THINKING IN THE GAPS: REUSING THE TEXT–SPEECH FRAME-RATE GAP FOR CHAIN-OF-THOUGHT REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Spoken language models (SLMs) enable natural human-computer interaction, but most decode text and speech at different frame rates on a shared autoregressive grid, leaving many text positions idle while speech generation continues. We introduce **Thinking in the Gaps**, which reuses these idle positions for stepwise reasoning in parallel with speech generation. Because full reasoning traces often exceed the available gap capacity, a context-conditioned compressor constructs length-constrained reasoning targets that fit within these gaps. Across five spoken mathematical question-answering benchmarks, this method improves average accuracy over the matched baseline at every tested frame rate, with the largest gain of 14.05 percentage points at 25 Hz. Crucially, these gains come at virtually no additional inference cost and without any interruption to speech generation.[1]

1 INTRODUCTION

Spoken language models (SLMs) are emerging as a promising foundation for natural and expressive human-computer interaction (Cui et al., 2025). By directly modeling speech as both input and output, they enable richer interaction patterns than text-only systems, including natural turn-taking (OpenAI, 2024), prosody-aware responses (Wang et al., 2024; Xue et al., 2024), and end-to-end spoken dialogue (Zhang et al., 2023; Wang et al., 2023). Mainstream SLM architectures can be broadly categorized as either parallel or interleaved. Parallel architectures generate text and speech outputs concurrently (Défossez et al., 2024; Xie & Wu, 2024; Mitsui et al., 2024; Chen et al., 2025b; Xu et al., 2025a; Team, 2025b; Xiaomi, 2025; Team et al., 2025; Yu et al., 2025; KimiTeam et al., 2025; Zhao et al., 2025; Cui et al., 2026; Hu et al., 2026; Fang et al., 2025), whereas interleaved architectures place text and speech tokens in a single output stream (Team, 2026b; Zeng et al., 2024; Li et al., 2025c; Huang et al., 2025). Both types of architectures must simultaneously accommodate a speech-token frame rate and a text-token frame rate, a fundamental distinction from purely text-based LLM generation. Although SLMs vary in their specific speech-token frame rates, these rates are generally higher than text-token frame rates (Ye et al., 2026; Li et al., 2026a). Consequently, once text generation is complete, idle tokens must be appended to the text stream to maintain alignment while speech generation continues. Such idle-token padding, however, represents an evident waste of inference resources–computation that could otherwise be devoted to more productive purposes.

We exploit precisely this property: *the speech frame rate exceeds the text frame rate*. Instead of leaving the idle positions as a single trailing block, we redistribute them across the ends of text segments, thereby constructing gaps induced by the speech–text frame-rate difference. This differs from word-level speech–text alignment, which enforces tight synchronization between speech and text (Défossez et al., 2024) but can hinder generation when the text stream is semantically sparse (Liu et al., 2026b). Replacing the idle tokens in these gaps with semantically meaningful reasoning tokens enables stepwise in-gap reasoning while preserving the nominal text-stream length. Because each gap is much shorter than a typical reasoning trace, we introduce a context-conditioned compressor that retains the most important reasoning tokens within the available capacity, yielding what we term **Thinking in the Gaps (THINKGAP)**.

[1] We will release our dataset, code, and checkpoints.

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.

Following STiTCH (Chiang et al., 2025), we evaluate THINKGAP on spoken mathematical question answering benchmarks. This setting places substantial demands on reasoning: purely verbalized mathematical expressions (e.g., "x plus five y equals zero") can be harder to interpret than symbolic counterparts (Hyeon et al., 2025), and SLMs remain weaker than text-based LLMs on such tasks (Lin et al., 2025b; Wei et al., 2025). In the main experiments with the 1.7B model, THINKGAP consistently improves average answer accuracy over the standard parallel-generation baseline at the matched speech-token frame rate, across all five frame-rate settings considered (25, 12.5, 8.33, 6.25, and 5 Hz), yielding an average gain of $+6.03$ percentage points. The largest improvement, $+14.05$ points, occurs at 25 Hz, corresponding to the largest idle budget, although gains at the remaining, lower frame rates are not strictly monotonic. Moreover, THINKGAP keeps average SLM forward steps, first-packet latency, and end-to-end latency comparable to the baseline within a reasonable margin, while leaving speech generation uninterrupted by design. We also find that the measured UTMOS (a metric for speech quality and naturalness) is similar on average, while average word error rate (WER) increases from 0.096 to 0.117.

Our contributions are summarized as follows:

- We introduce **Thinking in the Gaps**, a new interaction paradigm for spoken language models that reuses frame-rate-induced idle text positions for stepwise reasoning.
- We propose a context-conditioned reasoning-compression method for this gap-constrained setting, along with an automatically generated stepwise-reasoning dataset.
- We demonstrate the effectiveness and efficiency of this paradigm across five speech frame rates on five spoken mathematical question-answering benchmarks.

2 RELATED WORK

2.1 SPOKEN LANGUAGE MODEL REASONING

Prior works on SLM reasoning generally follow two directions: *(i)* improving the reasoning capability of SLMs on speech and audio tasks, and *(ii)* improving the engineering realization of reasoning under real-time spoken interaction.

The first direction focuses on mitigating the gap between waveform signals and the language semantics used for reasoning (Wang et al., 2025). For understanding and translation, CoT-ST (Du et al., 2024) and the approach of Xie et al. (Xie et al., 2025a) combine acoustic processing with downstream semantic tasks, while Audio-CoT (Ma et al., 2025) and Audio-Reasoner (Xie et al., 2025b) introduce structured, multi-stage reasoning for general audio understanding and long-horizon tasks. In generative settings, SpatialSonic (Sun et al., 2025) and SpeechGPT-Gen (Zhang et al., 2024) use stepwise attribute extraction and information chaining to progressively align acoustic features with high-level semantic planning. TARS (Wang et al., 2026) proposes systematic alignment strategies to harmonize acoustic perception with linguistic reasoning in SLMs.

The second direction focuses on making reasoning compatible with low-latency spoken interaction. Recent studies explore simultaneous or incremental processing, where models reason during streaming input rather than after the full utterance is observed (Shih et al., 2025). Representative examples include chunked or simultaneous reasoning frameworks such as STiTCH (Chiang et al., 2025), SHANKS (Chiang et al., 2026), Dual-Reasoner (Li et al., 2026b), and Mini-Omni-Reasoner (Xie et al., 2025c). Another approach adopts a dual-brain model architecture that enables the model to think while speaking, allowing reasoning and speech generation to proceed concurrently (Wu et al., 2025; Lab, 2026; Lin et al., 2026; Liang et al., 2026). Additional work addresses reasoning for cascaded dialogue systems, including input-time speculation (Li & Grover, 2025), listen-think-speak frameworks (Zou et al., 2026), and dual-track streaming response architectures (Liu et al., 2026a).

However, previous think-while-speaking paradigms typically allocate explicit positions for reasoning. We instead formulate think-while-speaking for SLMs by *reusing text idle positions already present during parallel speech–text generation*. Our method combines semantically meaningful segmentation with context-conditioned compression, placing reasoning within the existing gap budget rather than appending a full reasoning trace.

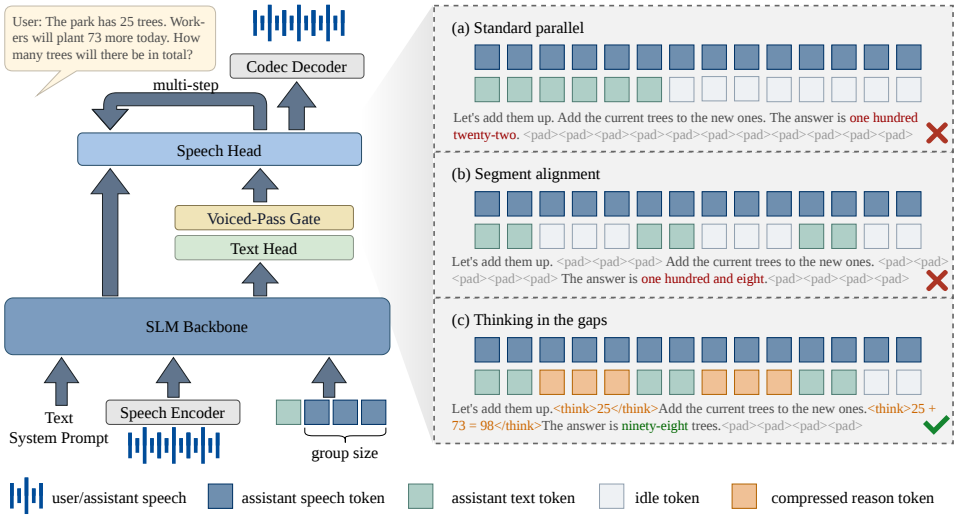

Figure 1: Overview of (a) the **Standard parallel** framework, (b) the **Segment alignment** framework, and (c) the **Thinking in the gaps** framework, all built on a parallel speech–text SLM architecture. The example shows the responses of the three frameworks to the same question.

2.2 Efficient Reasoning

Recent works on efficient reasoning aim to preserve the benefits of Chain-of-Thought (CoT) prompting while reducing the latency and computational cost of long reasoning traces (Sui et al., 2025). Existing approaches broadly fall into two categories: *(i)* efficient explicit CoT, which shortens or compresses the explicit reasoning trace, and *(ii)* implicit CoT, which internalizes reasoning without generating intermediate tokens.

Efficient explicit CoT methods reduce reasoning cost while retaining a natural-language trace. These methods include replacing full CoT with short, draft-style reasoning (Xu et al., 2025b) and allocating token budgets adaptively according to problem complexity (Lin et al., 2025a). TokenSkip (Xia et al., 2025) enables the model to omit less-informative reasoning tokens. Most closely related to our work, ECoM (Feng et al., 2026) brings efficient CoT to SLMs that generate text and speech serially within a single interleaved autoregressive stream.

Implicit CoT methods instead distill reasoning into the model hidden states (Chen et al., 2025c). Early work distills explicit CoT into implicit reasoning (Deng et al., 2023), progressively internalizes CoT supervision (Deng et al., 2024; Shen et al., 2025), or treats implicit reasoning as an alternative to verbose rationale generation (Li et al., 2025b). More recent methods compress reasoning adaptively through curriculum learning under progressively tighter token budgets (Hammoud et al., 2025); token-level skipping, dynamic stopping, and self-distillation (Chen et al., 2025a; Sang et al., 2026; Li et al., 2025d); or continuous latent representations, either training-free (Zhang et al., 2025; Zhuang et al., 2025) or beyond next-token prediction (Tan et al., 2025b).

Most of these methods target text-only LLMs. ECoM addresses an *interleaved* SLM, but it does not exploit the text–speech frame-rate mismatch of *parallel-generation* SLMs, where the available reasoning space is determined by the pace of speech generation rather than by a fixed token budget.

3 Method

3.1 Background

Single-turn dialogue. For clarity of exposition, we describe our method for a single-turn, multimodal dialogue; the same procedure can be extended to multi-turn dialogue by additionally conditioning on the dialogue history. Let $\mathcal{V}$ denote the text vocabulary, and $\mathcal{V}_{\mathrm{r}} \subseteq \mathcal{V}$ denote the subset

used for reasoning tokens. Let $\mathcal{A} = \prod_{c=1}^{d} \mathcal{A}_c$ be the alphabet of one speech frame, represented as a d-codebook residual/semantic token tuple. A system turn consists of

- a *context* C, formed by the system prompt and the user question (given as speech or text);
- a *content text* sequence $w_{1:L} \in \mathcal{V}^L$ (the semantically meaningful tokens);
- a *speech* sequence $y_{1:N} \in \mathcal{A}^N$ realizing the same content.

Frame-rate mismatch and idle budget. Let f_{a} be the speech token rate (Hz) and f_{w} the effective text token rate of the same utterance. The *frame-rate ratio* is defined as follows:

$$\rho \;:=\; \frac{f_{\mathrm{a}}}{f_{\mathrm{w}}} \;>\; 1. \tag{1}$$

For a turn of duration Δ, let $N = \lceil f_{\mathrm{a}}\Delta \rceil$ and $L = \lceil f_{\mathrm{w}}\Delta \rceil$. When $N > L$, the text stream has

$$B \;:=\; N - L \;>0 \tag{2}$$

free text slots[2]. In the standard parallel generation scheme (Team et al., 2025), these slots are filled with $\langle \mathrm{pad} \rangle$ tokens and do not contribute meaningful text computation. We refer to B as the *idle budget*.

Baseline parallel generation. As shown in Fig. 1(a), the baseline parallel speech–text model places all idle tokens at the tail:

$$x_{1:N}^{\mathrm{base}} \;=\; \big(w_1, \ldots, w_L, \underbrace{\langle \mathrm{pad} \rangle, \ldots, \langle \mathrm{pad} \rangle}_{B}\big), \tag{3}$$

At each slot, a shared SLM backbone produces a hidden state

$$h_n \;:=\; p_\theta^b(x_{\le n},\, y_{\le n},\, C)\,, \quad h_0 = p_\theta^b(C) \tag{4}$$

which is then used by the text and speech heads:

$$p_\theta^{\mathrm{base}}(x_{1:N}, y_{1:N}) = \prod_{n=1}^{N} \underbrace{p_\theta^x(x_n \mid h_{n-1})}_{\text{text head}} \cdot \underbrace{p_\theta^y(y_n \mid h_{n-1}, x_n)}_{\text{speech head}}. \tag{5}$$

This factorization exposes two limitations: **(D1) wasted budget**: the B tail slots approach zero conditional entropy, $p_\theta^x(\langle \mathrm{pad} \rangle \mid h_{n-1}) \to 1$ for $n > L$, so these text-chain positions carry no semantic content; and **(D2) shared conditioning**: text tokens not intended to be spoken, such as $\langle \mathrm{pad} \rangle$, can directly condition the speech head p_θ^y and perturb the synthesized speech.

3.2 Segment Alignment

Amortizing the idle budget. As shown in Fig. 1(b), rather than placing all B free slots at the tail, we distribute the same budget across the response. Let $0 = l_0 < l_1 < \cdots < l_J = L$ define J consecutive content blocks:

$$w^{(j)} = w_{l_{j-1}+1:l_j}, \qquad m_j := |w^{(j)}| = l_j - l_{j-1}. \tag{6}$$

A gap schedule $\boldsymbol{g} = (g_1, \ldots, g_J)$ satisfies $\sum_{j=1}^{J} g_j = B$, and inserts g_j free slots after the j-th content block:

$$x_{1:N}^{\boldsymbol{g}} \;=\; \big(w^{(1)}, \langle \mathrm{pad} \rangle^{g_1}, w^{(2)}, \langle \mathrm{pad} \rangle^{g_2}, \ldots, w^{(J)}, \langle \mathrm{pad} \rangle^{g_J}\big). \tag{7}$$

Thus, the total number of positions is unchanged; we only redistribute the existing free computation budget from the tail to the interior of the sequence.

To enable SLMs to acquire this generation paradigm at inference time, we construct dedicated training data for segment alignment and apply supervised fine-tuning (SFT). Concretely, each block is instantiated via step decomposition performed by an external LLM, which is prompted to produce

[2]Before integer rounding, the nominal surplus is $(f_{\mathrm{a}} - f_{\mathrm{w}})\Delta = (\rho - 1) f_{\mathrm{w}}\Delta$.

the response as J semantically coherent steps. We denote by c_j the number of grouped speech-decoder slots aligned with the j-th step, obtained by converting the segment's raw acoustic-token count into decoder slots using the speech group size. The nominal local speech–text surplus is

$$d_j = c_j - m_j, \tag{8}$$

which defines the final nonnegative gap allocation. We do not enforce $g_j = d_j$ locally during data construction; instead, the signed value d_j enters the carry-over recursion described in Appendix E.6, and the final sequence length is determined globally. This construction introduces no additional nominal positions relative to standard training data, and Appendix C confirms empirically that SLMs learn this paradigm without loss in performance.

Non-spoken tokens must be invisible at the current slot. The slots created by the dilation (Equation (7)) do not carry spoken content: each slot holds either the reserved symbol $\langle\mathrm{pad}\rangle$ or, after the subsequent reasoning insertion, a token for reasoning. Because these symbols are not realized acoustically, they should not directly condition the speech head at the corresponding slot.

We design an explicit voiced-pass gating mechanism to address Limitation **(D2)**:

$$\sigma_n := \begin{cases} 0, & \text{if slot } n \text{ is padding or lies in a } \langle\mathrm{think}\rangle \ldots \langle/\mathrm{think}\rangle \text{ span,} \\ 1, & \text{if slot } n \text{ contains spoken content.} \end{cases} \tag{9}$$

The text head still models the full text stream, whereas the current text embedding supplied to the speech head is zeroed when $\sigma_n = 0$. Earlier non-spoken positions may still influence the shared history h_{n-1}; the gate removes only the direct current-slot conditioning path.

3.3 Thinking in the Gaps

Turning idle positions into reasoning. As shown in Fig. 1(c), for each non-final block $j < J$ we place a retained reasoning sequence $z^{(j)} = (z_1^{(j)}, \ldots, z_{k_j}^{(j)})$, $z_i^{(j)} \in \mathcal{V}_\mathrm{r}$, inside the available gap capacity. Let $q^{(j)} = (\langle\mathrm{think}\rangle, z^{(j)}, \langle/\mathrm{think}\rangle)$ when a reasoning block is inserted and let $q^{(j)}$ be empty otherwise. Since the response is already complete after the final content block, no reasoning block is inserted following it, so $q^{(J)} = \langle\mathrm{pad}\rangle^{(J)}$. Let $p_j \geq 0$ be the residual padding after block j. The budget recursion allocates reasoning and padding so that $\sum_{j=1}^{J}(|q^{(j)}| + p_j) = B$, yielding

$$\tilde{x}_{1:N} = \big(w^{(1)}, q^{(1)}, \langle\mathrm{pad}\rangle^{p_1}, \ldots, w^{(J)}, \langle\mathrm{pad}\rangle^{p_J}\big) \in \mathcal{V}^N. \tag{10}$$

The construction preserves the order of all content tokens and accounts explicitly for reasoning markers and residual padding. Since $\mathcal{V}_\mathrm{r} \subseteq \mathcal{V}$, reasoning requires no new vocabulary, embedding table, or architectural module; *it only changes how existing free slots are used*. Models with a larger speech-to-text token-rate ratio therefore provide more potential positions for reasoning. We do not impose a fixed reasoning length across models: the available budget is determined by the realized gaps, and at inference time by whether synthesis of the current sentence has completed. In this way, the SLM can use non-content positions to make additional reasoning information available when predicting the next content block, effectively addressing Limitation **(D1)**.

Context-conditioned reasoning compression. The main challenge in this paradigm is that a gap is usually much shorter than the reasoning trace that we would like to insert into it. Let $r^{(j)} = (r_1^{(j)}, \ldots, r_{R_j}^{(j)})$ be the full reasoning trace for content block $w^{(j+1)}$ ($j < J$), where $R_j = |r^{(j)}|$ can be substantially larger than the realizable reasoning budget b_j. Note that b_j denotes the realizable reasoning-content budget in target-SLM tokens after accounting for the two boundary markers and carry-over from earlier segments (detailed in Appendix E.6). Rather than generating a new short reasoning trace, we perform *extractive compression*: we retain a subset of the original reasoning content and keep its original order.

The compression is conditioned on the dialogue context available before the gap. Specifically, let

$$\mathcal{C}_j = \big(C,\, w^{(1:j)},\, z^{(1:j-1)}\big) \tag{11}$$

denote the dialogue context together with preceding answer blocks and previously retained reason tokens. Since our objective is to preserve the textual reasoning information that is useful for generating the subsequent *textual* answer, we condition the compressor only on textual context and do not

condition it on speech tokens. We choose LLMLingua-2 (Pan et al., 2024) as the compressor, which formulates compression as binary token classification and uses the predicted preserve probability as the token importance score. We adopt this module because its Transformer encoder leverages bidirectional context—rather than the left-to-right perplexity used by earlier entropy-based methods—to better exploit the *full* reasoning trace for estimating token importance. We feed the concatenation of $\mathcal{C}_j$ with the full reasoning trace $r^{(j)}$ into the compressor to obtain a token-level preservation score

$$s_i^{(j)} = p_\phi\Big(\mathcal{C}_j, r^{(j)}\Big)_i . \tag{12}$$

If the complete trace costs at most b_j tokens, we retain it and carry the remaining capacity forward. Otherwise, the selection strategy aggregates subword preservation probabilities into a mean score $\bar{s}_i^{(j)}$ for each source word $u_i^{(j)}$ and measures its cost $\ell_i^{(j)}$ with the target SLM tokenizer. Words are considered in the order of decreasing scores and retained whenever they fit the remaining budget; the selected words are then restored to the source order in the full reasoning trace:

$$\mathcal{I}_j = \mathrm{GreedyBudget}\Big(\{(\bar{s}_i^{(j)}, \ell_i^{(j)})\}_{i=1}^{M_j}, b_j\Big), \qquad z^{(j)} = \mathrm{Tok}\Big(\big(u_i^{(j)}\big)_{i\in\mathcal{I}_j \text{ in source order}}\Big), \tag{13}$$

where $\sum_{i\in\mathcal{I}_j} \ell_i^{(j)} \le b_j$. The effective compression ratio is therefore determined by the realized gap capacity rather than by a fixed ratio. Because compression functions as an *offline* data augmentation technique and the model implicitly learns this compression strategy during training, it can directly generate compressed reasoning during inference without requiring a compressor.

4 Experiments

4.1 Experimental Setup

Training Setting. We use two categories of training data: **(i) general dialogue data**, including *MagpiePro* (Xu et al., 2025c) and *InfGen* (Li et al., 2025a), with speech synthesized by CosyVoice3 (Du et al., 2025), following DrVoice (Tan et al., 2025a), used during pre-training to develop the model's spoken dialogue ability. **(ii) mathematical reasoning data**, including the training split of GSM8K (Cobbe et al., 2021), NuminaMath (Li et al., 2024), and MathQA (Amini et al., 2019), used during post-training to enhance the model's spoken mathematical question-answering capability and to construct data with a stepwise reasoning structure [3].

Our training framework is built upon FunAudioChat (Team et al., 2025), a high-performing speech-language model that supports parallel generation of speech and text and releases the complete model architecture and training code. To ensure a fair comparison, all models in the main experiments share the same training data, training configurations, and model initialization, with Whisper (Radford et al., 2023) as the audio encoder, the language model split of Qwen3-VL-1.7B (Team, 2025c) as the SLM backbone [4], and CosyVoice3 (Du et al., 2025) as the speech tokenizer.

Evaluation Setting. We evaluate our method on five spoken mathematical question-answering benchmarks: AddSub (Hosseini et al., 2014), SingleEq (Koncel-Kedziorski et al., 2015), MultiArith (Roy & Roth, 2015), SVAMP (Patel et al., 2021), and GSM8K (Cobbe et al., 2021). The questions and answers in these datasets are of moderate difficulty and can be naturally expressed in speech, making them suitable for evaluating SLMs of moderate scale. For spoken response evaluation, we use the same complete evaluation set as STiTCH (Chiang et al., 2025), including the synthesized speech items. All experiments are conducted under the URO-Bench (Yan et al., 2025) evaluation framework: the generated speech is transcribed using Whisper-large-v3 (Radford et al., 2023), and the resulting transcript is scored against the ground-truth answer by an LLM-as-judge[5], conditioned on the corresponding question and a dedicated evaluation prompt.

[3] The data-construction pipeline is described in Appendix E.

[4] We choose Qwen3-VL-1.7B because our empirical results indicate that it achieves better performance than Qwen3-1.7B-Base (Team, 2025a) when fine-tuned to incorporate speech inputs.

[5] We employ Qwen-flash-2025-07-28 as the judge model, which has been shown to exhibit high agreement with human judgments (Team, 2026a). For each evaluation instance, the judge is invoked three times and the resulting scores are averaged to reduce variance.

Table 1: **Main results on five spoken math QA benchmarks across speech frame rates f_a.** *Baseline* keeps the idle tokens in a single trailing block; *Alignment* redistributes the same padding budget across the ends of text segments (re-layout only, with no new content); THINKGAP (ours) replaces the redistributed idle tokens with context-conditioned compressed reasoning tokens. $\bar{B}$ is the average number of idle slots allocated per 3.3 text tokens. Δ reports the change in average accuracy w.r.t. the baseline at the same frame rate. The best result within each block is in **bold**.

f_a	$\bar{B}$	Method	AddSub	MultiArith	SingleEq	SVAMP	GSM8K	Avg.	Δ
25.0 Hz	21.7	Baseline	59.32	74.32	59.32	52.73	27.70	54.68	–
		Alignment	60.55	69.92	62.69	52.84	27.91	54.78	↑0.10
		THINKGAP	**66.66**	**88.12**	**81.34**	**64.10**	**43.43**	**68.73**	↑14.05
12.5 Hz	9.2	Baseline	60.85	75.28	62.38	54.18	29.05	56.35	–
		Alignment	62.69	74.13	62.07	**55.62**	27.83	56.47	↑0.12
		THINKGAP	**69.11**	**81.03**	**69.41**	54.73	**36.56**	**62.17**	↑5.82
8.33 Hz	5.0	Baseline	**66.36**	70.11	64.52	**54.73**	27.78	56.70	–
		Alignment	57.49	73.56	63.30	52.61	26.59	54.71	↓1.99
		THINKGAP	62.99	**76.62**	**76.14**	53.84	**34.04**	**60.73**	↑4.03
6.25 Hz	3.0	Baseline	**65.74**	74.13	66.66	53.73	28.9	57.83	–
		Alignment	60.55	69.34	64.22	**54.73**	26.33	55.03	↓2.80
		THINKGAP	65.44	**76.43**	**70.64**	53.51	**31.01**	**59.41**	↑1.57
5.00 Hz	1.7	Baseline	**65.13**	78.35	65.44	49.94	26.41	57.05	–
		Alignment	62.99	71.07	70.03	**53.73**	31.70	57.90	↑0.85
		THINKGAP	63.60	**81.60**	**77.37**	53.62	**32.36**	**61.71**	↑4.66

4.2 MAIN RESULTS

Thinking in the gaps reuses existing decoding slots. Across the five frame rates in Table 1, turning frame-rate–induced idle tokens into compressed reasoning tokens improves average accuracy by $+14.05$ (25.0 Hz), $+5.82$ (12.5 Hz), $+4.03$ (8.33 Hz), $+1.57$ (6.25 Hz), and $+4.66$ (5.00 Hz) points, averaging $+6.03$ points. THINKGAP places reasoning within positions allocated by the existing gap schedule rather than appending a full reasoning trace. These results support viewing the speech–text frame-rate mismatch as a reusable reasoning resource.

The largest gain occurs with the largest gap budget. $\bar{B}$ denotes the average number of idle slots allocated per 3.3 text tokens. At 25 Hz, THINKGAP yields its largest average gain and improves all five benchmarks. The gains are smaller at the four lower frame rates, but they are not monotonic in $\bar{B}$. The results therefore suggest that available idle capacity influences the benefit of in-gap reasoning, without establishing a strict scaling law.

Alignment-only redistribution has a small aggregate effect in the main setting. The alignment control redistributes the *same* number of idle tokens across text segments without inserting new content. Averaged over the five main frame-rate settings, its accuracy differs from the baseline by -0.74 points, although the differences vary across individual frame rates. Without uncertainty estimates, we do not interpret this aggregate difference as statistical equivalence. The larger improvements from compressed reasoning tokens—up to $+14.05$ points—indicate that the content placed in the gaps matters beyond redistribution alone (Goyal et al., 2024; Pfau et al., 2024).

4.3 SCALING UP AND APPLYING TO OTHER ARCHITECTURES

Scaling to Larger Models. To examine scaling behavior, we evaluate the FunAudioChat architecture at both 25 Hz and 5 Hz with an 8B backbone. For the 25 Hz model, we initialize from the publicly released 5 Hz FunAudioChat checkpoint (group size $= 5$), further train it on the general dialogue dataset, and then perform SFT on the spoken mathematical QA dataset; the 5 Hz model is directly fine-tuned on the same dataset. As Table 2 shows, THINKGAP improves average accuracy by $+3.37$ points at 25 Hz but changes it by -0.05 points at 5 Hz. The smaller effect observed relative to the 1.7B setting may be attributed to a stronger underlying model (the 8B model's baseline

Table 2: **Scaling up and applying to other architectures: results on five spoken math QA benchmarks.** *Baseline* keeps the idle tokens in a single trailing block; *Alignment* redistributes the same idle budget across the ends of text segments; THINKGAP (ours) replaces the redistributed idle tokens with context-conditioned compressed reasoning tokens. Gray rows are reported for reference only, as their training data and training procedures differ from our experimental setting.

Architecture	Method	AddSub	MultiArith	SingleEq	SVAMP	GSM8K	Avg.	Δ
FunAudioChat 8B 25Hz	Baseline	**77.06**	88.50	88.68	71.68	54.30	76.04	–
	Alignment	75.53	**88.88**	84.70	66.10	53.84	73.81	↓2.23
	THINKGAP	76.14	**88.88**	**90.51**	**73.69**	**67.83**	**79.41**	↑3.37
FunAudioChat 8B 5Hz	FunAudioChat	78.89	90.80	81.34	85.84	83.07	83.99	–
	Baseline	77.37	**90.22**	83.79	68.00	**59.41**	**75.76**	–
	Alignment	73.39	**90.22**	84.40	64.10	52.92	73.01	↓2.75
	THINKGAP	**81.03**	87.93	**84.70**	**69.78**	55.11	75.71	↓0.05
Mini-Omni 1.5B 25Hz	Baseline	9.17	24.13	22.62	11.59	12.31	15.96	–
	Alignment	22.32	33.14	37.30	17.83	14.45	25.01	↑9.04
	THINKGAP	**40.36**	**48.85**	**48.62**	**26.53**	**20.58**	**36.99**	↑21.02
Qwen2.5-Omni 3B 25Hz	Qwen2.5-Omni	60.55	88.69	80.12	74.13	49.89	70.68	–
	Baseline	66.36	82.95	68.80	58.97	34.73	62.36	–
	Alignment	12.84	3.44	17.43	6.02	0.20	7.99	↓54.38
	THINKGAP	**73.08**	**88.88**	**88.37**	**61.20**	**56.53**	**73.61**	↑11.25

accuracy is substantially higher than that of the 1.7B), limited training data for the larger model (our dataset comprises only 1.9k hours of speech, considerably less than the ∼300k hours of SFT data typically used to train 8B-scale models (KimiTeam et al., 2025)), or a combination of both factors.

Transfer to Other Model Architectures. We also re-implement THINKGAP on two other speech–text parallel-generation SLM designs: (i) a 1.5B model following Mini-Omni, using CosyVoice 3 speech tokens at 25 Hz and no separate talker module; and (ii) Qwen2.5-Omni-3B, whose original talker codec is not publicly available, with CosyVoice 3 adopted as the talker at 25 Hz and speech tokens not fed back into the SLM backbone. Table 2 shows improvements of $+21.02$ and $+11.25$ points over the corresponding modified baselines, respectively. These results establish transfer to the two tested designs rather than architecture-independent effectiveness. The alignment-only intervention proves sensitive to architecture. In the Mini-Omni-derived model, the backbone is sensitive to the relative distance between speech and text tokens, making explicit alignment beneficial. However, in Qwen2.5-Omni, the backbone (thinker) has no direct signal indicating when speech generation has terminated, as speech tokens are not autoregressively conditioned back into the backbone's context. As a result, the thinker tends to produce excessive idle tokens that interfere with subsequent generation. THINKGAP addresses this problem by replacing idle tokens with semantically meaningful ones, allowing the thinker to better estimate when to initiate the generation of the next utterance—thus circumventing this issue and preserving its effectiveness.

4.4 ABLATION ON REASONING COMPRESSION

Table 3 evaluates reasoning compression and context conditioning. Compared with *w/o Compressor*, *Compressor w/ Context* achieves higher accuracy at all five speech frame rates, indicating that importance-based selection preserves more task-relevant information than random token removal under the tested budgets. Compared with *Compressor w/o Context*, adding context improves accuracy in four of five settings; at 8.3 Hz, the context-free variant is higher (61.41 versus 60.73). Relative to the uncompressed *Full* reference, compressed variants use substantially fewer average SLM forward steps. Compared with the baseline, however, THINKGAP uses slightly more steps at 25, 12.5, and 8.3 Hz and slightly fewer at 6.25 and 5 Hz, which can be considered comparable within a reasonable margin of error, while still representing a substantial reduction relative to uncompressed reasoning. To further examine the outcome of reasoning compression, we conduct a small-scale manual sampling analysis of the retained reasoning tokens (see Appendix F) and find them to be consistent with our intended *extractive-compression, scratchpad-style* computation pattern (i.e., retaining only the key variables, numbers, and formulas needed for the computation).

Table 3: **Ablation on reasoning compression.** We report average accuracy and average SLM forward steps under different speech frame rates. *Full* does not compress the reasoning trace and inserts idle tokens between speech segments to provide the corresponding gap budget. *w/o Compressor* randomly discards reasoning tokens to match the compressed budget, while *Compressor w/o Context* removes context conditioning from the compressor. *Compressor w/ Context* denotes our full THINKGAP setting.

	25.0 Hz		12.5 Hz		8.33 Hz		6.25 Hz		5.00 Hz	
Setting	**Acc.**	**Steps**	**Acc.**	**Steps**	**Acc.**	**Steps**	**Acc.**	**Steps**	**Acc.**	**Steps**
Full	69.33	297.3	65.57	249.8	63.61	245.8	64.18	127.8	65.77	161.4
Baseline	54.68	228.5	56.35	116.5	56.70	79.7	57.83	61.8	57.05	49.8
Compressor w/ Context	**68.73**	235.7	**62.17**	119.5	60.73	81.4	**59.41**	61.2	**61.71**	48.8
Compressor w/o Context	66.83	243.8	61.50	120.9	**61.41**	81.5	56.96	61.9	60.32	48.9
w/o Compressor	68.08	242.3	60.92	129.0	54.88	86.6	54.58	65.6	57.72	49.1

Table 4: **Comparison of FST and E2E latency across speech frame rates.** We report the average First-Speech-Token latency and End-to-End latency (in seconds) of three settings: *Baseline*, *Alignment*, and THINKGAP. Both metrics are lower-is-better (↓).

	25.0 Hz		12.5 Hz		8.33 Hz		6.25 Hz		5.00 Hz		Avg	
Setting	**FST↓**	**E2E↓**	**FST↓**	**E2E↓**	**FST↓**	**E2E↓**	**FST↓**	**E2E↓**	**FST↓**	**E2E↓**	**FST↓**	**E2E↓**
Baseline	0.1344	10.09	0.1783	8.17	0.2245	7.55	0.2677	7.54	0.2540	6.99	0.2118	8.07
Alignment	**0.1168**	**9.21**	**0.1570**	**7.48**	**0.1974**	8.52	0.2418	6.89	**0.2428**	7.26	**0.1912**	7.87
THINKGAP	0.1313	9.85	0.1729	8.65	0.2162	**7.41**	**0.2388**	**6.72**	0.2482	**5.54**	0.2015	**7.63**

4.5 LATENCY CALCULATION

Does thinking in the gaps compromise real-time responsiveness? To answer this question, we measure two latency metrics for each setting: *First-Speech-Token (FST)* latency, i.e., the time elapsed from the end of the user utterance to the emission of the first speech token, corresponding directly to the first-packet delay perceived by the listener; and *End-to-End (E2E)* latency, i.e., the total time required to generate the complete speech response. Both metrics are reported in seconds, and all timing experiments are conducted on a single **PPU-ZW810** under identical decoding configurations.

As shown in Table 4, THINKGAP achieves latency on par with the *Baseline* in both first-packet latency (0.2015 vs. 0.2118 s) and end-to-end latency (7.63 vs. 8.07 s). These results confirm that **THINKGAP introduces nearly no additional first-packet latency and no extra inference cost.** Decoding steps remain unchanged in theory, as thinking simply occupies slots the standard SLM would otherwise fill with silence, while delivering better arithmetic accuracy.

We discuss the limitations of our approach and experiments in detail in Appendix A.

5 CONCLUSION

We identify an under-explored inefficiency in spoken language models (SLMs): the mismatch between speech and text token frame rates leaves idle text positions while speech generation continues. **Our approach, Thinking in the Gaps, reclaims these wasted positions as a bounded resource for reasoning.** The method redistributes idle positions across inter-segment gaps and fills them with context-conditioned compressed reasoning tokens. In the main experiments, it improves average accuracy over the matched baseline at all five tested frame rates, with a mean gain of $+6.03$ percentage points and a maximum of $+14.05$ points at 25 Hz. The effect transfers to the two tested alternative parallel-generation designs, although the 8B FunAudioChat experiment improves at 25 Hz but not at 5 Hz. The method uses the existing gap schedule rather than appending a full reasoning trace; measured forward-step counts remain close to the baseline. These results support further study of structural idle capacity as a reasoning resource while motivating broader task, architecture, latency, and human speech-quality evaluation.

AI USE STATEMENT

Generative AI tools were used to construct and curate synthetic training data, as documented in Appendix E. Specifically, large language models (LLMs) were used to decompose the "solutions" annotated in the original dataset into step-by-step procedures, generate spoken-style paraphrases and opening utterances, and assist with data cleaning and reordering. Generative text-to-speech (TTS) systems were used to synthesize the corresponding speech data. AI-based tools were also employed for language polishing, proofreading, and consistency checks between narrative claims and the authors' reported tables, figures, and methods.

These tools were not used to design experiments, develop models, or generate, modify, or select experimental results. The authors reviewed the outputs of all AI-assisted processes and take full responsibility for the technical contributions, interpretations, conclusions, and final content of this work.

REFERENCES

Aida Amini, Saadia Gabriel, Shanchuan Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. Mathqa: Towards interpretable math word problem solving with operation-based formalisms. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pp. 2357–2367. Association for Computational Linguistics, 2019. doi: 10.18653/V1/N19-1245. URL https://doi.org/10.18653/v1/n19-1245.

Qiguang Chen, Dengyun Peng, Jinhao Liu, Huikang Su, Jiannan Guan, Libo Qin, and Wanxiang Che. Aware first, think less: Dynamic boundary self-awareness drives extreme reasoning efficiency in large language models. *CoRR*, abs/2508.11582, 2025a. doi: 10.48550/ARXIV.2508.11582. URL https://doi.org/10.48550/arXiv.2508.11582.

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. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, volume ACL 2025 of *Findings of ACL*, pp. 2262–2282. Association for Computational Linguistics, 2025b. doi: 10.18653/V1/2025.FINDINGS-ACL.115. URL https://doi.org/10.18653/v1/2025.findings-acl.115.

Xinghao Chen, Anhao Zhao, Heming Xia, Xuan Lu, Hanlin Wang, Yanjun Chen, Wei Zhang, Jian Wang, Wenjie Li, and Xiaoyu Shen. Reasoning beyond language: A comprehensive survey on latent chain-of-thought reasoning. *CoRR*, abs/2505.16782, 2025c. doi: 10.48550/ARXIV.2505.16782. URL https://doi.org/10.48550/arXiv.2505.16782.

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. *CoRR*, abs/2507.15375, 2025. doi: 10.48550/ARXIV.2507.15375. URL https://doi.org/10.48550/arXiv.2507.15375.

Cheng-Han Chiang, Xiaofei Wang, Linjie Li, Chung-Ching Lin, Kevin Lin, Shujie Liu, Zhendong Wang, Zhengyuan Yang, Hung-yi Lee, and Lijuan Wang. Shanks: Simultaneous hearing and thinking for spoken language models. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2026, San Diego, California, United States, July 2-7, 2026*, pp. 8951–8972. Association for Computational Linguistics, 2026. doi: 10.18653/V1/2026.ACL-LONG.404. URL https://doi.org/10.18653/v1/2026.acl-long.404.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John

Schulman. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168, 2021. URL https://arxiv.org/abs/2110.14168.

Junbo Cui, Bokai Xu, Chongyi Wang, Tianyu Yu, Weiyue Sun, Yingjing Xu, Tianran Wang, Zhihui He, Wenshuo Ma, Tianchi Cai, Jiancheng Gui, Luoyuan Zhang, Xian Sun, Fuwei Huang, Moye Chen, Zhuo Lin, Hanyu Liu, Qingxin Gui, Qingzhe Han, Yuyang Wen, Huiping Liu, Rongkang Wang, Yaqi Zhang, Hongliang Wei, Chi Chen, You Li, Kechen Fang, Jie Zhou, Yuxuan Li, Guoyang Zeng, Chaojun Xiao, Yankai Lin, Xu Han, Maosong Sun, Zhiyuan Liu, and Yuan Yao. Minicpm-o 4.5: Towards real-time full-duplex omni-modal interaction. *CoRR*, abs/2604.27393, 2026. doi: 10.48550/ARXIV.2604.27393. URL https://doi.org/10.48550/arXiv.2604.27393.

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 Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pp. 13943–13970. Association for Computational Linguistics, 2025. doi: 10.18653/V1/2025.ACL-LONG.682. URL https://doi.org/10.18653/v1/2025.acl-long.682.

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. *CoRR*, abs/2410.00037, 2024. doi: 10.48550/ARXIV.2410.00037. URL https://doi.org/10.48550/arXiv.2410.00037.

Yuntian Deng, Kiran Prasad, Roland Fernandez, Paul Smolensky, Vishrav Chaudhary, and Stuart M. Shieber. Implicit chain of thought reasoning via knowledge distillation. *CoRR*, abs/2311.01460, 2023. doi: 10.48550/ARXIV.2311.01460. URL https://doi.org/10.48550/arXiv.2311.01460.

Yuntian Deng, Yejin Choi, and Stuart M. Shieber. From explicit cot to implicit cot: Learning to internalize cot step by step. *CoRR*, abs/2405.14838, 2024. doi: 10.48550/ARXIV.2405.14838. URL https://doi.org/10.48550/arXiv.2405.14838.

Yexing Du, Ziyang Ma, Yifan Yang, Keqi Deng, Xie Chen, Bo Yang, Yang Xiang, Ming Liu, and Bing Qin. Cot-st: Enhancing llm-based speech translation with multimodal chain-of-thought. *CoRR*, abs/2409.19510, 2024. doi: 10.48550/ARXIV.2409.19510. URL https://doi.org/10.48550/arXiv.2409.19510.

Zhihao Du, Changfeng Gao, Yuxuan Wang, Fan Yu, Tianyu Zhao, Hao Wang, Xiang Lv, Hui Wang, Chongjia Ni, Xian Shi, Keyu An, Guanrou Yang, Yabin Li, Yanni Chen, Zhifu Gao, Qian Chen, Yue Gu, Mengzhe Chen, Yafeng Chen, Shiliang Zhang, Wen Wang, and Jieping Ye. Cosyvoice 3: Towards in-the-wild speech generation via scaling-up and post-training. *CoRR*, abs/2505.17589, 2025. doi: 10.48550/ARXIV.2505.17589. URL https://doi.org/10.48550/arXiv.2505.17589.

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 Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pp. 18617–18629. Association for Computational Linguistics, 2025. doi: 10.18653/V1/2025.ACL-LONG.912. URL https://doi.org/10.18653/v1/2025.acl-long.912.

Pengchao Feng, Chao-Hong Tan, Qian Chen, Wen Wang, Xiangang Li, and Xie Chen. Efficient chain-of-modality reasoning via progressive compression for spoken language models. *CoRR*, abs/2607.19932, 2026. doi: 10.48550/ARXIV.2607.19932. URL https://doi.org/10.48550/arXiv.2607.19932.

Sachin Goyal, Ziwei Ji, Ankit Singh Rawat, Aditya Krishna Menon, Sanjiv Kumar, and Vaishnavh Nagarajan. Think before you speak: Training language models with pause tokens. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria,*

May 7-11, 2024. OpenReview.net, 2024. URL https://openreview.net/forum?id=ph04CRkPdC.

Hasan Abed Al Kader Hammoud, Kumail Alhamoud, Abed Hammoud, Elie Bou-Zeid, Marzyeh Ghassemi, and Bernard Ghanem. Train long, think short: Curriculum learning for efficient reasoning. *CoRR*, abs/2508.08940, 2025. doi: 10.48550/ARXIV.2508.08940. URL https://doi.org/10.48550/arXiv.2508.08940.

Mohammad Javad Hosseini, Hannaneh Hajishirzi, Oren Etzioni, and Nate Kushman. Learning to solve arithmetic word problems with verb categorization. In Alessandro Moschitti, Bo Pang, and Walter Daelemans (eds.), *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pp. 523–533. ACL, 2014. doi: 10.3115/V1/D14-1058. URL https://doi.org/10.3115/v1/d14-1058.

Yuxuan Hu, Heng Lu, Ruchao Fan, Yao Qian, Xiaofei Wang, Jian Xue, Heming Wang, Shuohang Wang, Young-Jin Kim, Yelong Shen, and Jinyu Li. Preserving speech-to-text LLM capabilities in speech-to-speech generation. *CoRR*, abs/2606.30944, 2026. doi: 10.48550/ARXIV.2606.30944. URL https://doi.org/10.48550/arXiv.2606.30944.

Ailin Huang, Bingxin Li, Bruce Wang, Boyong Wu, Chao Yan, Chengli Feng, Heng Wang, Hongyu Zhou, Hongyuan Wang, Jingbei Li, Jianjian Sun, Joanna Wang, Mingrui Chen, Peng Liu, Ruihang Miao, Shilei Jiang, Fei Tian, Wang You, Xi Chen, Xuerui Yang, Yechang Huang, Yuxiang Zhang, Zheng Ge, Zheng Gong, Zhewei Huang, Zixin Zhang, Bin Wang, Bo Li, Buyun Ma, Changxin Miao, Changyi Wan, Chen Xu, Dapeng Shi, Dingyuan Hu, Enle Liu, Guanzhe Huang, Gulin Yan, Hanpeng Hu, Haonan Jia, Jiahao Gong, Jiaoren Wu, Jie Wu, Jie Yang, Junzhe Lin, Kaixiang Li, Lei Xia, Longlong Gu, Ming Li, Nie Hao, Ranchen Ming, Shaoliang Pang, Siqi Liu, Song Yuan, Tiancheng Cao, Wen Li, Wenqing He, Xu Zhao, Xuelin Zhang, Yanbo Yu, Yinmin Zhong, Yu Zhou, Yuanwei Liang, Yuanwei Lu, Yuxiang Yang, Zidong Yang, Zili Zhang, Binxing Jiao, Heung-Yeung Shum, Jiansheng Chen, Jing Li, Xiangyu Zhang, Xinhao Zhang, Yibo Zhu, Daxin Jiang, Shaohua Kevin Zhou, and Chen Hu. Step-audio-aqaa: a fully end-to-end expressive large audio language model. *CoRR*, abs/2506.08967, 2025. doi: 10.48550/ARXIV.2506.08967. URL https://doi.org/10.48550/arXiv.2506.08967.

Sieun Hyeon, Kyudan Jung, Jaehee Won, Nam-Joon Kim, Hyun Gon Ryu, Hyuk-Jae Lee, and Jaeyoung Do. Mathspeech: Leveraging small lms for accurate conversion in mathematical speech-to-formula. In Toby Walsh, Julie Shah, and Zico Kolter (eds.), *Thirty-Ninth AAAI Conference on Artificial Intelligence, Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence, Fifteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2025, Philadelphia, PA, USA, February 25 - March 4, 2025*, pp. 24194–24202. AAAI Press, 2025. doi: 10.1609/AAAI.V39I23.34595. URL https://doi.org/10.1609/aaai.v39i23.34595.

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. *CoRR*, abs/2504.18425, 2025. doi: 10.48550/ARXIV.2504.18425. URL https://doi.org/10.48550/arXiv.2504.18425.

Rik Koncel-Kedziorski, Hannaneh Hajishirzi, Ashish Sabharwal, Oren Etzioni, and Siena Dumas Ang. Parsing algebraic word problems into equations. *Trans. Assoc. Comput. Linguistics*, 3:585–597, 2015. doi: 10.1162/TACL_A_00160. URL https://doi.org/10.1162/tacl_a_00160.

Thinking Machines Lab. Interaction models: A scalable approach to human-ai collaboration. *Thinking Machines Lab: Connectionism*, May 2026. doi: 10.64434/tml.20260511. https://thinkingmachines.ai/blog/interaction-models/.

Jia Li, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Huang, Kashif Rasul, Longhui Yu, Albert Q Jiang, Ziju Shen, et al. Numinamath: The largest public dataset in ai4maths with 860k pairs of competition math problems and solutions. *Hugging Face repository*, 2024.

Jiaqi Li, Chaoren Wang, Xiaohai Tian, Mingjie Chen, Xinyu Liang, Xu Li, Yufan Lin, Junwen Qiu, Jun Zhang, Lu Lu, Haizhou Li, and Zhizheng Wu. Flexislm: A dynamic and controllable frame rate spoken language model. *CoRR*, abs/2606.31247, 2026a. doi: 10.48550/ARXIV.2606.31247. URL https://doi.org/10.48550/arXiv.2606.31247.

Jijie Li, Li Du, Hanyu Zhao, Bowen Zhang, Liangdong Wang, Boyan Gao, Guang Liu, and Yonghua Lin. Infinity instruct: Scaling instruction selection and synthesis to enhance language models. *CoRR*, abs/2506.11116, 2025a. doi: 10.48550/ARXIV.2506.11116. URL https://doi.org/10.48550/arXiv.2506.11116.

Jindong Li, Yali Fu, Li Fan, Jiahong Liu, Yao Shu, Chengwei Qin, Menglin Yang, Irwin King, and Rex Ying. Implicit reasoning in large language models: A comprehensive survey. *CoRR*, abs/2509.02350, 2025b. doi: 10.48550/ARXIV.2509.02350. URL https://doi.org/10.48550/arXiv.2509.02350.

Shufan Li and Aditya Grover. Predgen: Accelerated inference of large language models through input-time speculation for real-time speech interaction. *CoRR*, abs/2506.15556, 2025. doi: 10.48550/ARXIV.2506.15556. URL https://doi.org/10.48550/arXiv.2506.15556.

Tianpeng Li, Jun Liu, Tao Zhang, Yuanbo Fang, Da Pan, Mingrui Wang, Zheng Liang, Zehuan Li, Mingan Lin, Guosheng Dong, Jianhua Xu, Haoze Sun, Zenan Zhou, and Weipeng Chen. Baichuan-audio: A unified framework for end-to-end speech interaction. *CoRR*, abs/2502.17239, 2025c. doi: 10.48550/ARXIV.2502.17239. URL https://doi.org/10.48550/arXiv.2502.17239.

Yangzhuo Li, Shengpeng Ji, Yifu Chen, Tianle Liang, Haoyu Yang, Junbo Li, Jun Fang, Lin Li, and Qingyang Hong. Dual-reasoner: Bridging interleaved atomicity and streaming latency via thinking-while-talking. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the Association for Computational Linguistics, ACL 2026, San Diego, California, United States, July 2-7, 2026*, pp. 4081–4105. Association for Computational Linguistics, 2026b. doi: 10.18653/V1/2026.FINDINGS-ACL.199. URL https://doi.org/10.18653/v1/2026.findings-acl.199.

Zeju Li, Jianyuan Zhong, Ziyang Zheng, Xiangyu Wen, Zhijian Xu, Yingying Cheng, Fan Zhang, and Qiang Xu. Compressing chain-of-thought in llms via step entropy. *CoRR*, abs/2508.03346, 2025d. doi: 10.48550/ARXIV.2508.03346. URL https://doi.org/10.48550/arXiv.2508.03346.

Tianle Liang, Yifu Chen, Shengpeng Ji, Yijun Chen, Zhiyang Jia, Jingyu Lu, Fan Zhuo, Xueyi Pu, Yangzhuo Li, and Zhou Zhao. Voxmind: An end-to-end agentic spoken dialogue system. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2026, San Diego, California, United States, July 2-7, 2026*, pp. 10117–10139. Association for Computational Linguistics, 2026. doi: 10.18653/V1/2026.ACL-LONG.459. URL https://doi.org/10.18653/v1/2026.acl-long.459.

Bin Lin, Bo Zhao, Boyang Zhang, Boyong Wu, Chao Yan, Chen Geng, Chen Wu, Cheng Yi, Chengli Feng, Chenglin Zhu, et al. Stepaudio 3 realtime technical report. *arXiv preprint arXiv:2609.14005*, 2026.

Junhong Lin, Xinyue Zeng, Jie Zhu, Song Wang, Julian Shun, Jun Wu, and Dawei Zhou. Plan and budget: Effective and efficient test-time scaling on large language model reasoning. *CoRR*, abs/2505.16122, 2025a. doi: 10.48550/ARXIV.2505.16122. URL https://doi.org/10.48550/arXiv.2505.16122.

Yueqian Lin, Zhengmian Hu, Qinsi Wang, Yudong Liu, Hengfan Zhang, Jayakumar Subramanian, Nikos Vlassis, Hai Helen Li, and Yiran Chen. Voice evaluation of reasoning ability: Diagnosing

the modality-induced performance gap. *CoRR*, abs/2509.26542, 2025b. doi: 10.48550/ARXIV.2509.26542. URL https://doi.org/10.48550/arXiv.2509.26542.

Siyuan Liu, Jiahui Xu, Feng Jiang, Kuang Wang, Zefeng Zhao, Chu-Ren Huang, Jinghang Gu, Changqing Yin, and Haizhou Li. Discourse-aware dual-track streaming response for low-latency spoken dialogue systems. *CoRR*, abs/2602.23266, 2026a. doi: 10.48550/ARXIV.2602.23266. URL https://doi.org/10.48550/arXiv.2602.23266.

Zhenyu Liu, Xuanyu Zhang, Yunxin Li, Qixun Teng, Shenyuan Jiang, Haolan Chen, Mingjun Zhao, Fanbo Meng, Yu Xu, Yancheng He, Baotian Hu, Haizhou Li, and Min Zhang. Hierarchical acoustic-semantic modeling: Modality separation and semantic coherence for full-duplex slms. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2026, San Diego, California, United States, July 2-7, 2026*, pp. 9264–9280. Association for Computational Linguistics, 2026b. doi: 10.18653/V1/2026.ACL-LONG.419. URL https://doi.org/10.18653/v1/2026.acl-long.419.

Ziyang Ma, Zhuo Chen, Yuping Wang, Eng Siong Chng, and Xie Chen. Audio-cot: Exploring chain-of-thought reasoning in large audio language model. In *IEEE Automatic Speech Recognition and Understanding Workshop, ASRU 2025, Honolulu, HI, USA, December 6-10, 2025*, pp. 1–6. IEEE, 2025. doi: 10.1109/ASRU65441.2025.11434628. URL https://doi.org/10.1109/ASRU65441.2025.11434628.

Kentaro Mitsui, Koh Mitsuda, Toshiaki Wakatsuki, Yukiya Hono, and Kei Sawada. PSLM: parallel generation of text and speech with llms for low-latency spoken dialogue systems. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, volume EMNLP 2024 of *Findings of ACL*, pp. 2692–2700. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.FINDINGS-EMNLP.151. URL https://doi.org/10.18653/v1/2024.findings-emnlp.151.

OpenAI. Gpt-4o system card. *CoRR*, abs/2410.21276, 2024. doi: 10.48550/ARXIV.2410.21276. URL https://doi.org/10.48550/arXiv.2410.21276.

Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Menglin Xia, Xufang Luo, Jue Zhang, Qingwei Lin, Victor Rühle, Yuqing Yang, Chin-Yew Lin, H. Vicky Zhao, Lili Qiu, and Dongmei Zhang. Llmlingua-2: Data distillation for efficient and faithful task-agnostic prompt compression. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, volume ACL 2024 of *Findings of ACL*, pp. 963–981. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.FINDINGS-ACL.57. URL https://doi.org/10.18653/v1/2024.findings-acl.57.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. Are NLP models really able to solve simple math word problems? In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tür, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou (eds.), *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pp. 2080–2094. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.NAACL-MAIN.168. URL https://doi.org/10.18653/v1/2021.naacl-main.168.

Jacob Pfau, William Merrill, and Samuel R. Bowman. Let's think dot by dot: Hidden computation in transformer language models. *CoRR*, abs/2404.15758, 2024. doi: 10.48550/ARXIV.2404.15758. URL https://doi.org/10.48550/arXiv.2404.15758.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 28492–28518. PMLR, 2023. URL https://proceedings.mlr.press/v202/radford23a.html.

Subhro Roy and Dan Roth. Solving general arithmetic word problems. In Lluís Màrquez, Chris Callison-Burch, Jian Su, Daniele Pighin, and Yuval Marton (eds.), *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pp. 1743–1752. The Association for Computational Linguistics, 2015. doi: 10.18653/V1/D15-1202. URL https://doi.org/10.18653/v1/d15-1202.

Hejian Sang, Yuanda Xu, Zhengze Zhou, Ran He, Zhipeng Wang, and Jiachen Sun. On-policy self-distillation for reasoning compression. *CoRR*, abs/2603.05433, 2026. doi: 10.48550/ARXIV.2603.05433. URL https://doi.org/10.48550/arXiv.2603.05433.

Xuan Shen, Yizhou Wang, Xiangxi Shi, Yanzhi Wang, Pu Zhao, and Jiuxiang Gu. Efficient reasoning with hidden thinking. *CoRR*, abs/2501.19201, 2025. doi: 10.48550/ARXIV.2501.19201. URL https://doi.org/10.48550/arXiv.2501.19201.

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? *CoRR*, abs/2510.07497, 2025. doi: 10.48550/ARXIV.2510.07497. URL https://doi.org/10.48550/arXiv.2510.07497.

Yang Sui, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu, Andrew Wen, Shaochen Zhong, Na Zou, Hanjie Chen, and Xia Hu. Stop overthinking: A survey on efficient reasoning for large language models. *Trans. Mach. Learn. Res.*, 2025, 2025. URL https://openreview.net/forum?id=HvoG8SxggZ.

Peiwen Sun, Sitong Cheng, Xiangtai Li, Zhen Ye, Huadai Liu, Honggang Zhang, Wei Xue, and Yike Guo. Both ears wide open: Towards language-driven spatial audio generation. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=qPx3i9sMxv.

Chao-Hong Tan, Qian Chen, Wen Wang, Chong Deng, Qinglin Zhang, Luyao Cheng, Hai Yu, Xin Zhang, Xiang Lv, Tianyu Zhao, Chong Zhang, Yukun Ma, Yafeng Chen, Hui Wang, Jiaqing Liu, and Jieping Ye. Omnidrca: Parallel speech-text foundation model via dual-resolution speech representations and contrastive alignment. *CoRR*, abs/2506.09349, 2025a. doi: 10.48550/ARXIV.2506.09349. URL https://doi.org/10.48550/arXiv.2506.09349.

Wenhui Tan, Jiaze Li, Jianzhong Ju, Zhenbo Luo, Ruihua Song, and Jian Luan. Think silently, think fast: Dynamic latent compression of LLM reasoning chains. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025b. URL http://papers.nips.cc/paper_files/paper/2025/hash/0706261aedab63814a2b73c32564b4c4-Abstract-Conference.html.

GlobalHealthAtlas Team. From knowledge to inference: Scaling laws of specialized reasoning on globalhealthatlas. 2026a.

Qwen Team. Qwen3 technical report. *CoRR*, abs/2505.09388, 2025a. doi: 10.48550/ARXIV.2505.09388. URL https://doi.org/10.48550/arXiv.2505.09388.

Qwen Team. Qwen3-omni technical report. *CoRR*, abs/2509.17765, 2025b. doi: 10.48550/ARXIV.2509.17765. URL https://doi.org/10.48550/arXiv.2509.17765.

Qwen Team. Qwen3-vl technical report. *CoRR*, abs/2511.21631, 2025c. doi: 10.48550/ARXIV.2511.21631. URL https://doi.org/10.48550/arXiv.2511.21631.

Qwen Team. Qwen3.5-omni technical report. *CoRR*, abs/2604.15804, 2026b. doi: 10.48550/ARXIV.2604.15804. URL https://doi.org/10.48550/arXiv.2604.15804.

Tongyi Fun Team, Qian Chen, Luyao Cheng, Chong Deng, Xiangang Li, Jiaqing Liu, Chao-Hong Tan, Wen Wang, Junhao Xu, Jieping Ye, Qinglin Zhang, Qiquan Zhang, and Jingren Zhou. Fun-audio-chat technical report. *CoRR*, abs/2512.20156, 2025. doi: 10.48550/ARXIV.2512.20156. URL https://doi.org/10.48550/arXiv.2512.20156.

Chaoren Wang, Heng Lu, Xueyao Zhang, Shujie Liu, Yan Lu, Jinyu Li, and Zhizheng Wu. Closing the modality reasoning gap for speech large language models. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2026, San Diego, California, United States, July 2-7, 2026*, pp. 18821–18835. Association for Computational Linguistics, 2026. doi: 10.18653/V1/2026.ACL-LONG.857. URL https://doi.org/10.18653/v1/2026.acl-long.857.

Chen Wang, Minpeng Liao, Zhongqiang Huang, Junhong Wu, Chengqing Zong, and Jiajun Zhang. Blsp-emo: Towards empathetic large speech-language models. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pp. 19186–19199. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.EMNLP-MAIN.1070. URL https://doi.org/10.18653/v1/2024.emnlp-main.1070.

Mingqiu Wang, Wei Han, Izhak Shafran, Zelin Wu, Chung-Cheng Chiu, Yuan Cao, Nanxin Chen, Yu Zhang, Hagen Soltau, Paul K. Rubenstein, Lukas Zilka, Dian Yu, Golan Pundak, Nikhil Siddhartha, Johan Schalkwyk, and Yonghui Wu. SLM: bridge the thin gap between speech and text foundation models. In *IEEE Automatic Speech Recognition and Understanding Workshop, ASRU 2023, Taipei, Taiwan, December 16-20, 2023*, pp. 1–8. IEEE, 2023. doi: 10.1109/ASRU57964.2023.10389703. URL https://doi.org/10.1109/ASRU57964.2023.10389703.

Yaoting Wang, Shengqiong Wu, Yuecheng Zhang, Shuicheng Yan, Ziwei Liu, Jiebo Luo, and Hao Fei. Multimodal chain-of-thought reasoning: A comprehensive survey. *CoRR*, abs/2503.12605, 2025. doi: 10.48550/ARXIV.2503.12605. URL https://doi.org/10.48550/arXiv.2503.12605.

Chengwei Wei, Bin Wang, Jung-Jae Kim, and Nancy F. Chen. Towards spoken mathematical reasoning: Benchmarking speech-based models over multi-faceted math problems. *CoRR*, abs/2505.15000, 2025. doi: 10.48550/ARXIV.2505.15000. URL https://doi.org/10.48550/arXiv.2505.15000.

Donghang Wu, Haoyang Zhang, Jun Chen, Xiangyu Tony Zhang, Hexin Liu, Eng Siong Chng, Fei Tian, Xuerui Yang, Xiangyu Zhang, Daxin Jiang, and Gang Yu. Mind-paced speaking: A dual-brain approach to real-time reasoning in spoken language models. *CoRR*, abs/2510.09592, 2025. doi: 10.48550/ARXIV.2510.09592. URL https://doi.org/10.48550/arXiv.2510.09592.

Heming Xia, Chak Tou Leong, Wenjie Wang, Yongqi Li, and Wenjie Li. Tokenskip: Controllable chain-of-thought compression in llms. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, EMNLP 2025, Suzhou, China, November 4-9, 2025*, pp. 3351–3363. Association for Computational Linguistics, 2025. doi: 10.18653/V1/2025.EMNLP-MAIN.165. URL https://doi.org/10.18653/v1/2025.emnlp-main.165.

LLM-Core Xiaomi. Mimo-audio: Audio language models are few-shot learners. *CoRR*, abs/2512.23808, 2025. doi: 10.48550/ARXIV.2512.23808. URL https://doi.org/10.48550/arXiv.2512.23808.

Jingran Xie, Shun Lei, Yue Yu, Yang Xiang, Hui Wang, Xixin Wu, and Zhiyong Wu. Leveraging chain of thought towards empathetic spoken dialogue without corresponding question-answering data. In *2025 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2025, Hyderabad, India, April 6-11, 2025*, pp. 1–5. IEEE, 2025a. doi: 10.1109/ICASSP49660.2025.10889870. URL https://doi.org/10.1109/ICASSP49660.2025.10889870.

Zhifei Xie and Changqiao Wu. Mini-omni: Language models can hear, talk while thinking in streaming. *CoRR*, abs/2408.16725, 2024. doi: 10.48550/ARXIV.2408.16725. URL https://doi.org/10.48550/arXiv.2408.16725.

Zhifei Xie, Mingbao Lin, Zihang Liu, Pengcheng Wu, Shuicheng Yan, and Chunyan Miao. Audio-reasoner: Improving reasoning capability in large audio language models. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, EMNLP 2025, Suzhou, China, November 4-9, 2025*, pp. 23829–23851. Association for Computational Linguistics, 2025b. doi: 10.18653/V1/2025.EMNLP-MAIN.1216. URL https://doi.org/10.18653/v1/2025.emnlp-main.1216.

Zhifei Xie, Ziyang Ma, Zihang Liu, Kaiyu Pang, Hongyu Li, Jialin Zhang, Yue Liao, Deheng Ye, Chunyan Miao, and Shuicheng Yan. Mini-omni-reasoner: Token-level thinking-in-speaking in large speech models. *CoRR*, abs/2508.15827, 2025c. doi: 10.48550/ARXIV.2508.15827. URL https://doi.org/10.48550/arXiv.2508.15827.

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. *CoRR*, abs/2503.20215, 2025a. doi: 10.48550/ARXIV.2503.20215. URL https://doi.org/10.48550/arXiv.2503.20215.

Silei Xu, Wenhao Xie, Lingxiao Zhao, and Pengcheng He. Chain of draft: Thinking faster by writing less. *CoRR*, abs/2502.18600, 2025b. doi: 10.48550/ARXIV.2502.18600. URL https://doi.org/10.48550/arXiv.2502.18600.

Zhangchen Xu, Fengqing Jiang, Luyao Niu, Yuntian Deng, Radha Poovendran, Yejin Choi, and Bill Yuchen Lin. Magpie: Alignment data synthesis from scratch by prompting aligned llms with nothing. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025c. URL https://openreview.net/forum?id=Pnk7vMbznK.

Hongfei Xue, Yuhao Liang, Bingshen Mu, Shiliang Zhang, Mengzhe Chen, Qian Chen, and Lei Xie. E-chat: Emotion-sensitive spoken dialogue system with large language models. In Yanmin Qian, Qin Jin, Zhijian Ou, Zhenhua Ling, Zhiyong Wu, Ya Li, Lei Xie, and Jianhua Tao (eds.), *14th IEEE International Symposium on Chinese Spoken Language Processing, ISCSLP 2024, Beijing, China, November 7-10, 2024*, pp. 586–590. IEEE, 2024. doi: 10.1109/ISCSLP63861.2024.10800447. URL https://doi.org/10.1109/ISCSLP63861.2024.10800447.

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 (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025, Suzhou, China, November 4-9, 2025*, pp. 17211–17242. Association for Computational Linguistics, 2025. doi: 10.18653/V1/2025.FINDINGS-EMNLP.933. URL https://doi.org/10.18653/v1/2025.findings-emnlp.933.

Zhen Ye, Xu Tan, Yiming Li, Guangyan Zhang, Chimin Chan, Haohe Liu, Zhengxi Liu, Hongzhan Lin, Zheqi Dai, Xinshen Zhang, Peiwen Sun, Qiuqiang Kong, and Wei Xue. Which speech representation better matches text-native reasoning? A study of speech-text alignment on frame rate and representation. *CoRR*, abs/2606.12199, 2026. doi: 10.48550/ARXIV.2606.12199. URL https://doi.org/10.48550/arXiv.2606.12199.

Wenyi Yu, Siyin Wang, Xiaoyu Yang, Xianzhao Chen, Xiaohai Tian, Jun Zhang, Guangzhi Sun, Lu Lu, Yuxuan Wang, and Chao Zhang. Salmonn-omni: A standalone speech LLM without codec injection for full-duplex conversation. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/233aee920dab065709145371b5900b8f-Abstract-Conference.html.

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. *CoRR*, abs/2412.02612, 2024. doi: 10.48550/ARXIV.2412.02612. URL https://doi.org/10.48550/arXiv.2412.02612.

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 Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, volume EMNLP 2023 of *Findings of ACL*, pp. 15757–15773. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.FINDINGS-EMNLP.1055. URL https://doi.org/10.18653/v1/2023.findings-emnlp.1055.

Dong Zhang, Xin Zhang, Jun Zhan, Shimin Li, Yaqian Zhou, and Xipeng Qiu. Speechgpt-gen: Scaling chain-of-information speech generation. *CoRR*, abs/2401.13527, 2024. doi: 10.48550/ARXIV.2401.13527. URL https://doi.org/10.48550/arXiv.2401.13527.

Zhen Zhang, Xuehai He, Weixiang Yan, Ao Shen, Chenyang Zhao, and Xin Wang. Soft thinking: Unlocking the reasoning potential of llms in continuous concept space. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/f7396d1c54d51416958d63e285377103-Abstract-Conference.html.

Xingjian Zhao, Zhe Xu, Qinyuan Cheng, Zhaoye Fei, Luozhijie Jin, Yang Wang, Hanfu Chen, Yaozhou Jiang, Qinghui Gao, Ke Chen, Ruixiao Li, Mingshu Chen, Ruiming Wang, Wenbo Zhang, Yiyang Zhang, Donghua Yu, Yang Gao, Xiaogui Yang, Yitian Gong, Yuanfan Xu, Yaqian Zhou, Xuanjing Huang, and Xipeng Qiu. Moss-speech: Towards true speech-to-speech models without text guidance. *CoRR*, abs/2510.00499, 2025. doi: 10.48550/ARXIV.2510.00499. URL https://doi.org/10.48550/arXiv.2510.00499.

Yufan Zhuang, Liyuan Liu, Chandan Singh, Jingbo Shang, and Jianfeng Gao. Mixture of inputs: Text generation beyond discrete token sampling. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/25c61012c845908bff2ea6223cdc6192-Abstract-Conference.html.

Wenhao Zou, Yuwei Miao, Zhanyu Ma, Jun Xu, Jiuchong Gao, Jinghua Hao, Renqing He, and Jingwen Xu. Lts-voiceagent: A listen-think-speak framework for efficient streaming voice interaction via semantic triggering and incremental reasoning. *CoRR*, abs/2601.19952, 2026. doi: 10.48550/ARXIV.2601.19952. URL https://doi.org/10.48550/arXiv.2601.19952.

A LIMITATION

Our experiments are limited to spoken mathematical question answering across AddSub, SingleEq, MultiArith, SVAMP, and GSM8K. We have not established generalization to open-domain conversation, multi-turn dialogue, or other reasoning-intensive tasks such as commonsense or logical QA. The motivation for focusing on the mathematical QA scenario is discussed in Section 1. Moreover, since our training data for stepwise reasoning was constructed specifically for mathematical QA, evaluating the model on out-of-domain questions would not accurately reflect its true performance.

Reinforcement learning (RL) was not incorporated into our framework. Although RL holds promise for further boosting model performance within the THINKGAP interaction paradigm, its incorporation at this stage risks introducing confounding factors that would compromise a clean evaluation of THINKGAP's intrinsic contribution. This design choice was made to ensure a more controlled and

interpretable assessment of the proposed paradigm. We consider the combination of reinforcement learning with THINKGAP a promising avenue for future research.

The evaluation uses automatic metrics—ASR transcription followed by LLM-as-judge scoring for answer accuracy, plus WER and UTMOS for speech output—and does not include a full human evaluation. These metrics may not capture speech naturalness, prosody, judge reliability, or overall user experience. We also report point estimates without repeated-run uncertainty or significance tests. Finally, the architecture-transfer experiments replace or modify components in the alternative SLMs, so they demonstrate transfer to those tested configurations rather than architecture-independent effectiveness. Human listening studies, repeated trials, and broader task and architecture coverage are needed.

B FRAME RATE ANALYSIS

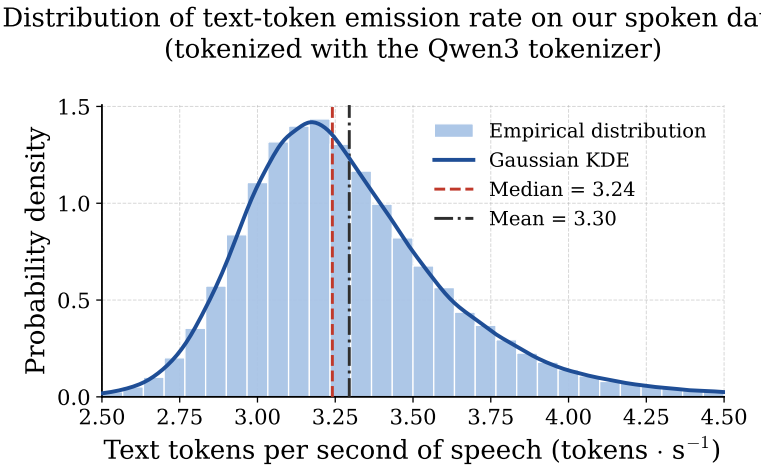

Figure 2: Distribution of the text-token emission rate on our spoken training set, tokenized with the Qwen3 tokenizer. The per-utterance rate is concentrated around 3.3 text tokens per second of speech.

Fig. 2 characterizes how many text tokens the SLM must emit per second of synthesized speech. The empirical distribution (light bars) is overlaid with a Gaussian kernel density estimate (solid curve). The text rate f_w is tightly concentrated, with a median of 3.24 and a mean of 3.30 tokens $\cdot$ s^{-1}. This shows that, on average, the model needs to produce roughly three text tokens for every second of speech, which directly informs the idle budget used in our design.

Fig. 3 shows that the largest gain occurs where the idle budget is greatest, although gains across the lower frame rates are not monotonic. Interestingly, the 25 Hz baseline has the lowest average accuracy among the tested frame rates, and with THINKGAP, the 25 Hz configuration instead achieves the highest average accuracy, coinciding with its larger available gap budget. These experiments show that a high frame rate can provide useful in-gap reasoning capacity in the evaluated setup.

C VALIDATION OF SEGMENT ALIGNMENT

We empirically examine two claims from Section 3.2: (1) redistributing the idle budget has little effect on text-only generation quality within the tested range, and (2) preventing non-spoken tokens from directly conditioning the speech head is beneficial. Detailed analyses follow.

As shown in Fig. 4, we evaluate five segment-alignment configurations by inserting idle tokens after each sentence, where the insertion length is $\alpha \in \{0, \frac{1}{6}, \frac{1}{3}, \frac{1}{2}, \frac{2}{3}\}$ times the sentence length. These insertion lengths reflect the approximate $5 : 3 : 2$ ratio of speech, text, and idle tokens at a 5 Hz speech-token rate. The left axis reports average Spoken Math QA accuracy for **S2T** (text-only generation), **S2S w/ invisible** (parallel speech-and-text generation with direct current-slot gating of

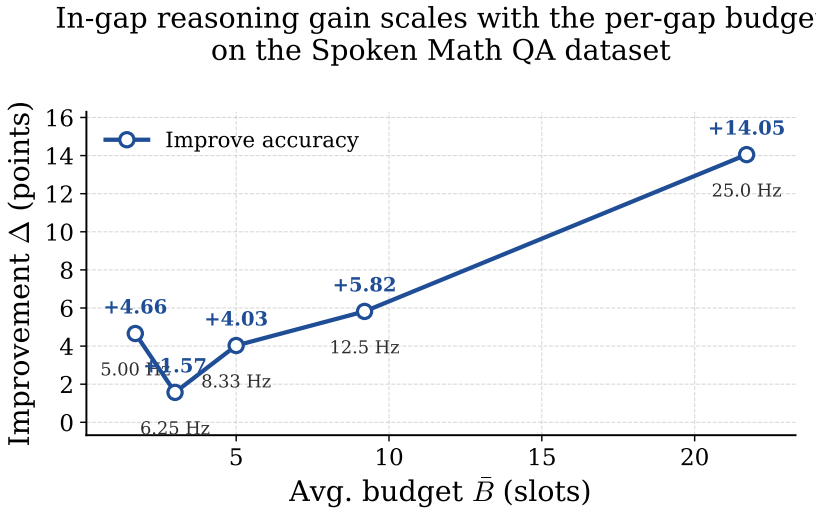

Figure 3: **ThinkGap gain scales with the per-gap budget on the Spoken Math QA dataset.** As the average per-gap budget $\bar{B}$ (equivalently, a higher speech-token frame rate) increases, the accuracy improvement Δ from in-gap reasoning grows steadily, reaching $+14.05$ points at $\bar{B} \approx 21.7$ slots (25.0 Hz).

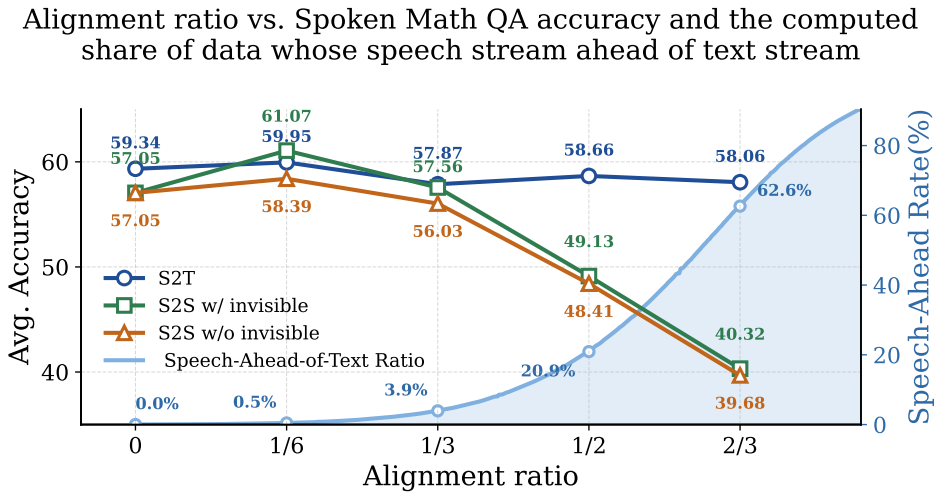

Figure 4: Spoken Math QA accuracy under three settings versus the alignment ratio (left), overlaid with the computed fraction of examples in which speech advances ahead of text (right). The speech-ahead-of-text fraction rises with the alignment ratio and is associated with lower accuracy in the speech-output settings.

non-spoken tokens), and **S2S w/o invisible** (without this constraint). The light-blue curve on the right axis is *derived*, rather than directly measured, from the text-token-rate distribution in Fig. 2. For the 5 Hz SLM used here, the effective text rate for utterance i after inserting idle tokens is $f'_{w,i} = (1+\alpha) f_{w,i}$ [6]. For a dataset of M utterances, we count an utterance as speech-ahead-of-text

[6]When computing the text rate used to assess whether speech advances ahead of text, we exclude the two text tokens artificially advanced by the alignment scheme. Following FunAudioChat, speech generation is intentionally delayed by two tokens relative to text generation; removing this fixed offset gives a more accurate text rate.

when $f'_{w,i} > 5$ and plot

$$P(\alpha) \;=\; \frac{1}{M}\sum_{i=1}^{M} \mathbf{1}[\,(1+\alpha) f_{w,i} > 5\,] \times 100\% \tag{14}$$

on a dense grid of α, with neighboring evaluations connected for visualization.

Accuracy on the text-only path (S2T) remains within approximately 58–60 across the tested alignment ratios, whereas both speech-output paths decline as more idle tokens are inserted. When the computed speech-ahead-of-text fraction remains low, the S2S setting with current-slot gating also stays at or above 57. These observations support the claim that redistributing the idle budget has little effect on text-only generation quality within the tested range. The increase in $P(\alpha)$ is associated with the decline in speech-output accuracy as the alignment ratio grows. Because $P(\alpha)$ is derived rather than experimentally manipulated, this result supports but does not by itself establish a causal explanation based on speech advancing ahead of text.

Finally, the gap between the green and orange curves suggests that directly gating non-spoken tokens from the speech head is beneficial in this experiment, although the validation introduces only one type of non-spoken token.

D Speech Quality Analysis

Table 5: **Comparison of WER and UTMOS across speech frame rates.** We report average WER and UTMOS of three settings—*Baseline*, *Alignment*, and *ThinkGap*—under five speech frame rates. WER is lower-is-better (↓) and UTMOS is higher-is-better (↑).

	25 Hz		12.5 Hz		8.33 Hz		6.25 Hz		5 Hz		Avg	
Setting	**WER↓**	**UTMOS↑**	**WER↓**	**UTMOS↑**	**WER↓**	**UTMOS↑**	**WER↓**	**UTMOS↑**	**WER↓**	**UTMOS↑**	**WER↓**	**UTMOS↑**
Baseline	0.119	4.43	0.104	4.44	**0.023**	**4.44**	0.135	4.43	0.100	4.43	**0.096**	4.43
Alignment	**0.029**	**4.45**	**0.025**	4.44	0.283	4.42	0.170	4.42	**0.028**	**4.44**	0.107	4.43
ThinkGap	0.088	4.44	0.090	**4.45**	0.090	**4.44**	**0.129**	**4.44**	0.188	4.43	0.117	**4.44**

We assess speech synthesis with WER between generated audio and the corresponding generated text after removing special and reasoning tokens, together with UTMOS as an automatic estimate of perceptual quality. Table 5 shows nearly unchanged average UTMOS (4.44 for ThinkGap versus 4.43 for the baseline) and a higher average WER (0.117 versus 0.096). WER varies substantially across frame rates, so these point estimates do not establish equivalence in intelligibility. Mismatches between spoken numerals and textual forms (e.g., *twelve* versus *12*) may contribute to WER on numerically dense examples (see Appendix F), but the present evaluation does not isolate that contribution. Human listening tests are required to assess naturalness, prosody, and overall user experience.

E Data Construction Pipeline

This appendix details an automatic pipeline that converts text-only mathematical reasoning corpora into speech dialogues and constructs compressed chain-of-thought (CoT) targets within the slack of the assistant's text channel. The pipeline preserves spoken content and enforces a final text–speech length invariant.

E.1 Pipeline overview

Given a text corpus of triples (question, reference CoT solution, final answer), the pipeline proceeds in five stages.

- **S1 Spoken-form reconstruction** (§E.2): an LLM splits the reference solution into coarse reasoning steps, writes one short spoken recap per step, and begins the first sentence with a direct analysis of the question.

- **S2 Speech rendering** (§E.3): the question and recaps are synthesized with voice-diverse TTS configurations.
- **S3 Discrete speech tokenization** (§E.4): waveforms are converted to 25 Hz CosyVoice3 tokens via chunked multi-GPU inference with order-preserving merging.
- **S4 Boundary localization** (§E.5): forced alignment maps each voiced segment to a token index; indices are snapped to the midpoint of the nearest silence gap, yielding one boundary per segment, with the last pinned to the total number of acoustic tokens.
- **S5 Budgeted inline reasoning** (§E.6): the text-channel slack of each segment is computed from the boundaries; each reasoning step is compressed to fit its slack and inserted between `<think>`/`</think>` markers; the tail is padded with `<|sil|>` tokens so that the length invariant holds exactly.

Design principles. Three principles guide the pipeline. (i) *Preserve source reasoning*: C1 is instructed to segment the reference solution by inserting markers rather than rewriting its content; later stages separately generate spoken recaps and compress hidden reasoning. (ii) *Validate, retry, and apply documented fallbacks*: each LLM call is checked by programmatic validators. Failures trigger bounded retries and the specified repair or degradation rule; unrecoverable samples are discarded. (iii) *Speech is authoritative*: spoken text is not truncated; when capacity is insufficient, reasoning is compressed or omitted.

E.2 Stage S1: Spoken-form reconstruction of a text CoT

Source normalisation. We start from public text-only mathematical reasoning corpora (GSM8K (Cobbe et al., 2021), NuminaMath (Li et al., 2024), and MathQA (Amini et al., 2019)), stored as (question, solution, answer) triples. For GSM8K, we remove inline calculator annotations `<<...>>` and the `####` answer marker via regular expressions, retaining the stripped rationale as the solution, the extracted formulas as metadata, and the string after `####` as the final answer. Samples whose solutions are shorter than 20 characters are discarded as degenerate.

Three-call decomposition. A single instruction that simultaneously segments, summarizes, and stylizes the solution is difficult to validate because failures are not attributable to a specific operation. We therefore decompose S1 into three separately validated LLM calls, each with a 100 s timeout and thinking disabled. **C1 (coarse segmentation)** receives the question and solution and is instructed to return the solution verbatim except for boundary-marker insertion. The prompt forbids other character-level edits and requests coarse granularity—one segment per reasoning *stage* (setup, computation, verification, or sub-problem), typically 2–5 segments per solution. **C2 (spoken recaps)** receives the question, numbered steps, and answer and writes exactly one recap per step as a numbered list. Recaps must match the solution language, contain at most 15 English words or Chinese characters, use plain text without LaTeX or Markdown, vary their openings for natural narration, and state the final answer in the last recap so that the audio alone contains a complete answer. **C3 (opening filler)** writes one casual spoken sentence that directly analyzes the question and serves as the opening of the spoken response, with at most 10 words or characters. Each prompt contains two few-shot exemplars, one English and one Chinese, demonstrating the language-matching rule.

Validation, retries and degradation. C1 is accepted only if all segments are non-empty, the segment count is at most 20, and concatenation preserves the source beginning: after stripping everything but letters, digits, and CJK characters, the first 10% of content-bearing characters must match exactly. This one-sided check deliberately catches the dominant failure mode (paraphrasing or rederiving instead of inserting markers) at negligible cost, while tolerating harmless tail differences.

C2 is accepted only if both lists are non-empty element-wise, the number of recaps equals the number of steps, each recap's detected language matches its step (CJK-to-alphabetic ratio threshold 0.3), and recaps are genuinely shorter than their steps (enforced only for steps longer than 80 characters). Each call is retried at most twice. If C2 still fails, count-alignment repair is attempted: extra recaps keep the first K-1 and the last (which carries the answer); fewer recaps trigger pairwise merging of the earliest steps until counts agree. If alignment is impossible, or if C1 or C3 exhausts retries, the sample is discarded.

Each surviving row augments the source with three fields: long segmented reasoning steps (future hidden thoughts), short spoken recaps (future speech), and the opening filler. The spoken recaps and filler define what is *said* and therefore the speech length, while the long steps define what is *thought* and must be compressed into the available slack.

E.3 Stage S2: Speech rendering

Question audio is synthesised with a commercial instruction-following TTS model (`gpt-4o-mini-tts`), chosen for its accurate rendering of mathematical formulas. To avoid speaker or prosody degeneracy, each utterance samples a voice uniformly from 13 voices and a style instruction from five casual, conversational paraphrases.

The assistant answer—already constrained to formula-free spoken language during generation—is synthesised *segment-wise* with a TTS model (`cosyvoice-v3-flash`) using two fixed voices. Segment-wise rendering guarantees the assistant audio is a concatenation of the acoustic units with known textual identity—precisely what stage S4 §E.5 recovers in token space.

Both renderers share the same fault-tolerance scheme: up to 8 attempts with truncated exponential backoff plus uniform jitter, retrying on rate limiting, server-side errors, and transport exceptions; hard failures are raised rather than silently writing empty files.

E.4 Stage S3: Discrete speech tokenization

The assistant waveform is discretised with the CosyVoice3 speech tokenizer (25 tokens per second) via ONNX Runtime with all graph optimisations enabled, one intra-op thread, and the CUDA execution provider (CPU fallback). Audio is resampled to 16 kHz and down-mixed to mono when necessary; features are 128-band Whisper log-mel spectrograms.

Due to the tokenizer's bounded receptive field, long audio is cut into 30s chunks with a tail rule: if the last chunk is shorter than 1s and more than one chunk exists, the last two are concatenated and re-split at their midpoint to avoid degenerately short chunks. Chunk token sequences are concatenated in order into a single token string per utterance; its length fixes the sample's text budget.

Tokenization is embarrassingly parallel but GPU-bound. We spawn one process per GPU (four in our setup), bind each to a single device, and shard by row-index residue. Each worker streams its records to its own temporary shard; the parent merges shards into a dense array indexed by row and *fails loudly* if any index is missing, preventing a silently truncated shard from producing a misaligned corpus. Rows already carrying a token field are passed through, making the stage restartable.

E.5 Stage S4: Cross-modal boundary localisation

Stage S5 needs to know *where*, in the 25 Hz token stream, each spoken segment ends. Although text-level segmentation is known exactly (the filler plus recaps), the assistant audio is tokenized as a single concatenated stream rather than per segment. This is deliberate: segment-wise tokenization produces token discontinuities at clip boundaries, which cause audible "thump" artifacts when the trained model generates between sentences. Concatenating audio before tokenization avoids this, but merges segment boundaries into one continuous token sequence—hence the need to re-identify and cut at segment positions, making this a forced-alignment rather than a segmentation problem.

Forced alignment. We run a 0.6B forced aligner (`Qwen3-ForcedAligner`) on the assistant audio against its full transcript to obtain word-level timestamps. The language tag is derived through CJK character detection. Alignment runs in batches of 8 in `bfloat16`; failed batches are retried item by item so that one pathological sample does not invalidate eight alignments. Unrecoverable items fall back to a degenerate boundary vector, with all boundaries at the total acoustic length, and are counted as failures.

From words to segments. The aligner emits a flat word sequence, so words must be attributed to segments. We match in normalised character space (lowercasing, punctuation removal, whitespace collapsing): cumulative character boundaries of segments are first located via monotone search (falling back to length-based estimates when a segment is not found verbatim), then aligned words

are walked with a monotone cursor (allowing a 10-character backward search), each word being assigned to the first segment whose character boundary its centre does not exceed. This formulation is robust to minor lexical differences (number formatting, contractions) and monotone by construction.

Snapping to silence gaps. Cutting exactly at aligned word ends is undesirable: boundaries inside voiced material can make neighboring chunks unnatural and misattribute budget frames. We therefore detect silence *in token space*: we identify 17 silence/breath token IDs, find maximal runs of at least 2 tokens, merge runs separated by at most 8 tokens (~ 320 ms) into single gaps, and place each boundary at the midpoint of the gap nearest to the aligned end time. The last boundary is pinned to the total acoustic token count; segments with no attributed word fall back to proportional estimates, which are also snapped. A forward pass enforces monotonicity. Samples with incorrect boundary counts or non-monotone boundaries before repair are flagged, and the stage exits with a nonzero status if any flag is raised, preventing defective batches from entering training unnoticed.

E.6 Stage S5: Budget-constrained inline reasoning

Stage S5 constructs "thinking in the gaps." Its input is the boundary vector, voiced texts, long reasoning steps, and assistant speech token string; its output is a single assistant text string satisfying the length invariant.

Interleaving layout. The text channel alternates voiced text and reasoning blocks: after each voiced segment except the last, the reasoning block for the next step is inserted between `<think>`/`</think>` markers. The reasoning block for a step thus occupies the text positions that remain *while the audio of the previous segment is still playing*: the model reads out a short recap and spends the rest of that segment's text budget thinking about the next step. The last voiced segment receives no reasoning block, because there is no subsequent step to prepare and because its recap already states the answer.

Budget recursion with carry-over. From the boundaries, we recover the acoustic length of each segment and, by dividing by the group size G, its text-slot budget c_j. Let m_j be the cost of the spoken words under the *model's own* tokenizer and let s_{j-1} be the surplus carried from preceding segments, with $s_0 = 0$. For each non-final segment,

$$a_j = c_j - m_j + s_{j-1}, \qquad b_j = \max(a_j - 2, 0), \qquad s_j = a_j - |q^{(j)}|, \tag{15}$$

where b_j is the reasoning-content budget after reserving the two marker positions and $|q^{(j)}| = k_j + 2$ when a nonempty k_j-token reasoning block is inserted, or 0 when no block is inserted. The compressor returns $k_j \le b_j$ target-SLM tokens. For the final segment, $q^{(J)}$ is empty and $s_J = a_J$. Thus, a local deficit is carried forward as a negative surplus, while unused capacity remains available to later segments. Carry-over is crucial because budgets are integral and highly non-uniform: a segment that overspends is repaid by later segments, and slack is not discarded. The D leading delay tokens do not participate in any per-segment budget—they are a pure channel delay—but they do enter the global target length, so final alignment is settled globally rather than segment by segment.

Length-controlled reasoning compression. The compressor must return text with at most the target number of tokens *under the training tokenizer*. We therefore replace the tokenizer used internally by the compression library with a wrapper around the speech-LLM's own tokenizer, so that the budget the compressor optimizes is exactly the budget the decoder will spend.

Off-the-shelf compressors treat the target token count as a soft objective and can overshoot. Here, an overshoot is a constraint violation rather than merely a quality issue. We therefore wrap the perplexity-based operators in an iterative tightening loop: after each pass, we measure the realized length and, if it exceeds the target, rerun compression with a rescaled target. We allow at most five passes and then apply hard truncation as a guaranteed fallback. For the question-aware token-classification operator, we reuse the public LLMLingua-2 retention head with an explicitly conditioned input. The question context—the question plus the thoughts and recaps produced so far, mirroring the causal information state at inference time—is prepended to the reasoning step and influences each token's retention probability through self-attention; probabilities are read only from

Table 6: Pipeline configuration.

Stage	Parameter	Value	Note
S1	generator LLM	Qwen3-Plus class	thinking mode off, 100 s timeout
	target segments K	2–5 (cap 20)	coarse, stage-level splitting
	recap length	≤ 15 words/chars	last recap states the answer
	retries per call	2	then degrade, then discard
S2	question TTS	`gpt-4o-mini-tts`	13 voices $\times$ 5 styles
	answer/recap TTS	`cosyvoice-v3-flash`	2 voices, segment-wise
	retry schedule	8 attempts, exp. backoff	capped at 20 s + jitter
S3	tokenizer	CosyVoice3 (ONNX)	25 Hz, 128-mel front-end
	chunk length	30 s	tail $<$ 1 s merged and re-split
	workers	4 (one per GPU)	residue sharding, dense merge
S4	aligner	Qwen3-ForcedAligner-0.6B	bf16, batch 8, per-item fallback
	silence token set	17 token ids	min run 2, merge distance 8
	fuzzy word search	10 chars	monotone cursor
	fallback boundary	proportional	then monotonicity repair
S5	group size G / delay D	$1 \sim 5$ / $2G$	D enters only the global target length
	marker cost	2 tokens	`<think>` + `</think>`
	budget tokenizer	speech-LLM tokenizer	shared with the compressor
	tightening passes	5	then hard truncation
	workers	16 (4 GPUs)	streaming k-way merge by index

the step span. Subword probabilities are mean-pooled into word probabilities, after which words are selected greedily in decreasing probability order subject to the token budget, with costs measured by the training tokenizer. Greedy selection under an explicit budget attains the target in a single forward pass without iterative recompression.

E.7 Configuration

Table 6 lists the key configured parameters of the pipeline. Two of them deserve comment. The audio group size G and tokenizer rate jointly determine the exchange rate between time and text positions: at $G = 5$ and 25 Hz, one second of speech corresponds to five text positions, so two seconds provide roughly ten positions before subtracting the spoken-text and marker costs. Reducing G increases the number of text positions per second but lengthens the sequence the decoder must produce. The silence-gap parameters control how far a boundary may move from its aligned position; making them stricter yields boundaries closer to the aligner's estimate but places them inside voiced material more often.

F Case Study

This appendix presents complete generation traces for five representative examples, one for each speech frame rate of the speech language model (SLM): 25.0 Hz, 12.5 Hz, 8.33 Hz, 6.25 Hz and 5.00 Hz. For every example we compare three settings — *Baseline* (the standard parallel SLM, which emits the full text stream token by token and pads the remaining slots with silence tokens `<|sil|>`), *Alignment* (thinking interleaved with, and aligned to, the spoken words, but restricted to placeholder reasoning tokens `<|reason|>`), and *ThinkGap* (our method, which explicitly exploits the idle speech–text gaps to perform real reasoning inside `<think>...</think>`). Note that both `<|sil|>` and `<|reason|>` correspond to the generic idle token $\langle\text{pad}\rangle$ defined in Section 3.

For each method we report (i) the *full generated token sequence*, using run-length notation for consecutive identical control tokens, and (ii) the *ASR transcription* of the speech synthesised from that sequence. Special tokens (`<|audio_bos|>`, `<|sil|>`, `<|reason|>`, `<|audio_eos|>`, `<|im_end|>`) and all thinking content are rendered in gray so that the useful linguistic/reasoning payload is easy to separate from the idle padding. Note that gray text is *never* vocalised: it is either a control token or an internal thought. To keep the tables readable without discarding information, a superscript such as $\times 218$ records the exact multiplicity of a consecutive control-token run.

Table 7: Case 1 at a speech frame rate of 25.0 Hz. Repeated control tokens are run-length encoded.

Question. Josh decides to try flipping a house. He buys a house for $80,000 and then puts in $50,000 in repairs. This increased the value of the house by 150%. How much profit did he make? **Ground truth.** `70000`

Method	Generation trace and ASR transcription
Baseline standard parallel SLM	*Generated sequence.* `<|audio_bos|> Let's calculate the profit. The house's new value is two hundred and forty thousand dollars. Subtracting the cost gives a profit of one hundred and fifty thousand. <|sil|>`$^{\times 218}$ `<|audio_eos|> <|im_end|>` *ASR transcription.* "let's calculate the profit the house's new value is two hundred and forty thousand dollars subtracting the cost gives a profit of one hundred and fifty thousand"
Alignment	*Generated sequence.* `<|audio_bos|> Let's calculate the profit. <think> <|reason|>`$^{\times 32}$ `</think> The new value is one hundred twenty thousand. <think> <|reason|>`$^{\times 63}$ `</think> Subtracting the cost gives a profit of seventy thousand. <|sil|>`$^{\times 74}$ `<|audio_eos|> <|im_end|>` *ASR transcription.* "Let's calculate the profit. The new value is 120,000. Subtracting the cost gives a profit of 70,000."
ThinkGap ours; thinking in the speech–text gaps	*Generated sequence.* `<|audio_bos|> Let's calculate the profit. <think> The house increased in value by 80,000*150/100 = $120,000 </think> The value rose by one hundred twenty thousand. <think> So the total value of the house is 80,000+120,000 = $200,000 </think> Total value became two hundred thousand dollars. <think> So the profit is 200,000-80,000-50,000 = $70,000 </think> The answer is seventy thousand dollars profit. <\|sil\|>`$^{\times 143}$ `<|audio_eos|> <|im_end|>` *ASR transcription.* "Let's calculate the profit. The value rose by 120,000. Total value became $200,000. The answer is $70,000 profit."

Table 8: Case 2 at a speech frame rate of 12.5 Hz. Repeated control tokens are run-length encoded.

Question. Haley picked 28 carrots from her garden. If she threw out 11 of them and then picked 9 more the next day, how many carrots would she have total? **Ground truth.** `26`

Method	Generation trace and ASR transcription
Baseline standard parallel SLM	*Generated sequence.* `<|audio_bos|> Let's break this down. She started with twenty-eight and threw out eleven. The answer is twenty-eight. <|sil|>`$^{\times 54}$ `<|audio_eos|> <|im_end|>` *ASR transcription.* "Let's break this down. She started with 28 and threw out 11. The answer is 28."
Alignment	*Generated sequence.* `<|audio_bos|> Let's break this down. <think> <|reason|>`$^{\times 8}$ `</think> She picked thirty-seven carrots the next day. <think> <|reason|>`$^{\times 26}$ `</think> The answer is fifty-six carrots in total. <|sil|>`$^{\times 33}$ `<|audio_eos|> <|im_end|>` *ASR transcription.* "let's break this down she picked 37 carats the next day the answer is 56 carats in total."
ThinkGap ours; thinking in the speech–text gaps	*Generated sequence.* `<|audio_bos|> Let's break this down. <think> 28 11 17 </think> Subtract the eleven she threw out. <think> Haley picked 9 more carrots 17 + 9 = 26 </think> Add nine more to get twenty-six. <|sil|>`$^{\times 31}$ `<|audio_eos|> <|im_end|>` *ASR transcription.* "Let's break this down. Subtract the 11 she threw out. Add 9 more to get 26."

Table 9: Case 3 at a speech frame rate of 8.33 Hz. Repeated control tokens are run-length encoded.

Question. The school cafeteria ordered 42 red apples and 7 green apples for students lunches. But, if only 9 students wanted fruit, how many extra did the cafeteria end up with? **Ground truth.** 40

Method	Generation trace and ASR transcription
Baseline standard parallel SLM	*Generated sequence.* <|audio_bos|> Let's find the extra apples. First, add the red and green apples to get forty-nine. The answer is twenty-nine extra apples. <|sil|>$^{\times 44}$ <|audio_eos|> <|im_end|> *ASR transcription.* "Let's find the extra apples. First, add the red and green apples to get 14ing. The answer is 29 extra apples."
Alignment	*Generated sequence.* <|audio_bos|> Let's see how many extras. <think> <|reason|>$^{\times 10}$ </think> The cafeteria ordered fifty apples in total. <think> <|reason|>$^{\times 19}$ </think> The answer is twenty-five extra apples. <|sil|>$^{\times 20}$ <|audio_eos|> <|im_end|> *ASR transcription.* "Let's see how many extras. The cafeteria ordered 50 apples in total. The answer is 25 extra apples."
ThinkGap ours; thinking in the speech–text gaps	*Generated sequence.* <|audio_bos|> Let's see how many extras there are. <think> 42 </think> First, add the red and green apples to get forty-nine. <think> 9 students 49 - 9 = 40 apples </think> Subtract the nine students, so the answer is forty. <|sil|>$^{\times 26}$ <|audio_eos|> <|im_end|> *ASR transcription.* "Let's see how many extras there are. First, add the red and green apples to get 49. Subtract the 9 students, so the answer is 40."

Table 10: Case 4 at a speech frame rate of 6.25 Hz. Repeated control tokens are run-length encoded.

Question. A waiter at 'The Greasy Spoon' restaurant had 39 customers to wait on. During the lunch rush he added another 12 customers. If 49 of the customers didn't leave him a tip, how many customers did leave a tip? **Ground truth.** 2

Method	Generation trace and ASR transcription
Baseline standard parallel SLM	*Generated sequence.* <|audio_bos|> Let's find the tipping customers. Total customers are fifty-one, so twenty-two left tips. The answer is twenty-two. <|sil|>$^{\times 23}$ <|audio_eos|> <|im_end|> *ASR transcription.* "Let's find the tipping customers. Total customers are fifty-one, so twenty-two left tips. The answer is twenty-two."
Alignment	*Generated sequence.* <|audio_bos|> Let's find the tip-takers. <think> <|reason|>$^{\times 3}$ </think> Subtract the twelve new customers from thirty-nine. <think> <|reason|>$^{\times 10}$ </think> The answer is twenty-eight customers left a tip. <|sil|>$^{\times 13}$ <|audio_eos|> <|im_end|> *ASR transcription.* "Let's find the tip takers. Subtract the 12 new customers from Thurtonine. The answer is 28 customers left a tip."
ThinkGap ours; thinking in the speech–text gaps	*Generated sequence.* <|audio_bos|> Let's find the tip-takers.Add the lunch rush customers to get fifty-one total. <think> 51 - 49 2 </think> Subtract the non-tippers, so the answer is two. <|sil|>$^{\times 14}$ <|audio_eos|> <|im_end|> *ASR transcription.* "Let's find the tip takers. Add the lunch rust customers to get 51 total. Subtract the non-tippers, so the answer is 2."

F.1 Summary and Observations

Standard parallel SLMs waste a large fraction of their text tokens. Across the five cases above, the Baseline emits a total of **353** idle <|sil|> tokens, i.e. roughly **71** wasted text tokens per ex-

Table 11: Case 5 at a speech frame rate of 5.00 Hz. Repeated control tokens are run-length encoded.

Question. Priya needs 54 cupcakes for a birthday party. She already has 15 chocolate cupcakes and 25 vanilla cupcakes. How many more cupcakes should Priya buy? **Ground truth.** 14

Method	Generation trace and ASR transcription
Baseline standard parallel SLM	*Generated sequence.* <|audio_bos|> Let's figure this out. She has forty cupcakes in total. The answer is ten more cupcakes. <|sil|>$^{\times 14}$ <|audio_eos|> <|im_end|> *ASR transcription.* "Let's figure the South. She has 40 cupcakes in total. The answer is 10 more cupcakes."
Alignment	*Generated sequence.* <|audio_bos|> Let's figure this out.She has forty cupcakes in total. <think> <|reason|>$^{\times 3}$ </think> The answer is eleven more cupcakes. <|sil|>$^{\times 9}$ <|audio_eos|> <|im_end|> *ASR transcription.* "Let's figure this out. She has 40 cupcakes in total. The answer is 11 more cupcakes."
ThinkGap ours; thinking in the speech–text gaps	*Generated sequence.* <|audio_bos|> Let's see how many she needs.She has forty cupcakes in total. <think> 54 40 14 </think> The answer is fourteen more cupcakes. <|sil|>$^{\times 9}$ <|audio_eos|> <|im_end|> *ASR transcription.* "Let's see how many she needs. She has 40 cupcakes in total. The answer is 14 more cupcakes."

ample, against only a few dozen tokens that actually carry linguistic content (per-case idle counts, Baseline/Alignment/ThinkGap: 218/74/143; 54/33/31; 44/20/26; 23/13/14; 14/9/9). Because the text stream is forced to advance at the speech frame rate while natural language is far sparser than that, the vast majority of the generation budget is spent on padding: the model pays full attention/decoding cost for tokens that encode no information at all. At 25.0 Hz (Case 1) this degenerates into more than a hundred consecutive silence tokens after a two-sentence answer.

Explicitly exploiting the gap yields clear gains. Alignment replaces part of the padding with <|reason|> placeholders (174 in total), which consumes the idle slots but carries no semantics, so its answers remain unstable and frequently wrong (e.g. *"56 carrots"* instead of 26 in Case 2, *"25 extra apples"* instead of 40 in Case 3, *"28 customers"* instead of 2 in Case 4, and *"eleven more cupcakes"* instead of 14 in Case 5). ThinkGap instead fills the same free slots with real, human-readable reasoning — <think>28 11 17</think>, <think>9 students 49 - 9 = 40 apples</think>, <think>51 - 49 2</think>, <think>54 40 14</think> — at *no additional latency*, since these tokens are produced in parallel with speech and are never vocalised. The cases show that **the surviving tokens are dominated by numerical values, followed by mathematical operators and entity nouns, a distribution broadly consistent with our intended "scratchpad-style" computation pattern.** The arithmetic is therefore completed before the corresponding spoken sentence is uttered. In short, the same budget that the standard parallel SLM spends on silence is turned into effective thinking, and the quality gain is substantial.