3.18 (±1.43)/3.18 (±1.36) | 3.21 (±0.79)/3.19 (±0.65) |
| GLM-4-Voice | 50% Synthetic | 3.14 | 2.23 (±0.13) | 29.08 (±3.40)/27.59 (±3.94) | 9.23 (±4.08)/8.75 (±3.54) | 7.56 (±1.58)/7.40 (±1.59) | 15.29 (±2.43)/14.58 (±1.80) |
| GLM-4-Voice | Synthetic | 4.10 | 1.58 (±0.28) | 31.42 (±5.18)/28.92 (±3.96) | 10.30 (±3.24)/9.83 (±2.68) | 11.68 (±3.28)/11.05 (±3.47) | 17.80 (±2.73)/16.60 (±2.17) |
| Kimi-Audio | Real | 3.07 | 1.92 (±0.19) | 6.92 (±3.54)/6.67 (±3.31) | 2.98 (±1.31)/2.75 (±0.76) | 5.05 (±3.15)/4.95 (±3.04) | 4.98 (±2.35)/4.79 (±1.97) |
| Kimi-Audio | 50% Synthetic | 3.10 | 1.95 (±0.17) | 1.42 (±1.57)/1.67 (±1.30) | 0.30 (±0.23)/0.23 (±0.10) | 0.25 (±0.30)/0.28 (±0.43) | 0.66 (±0.64)/0.72 (±0.58) |
| Kimi-Audio | Synthetic | 3.67 | 1.68 (±0.22) | 1.84 (±1.35)/2.25 (±1.17) | 0.75 (±0.21)/0.63 (±0.15) | 0.18 (±0.17)/0.08 (±0.10) | 0.92 (±0.46)/0.98 (±0.41) |

Table 17: RQ4 FDB results: pause handling and backchanneling. Entries show mean (± sample standard deviation) across four inference runs unless marked.

| Backbone | Setting | $s_c$ | Pause Synth. TOR ↓ | Pause Candor TOR ↓ | ICC TOR ↓ | ICC Frequency | ICC JSD ↓ |
|---|---|---|---|---|---|---|---|
| GLM | Real | 1 | 0.47 (±0.24) | 0.46 (±0.24) | 0.42 (±0.13) | 0.09 (±0.02) | 0.77 (±0.02) |
| GLM | 50% Synthetic | 1 | 0.75 (±0.13) | 0.56 (±0.16) | 0.60 (±0.10) | 0.11 (±0.01) | 0.72 (±0.02) |
| GLM | Synthetic | 1 | 0.85 (±0.07) | 0.70 (±0.06) | 0.68 (±0.13) | 0.14 (±0.01) | 0.70 (±0.01) |
| Kimi | Real | 1 | 0.20 (±0.15) | 0.39 (±0.19) | 0.31 (±0.23) | 0.05 (±0.02) | 0.84 (±0.07) |
| Kimi | 50% Synthetic | 1 | 0.65 (±0.25) | 0.53 (±0.09) | 0.22 (±0.06) | 0.06 (±0.02) | 0.81 (±0.04) |
| Kimi | Synthetic | 1 | 0.67 (±0.14) | 0.78 (±0.23) | 0.44 (±0.13) | 0.10 (±0.02) | 0.74 (±0.02) |

Table 18: RQ4 FDB results: smooth turn taking and user interruption. Entries show mean (± sample standard deviation) across four inference runs unless marked.

| Backbone | Setting | $s_c$ | Candor TT TOR ↑ | Candor TT Latency ↓ | Interrupt Pre-TOR ↓ | Interrupt Post-TOR ↑ | Interrupt Rating ↑ | Interrupt Latency ↓ |
|---|---|---|---|---|---|---|---|---|
| GLM | Real | 1 | 0.77 (±0.12) | 0.50 (±0.32) | 0.32 (±0.13) | 0.71 (±0.10) | 1.05 (±0.32) | 1.26 (±0.26) |
| GLM | 50% Synthetic | 1 | 0.84 (±0.08) | 0.30 (±0.34) | 0.29 (±0.09) | 0.67 (±0.07) | 1.90 (±0.39) | 0.87 (±0.12) |
| GLM | Synthetic | 1 | 0.81 (±0.06) | 0.09 (±0.10) | 0.19 (±0.09) | 0.52 (±0.14) | 2.49 (±0.23) | 0.51 (±0.03) |
| Kimi | Real | 1 | 0.91 (±0.05) | 0.26 (±0.13) | 0.18 (±0.04) | 0.71 (±0.07) | 1.51 (±0.32) | 1.38 (±0.18) |
| Kimi | 50% Synthetic | 1 | 0.82 (±0.05) | 0.10 (±0.02) | 0.09 (±0.03) | 0.16 (±0.03) | 1.81 (±0.54) | 1.00 (±1.00) |
| Kimi | Synthetic | 1 | 0.83 (±0.18) | 0.01 (±0.02) | 0.16 (±0.08) | 0.26 (±0.08) | 2.66 (±0.56) | 0.78 (±0.32) |