Mind-Paced Speaking: A Dual-Brain Approach to Real-Time Reasoning in Spoken Language Models

Anonymous ACL submission

Abstract

Real-time Spoken Language Models (SLMs) struggle to leverage Chain-of-Thought (CoT) reasoning due to the prohibitive latency of generating the entire thought process sequentially. Enabling SLMs to think while speaking, similar to humans, is attracting increasing attention. We present, for the first time, Mind-Paced Speaking (MPS), a brain-inspired framework that enables high-fidelity, real-time reasoning. Similar to how humans utilize distinct brain regions for thinking and responding, we propose a novel dual-brain approach, employing a "Formulation Brain" for high-level reasoning to pace and guide a separate "Articulation Brain" for fluent speech generation. This division of labor eliminates mode-switching, preserving the integrity of the reasoning process. Experiments show that MPS significantly outperforms existing think-while-speaking methods and achieves reasoning performance comparable to models that pre-compute the full CoT before speaking, while drastically reducing latency. Under a zero-latency configuration, the proposed method achieves an accuracy of 92.8% on the mathematical reasoning task Spoken-MQA and attains a score of 82.5 on the speech conversation task URO-Bench. Our work effectively bridges the gap between high-quality reasoning and real-time interaction.

1 Introduction

Speech has emerged as a more natural and fundamental modality for human-computer interaction, leading to growing emphasis on spoken language models (SLMs) (Cui et al., 2024; Wu et al., 2025; Hu et al., 2025; Yu et al., 2024; Défossez et al., 2024). These models facilitate seamless communication by processing and generating audio-based inputs and outputs. A key component enhancing their capability is the integration of thinking, particularly through Chain-of-Thought (CoT) processes and its extensions (Wei et al., 2022; Wang et al., 2023; Yao et al., 2023; Gao et al., 2023), as implemented in frameworks like Think-Before-Speak (TBS) (Wei et al., 2022; Xie et al., 2025a; Guo et al., 2025). This approach enables models to decompose complex tasks into step-by-step reasoning sequences, thereby improving interpretability and performance in dialogue systems.

However, generating complete CoT sequences often introduces significant latency, which hinders real-time applications. Recent efforts to reduce reasoning latency have garnered significant attention (Chiang et al., 2025; Xie et al., 2025c). These methods explore "think-while-speaking" paradigms, where models interleave thinking and response tokens. The Large Language Model (LLM) continuously switches between think and response modes. It first generates several think tokens, then produces several response tokens based on them. These response tokens are sent to the Text To Speech (TTS) system for speech synthesis. While the speech is synthesizing, the LLM continues to generate more think tokens. However, this interleaving disrupts semantic coherence by forcing the model to frequently switch between thinking and response generation, potentially degrading the performance.

In fact, the human brain provides a biological analogy for efficient parallel processing. Cognitive neuroscience reveals that thinking and speaking involve distinct brain areas (Nersessian, 2002; Hickok and Poeppel, 2007). Speech does not follow a rigid "think-then-speak" sequence or interleaved sequence. Crucially, it exhibits an incremental nature where later parts of a thought are still being processed while the initial parts of the utterance are already being spoken (Indefrey, 2011). Inspired by this, we introduce **M**ind-**P**aced **S**peaking (MPS), a novel architecture for enabling SLMs to "think" and "speak" in a concurrent and integrated manner. The core of MPS is a dual-brain framework that operates analogously to the human cognitive-speech system. One LLM acts as a central "Formulation Brain", continuously generating an internal stream of thought. The other functions as an Articulation Brain, which receives this thought stream in segments and generates the corresponding spoken output. The Formulation Brain does not need to complete a full reasoning chain before the Articulation Brain begins. Instead, the ongoing thinking process actively sets the pace and provides the contextual guidance for the Articulation Brain, allowing it to vocalize fluently even as the underlying thoughts are still being formed and refined by the Formulation Brain. This mind-paced mechanism ensures that the spoken output is not only grounded in a thinking process but also maintains semantic coherence, closely mimicking the natural human process of thinking while speaking. Furthermore, we propose a think-incomplete Supervised Fine-Tuning (SFT) method to enable the Articulation Brain to respond based on incomplete thinking content. The experimental results on benchmarks such as mathematical reasoning, dialogue, and question-answering, prove that compared to methods that answer directly without thinking, or existing methods that think while speaking, the proposed MPS method effectively utilizes the thinking process and continuous semantic context, obtaining more accurate and higher-quality responses. Compared to TBS method, the proposed MPS significantly reduces response latency while maintaining performance.

Our main contribution can be summarized as follows:

(1) We propose an MPS architecture that enables SLMs to achieve human-like think-while-speaking capabilities. This method significantly reduces the latency of the CoT process while maintaining the semantic coherence of the LLM. Consequently, the LLM leverages the CoT content to deliver superior performance.

(2) We develop a think-incomplete SFT to train LLMs to generate responses based on partial thinking processes, thereby enabling them to perform think-while-speaking.

(3) We evaluate two distinct MPS architectures: Speak-First and Think-First, against baseline methods. Experimental results demonstrate that the proposed think-while-speaking MPS method significantly outperforms both the direct response approach without a thinking process and existing interleaved think-while-speaking methods. Compared to the TBS architecture, our method substantially reduces response latency while maintaining the quality of the LLM's responses.

(4) Our proposed MPS architecture mimics the neuroscientific mechanisms of human thinking and speaking, transcending the structural limitations of existing interleaved think-while-speaking methods. It provides the research community with a reference paradigm for subsequent studies on anthropomorphic, real-time dialogue systems

2 Related Work

2.1 Spoken Language Models

Spoken Language Models (SLMs) accept user speech input and generate speech output, enabling real-time speech dialogue with users. Since the LLM backbone is typically trained in the text domain, directly generating speech tokens presents a challenge (Chiang et al., 2025). Most current SLMs first generate text tokens and then generate speech. This is achieved through two primary methods: one approach uses the LLM to output text, which is then synthesized into speech by an additional TTS model (Xu et al., 2025; Fang et al., 2025). Another method employs the LLM to generate interleaved text and audio tokens, where each output chunk contains a fixed number of text tokens and speech tokens, and a speech decoder directly synthesizes the speech signal (Zeng et al., 2024; Wu et al., 2025). For example, Step-Audio 2 produces output in the *ta4* format, which means it outputs one text token followed by four audio tokens. This chunk of output is then passed through a speech detokenizer to obtain the speech signal (Wu et al., 2025).

2.2 Reason for SLMs

Although explicit CoT has been proven helpful in text LLMs, most SLMs still lack CoT capability. One reason is that audio and text have different structures; another reason is that directly synthesizing CoT into speech increases the confusion of responses, while generating silent CoT introduces significant latency, which becomes unreasonable in daily conversations. Some studies introduce the reasoning ability into Audio LLMs (Chiang et al., 2025). For example, Xie et al. have proposed an audio CoT reasoning dataset to fine-tune models (Xie et al., 2025b). Some studies use reinforcement learning, such as GRPO (Shao et al., 2024), to fine-tune models and enhance their reasoning ability (Wen et al., 2025; Li et al., 2025). However, these studies remain limited to audio-in-text-out Audio LLMs, not SLMs that can engage in dialogue with humans. In (Wu et al., 2025), Step-Audio 2, which takes speech as input and output, using CoT and reinforcement learning to improve the response qualities, is proposed. Step-Audio 2 offers a solution for introducing explicit reasoning into SLMs. Some methods achieve simultaneous thinking and speaking by segmenting CoT content and response content, using the LLM to generate interleaved think tokens and response tokens (Chiang et al., 2025; Xie et al., 2025c). However, this approach differs from the LLM's original response generation format. The LLM needs to continuously switch between think mode and response mode, which disrupts semantic coherence and affects its performance.

3 Method

This section first outlines the conventional TBS-based SLM. We then present the proposed MPS method. We also introduce the think-incomplete SFT, which is designed to teach LLMs the think-while-speaking capability.

3.1 Think Before Speaking

The architecture of TBS-based SLM is shown in Figure 1. To enhance the reasoning ability of SLM, the TBS paradigm, after receiving user speech $\boldsymbol{X}^{\mathrm{spc}}$ and optional text instructions $\boldsymbol{X}^{\mathrm{txt}}$, first generates step-by-step CoT tokens $\boldsymbol{Y}^{\mathrm{cot}} \in R^{T_c}$, and then generates the response tokens $\boldsymbol{Y}^{\mathrm{res}} \in R^{T_r}$, where T_c and T_r denote the number of CoT tokens and response tokens, respectively. This can be divided into two processes: the thinking process and the speaking process. The thinking process can be written as:

$$P_{\theta_l}(\boldsymbol{Y}^{\mathrm{cot}}|\langle \boldsymbol{X}^{\mathrm{spc}}, \boldsymbol{X}^{\mathrm{txt}}\rangle) = \prod_{t=1}^{T_c} P_{\theta_l}(Y_t^{\mathrm{cot}}|\langle \boldsymbol{Y}_{1:t-1}^{\mathrm{cot}}, \boldsymbol{X}^{\mathrm{spc}}, \boldsymbol{X}^{\mathrm{txt}}\rangle), \tag{1}$$

where θ_l denotes the parameters of the SLM. After that, the LLM generates response tokens for speaking, which can be formulated as:

$$P_{\theta_l}(\boldsymbol{Y}^{\mathrm{res}}|\langle \boldsymbol{Y}^{\mathrm{cot}}, \boldsymbol{X}^{\mathrm{spc}}, \boldsymbol{X}^{\mathrm{txt}}\rangle) = \prod_{t=1}^{T_r} P_{\theta_l}(Y_t^{\mathrm{res}}|\langle \boldsymbol{Y}_{1:t-1}^{\mathrm{res}}, \boldsymbol{Y}_{1:T_c}^{\mathrm{cot}}, \boldsymbol{X}^{\mathrm{spc}}, \boldsymbol{X}^{\mathrm{txt}}\rangle). \tag{2}$$

Through this method, the task is decomposed into a step-by-step process. Additionally, by introducing CoT tokens, it enables more Transformer forward operations and thus gives LLM a deeper inference depth (Goyal et al., 2023; Pfau et al., 2024).

3.2 Architecture

In the human brain, speech production is not a monolithic process but the result of two highly specialized and collaborative systems. The first, a network centered around the prefrontal-temporal cortex, is responsible for high-level cognitive functions such as conceptualization, logical reasoning, and content planning. Subsequently, a second system, primarily involving the motor cortex and subcortical pathways, translates these abstract thoughts into natural language for articulation, enabling fluent speech. These two systems operate in parallel, with the cognitive system continuously supplying thinking content to the articulatory system, creating a natural flow where the mind paces speech (Nersessian, 2002; Hickok and Poeppel, 2007).

Inspired by this, we abstract this mechanism of separated "formulation" and "articulation" into our model architecture. Instead of relying on a single LLM to handle both thinking and speaking, we propose a dual-brain system composed of two distinct LLMs.

Our proposed framework, illustrated in Figure 2, leverages a dual-LLM architecture consisting of a Formulation Brain LLM and an Articulation Brain LLM. The Formulation Brain LLM is dedicated to user intent understanding and performs deliberate CoT reasoning, with its internal process materialized as "think tokens". Subsequently, the Articulation Brain LLM converts this structured reasoning and the dialogue context into natural language, producing the final "response tokens" for spoken output.

Formulation Brain: The Formulation Brain's operating mode is identical to that of TBS Audio LLMs but with only the thinking process. After receiving user input $\boldsymbol{X}^{\mathrm{spc}}$ and $\boldsymbol{X}^{\mathrm{txt}}$, it aims to generate the step-by-step CoT tokens $\boldsymbol{Y}^{\mathrm{cot}} \in R^{T_c}$. We use the tokens <think> and </think> to mark the beginning and end of the CoT. This process can be formulated as (1). In the MPS architecture, we do not wait for the Formulation Brain to complete the entire CoT before the Articulation Brain starts speaking. We divide the CoT tokens $\boldsymbol{Y}^{\mathrm{cot}}$ into N segments, denoted as $[\boldsymbol{S}_1^{\mathrm{cot}}, \boldsymbol{S}_2^{\mathrm{cot}}, ..., \boldsymbol{S}_N^{\mathrm{cot}}]$. Each time the Formulation Brain produces a think seg-

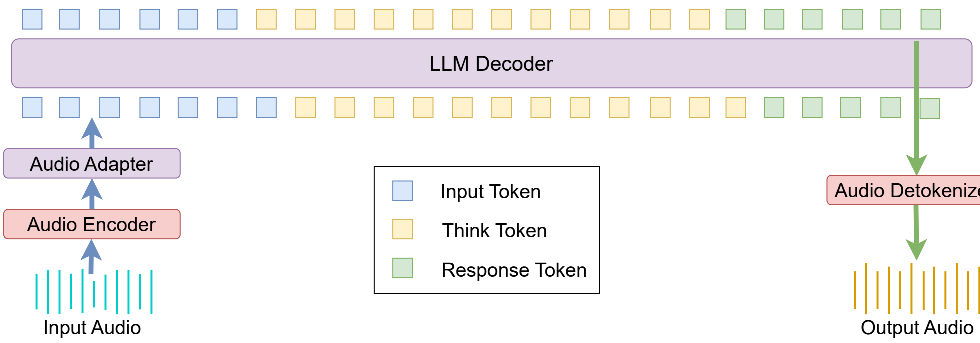

Figure 1: Architecture of the TBS architecture. For the sake of conciseness, we remove the input text, which is optional in SLMs. The TBS SLM first generates the full CoT and then produces response tokens.

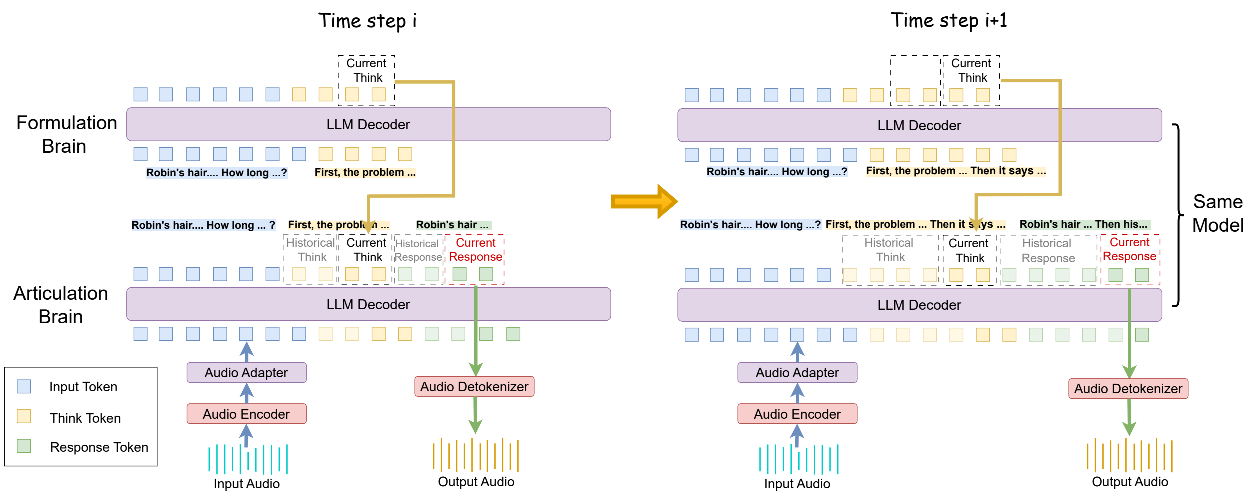

Figure 2: Architecture of the proposed MPS. For the sake of conciseness, we remove the input text, which is optional in SLMs. We demonstrate the process from step i to step i+1 when generating think segments and response segments. The Formulation Brain LLM continuously generates the think segments. The newly generated think segment and the response segment from the previous step are both added as the prefix to the Articulation Brain LLM, pacing the Articulation Brain LLM to produce response segment correspondingly.

ment S_n^{cot}, we feed the segment to the Articulation Brain, which then generates a response segment based on the current think segment and historical thinking and response contents. After the Formulation Brain LLM finishes CoT segments, it stops generating response tokens as we do not require the Formulation Brain to speak.

Articulation Brain: The Articulation Brain accepts the same user input as the Formulation Brain. After obtaining the current think segment S_n^{cot} from the Formulation Brain, we concatenate it with the historical think segments $[S_1^{\text{cot}}, S_2^{\text{cot}}, ..., S_{n-1}^{\text{cot}}]$, placing <think> and </think> at the beginning and the end, and then append the historical response segments, which are defined as $[S_1^{\text{res}}, S_2^{\text{res}}, ..., S_{n-1}^{\text{res}}]$. This allows the Articulation Brain to continue generating the subsequent response content. After that, we use a streaming TTS model to synthesize speech in real-time. The Articulation Brain's output is incremental. For every think segment that the Formulation Brain produces, the Articulation Brain generates a segment of the response S_n^{res}. This process can be written as:

$$P_{\theta_l}(S^{\text{res}} | \langle S^{\text{cot}}, X^{\text{spc}}, X^{\text{txt}} \rangle) = \prod_{n=1}^{N} P_{\theta_l}(S_n^{\text{res}} | \langle S_{1:n-1}^{\text{res}}, S_{1:n}^{\text{cot}}, X^{\text{spc}}, X^{\text{txt}} \rangle). \tag{3}$$

When the Formulation Brain just begins its thinking, the Articulation Brain can only generate a response segment based on a small amount of CoT. The response segment it generates at this stage may be of lower quality. As the Formulation Brain's thinking content increases, the Articulation Brain receives more CoT content, and it subsequently generates responses of increasingly higher quality.

Compared with existing think-while-speaking methods that use a single LLM to predict interleaved think and response tokens, thereby forcibly interrupting and splitting the originally continuous think and response content (Chiang et al., 2025; Xie et al., 2025c), our method adopts a dual-brain design consisting of the Formulation Brain and the Articulation Brain. From the perspectives of the Formulation Brain and the Articulation Brain, both are classic TBS LLMs that, after receiving user input, first generate step-by-step CoT content and then generate response content conditioned on the CoT, thereby greatly ensuring the semantic coherence of the LLM output. By allowing the Formulation Brain to pace the Articulation Brain, our method achieves a human-like think-while-speaking process.

3.3 Think-incomplete SFT

Since the proposed MPS method does not change the input-output patterns of the classic LLM for the individual Formulation Brain and Articulation Brain, the proposed MPS, unlike existing think-while-listening methods (Chiang et al., 2025; Xie et al., 2025c), does not require repretraining the LLM. To ensure that the Articulation Brain LLM possesses the ability to accept incomplete think content and produce reasonable output, we introduce think-incomplete SFT. In the construction of training data, we randomly retain the content of the first L steps of the step-by-step CoT, delete the subsequent CoT content, then place this incomplete CoT with <think> and </think> tokens at the beginning and end, concatenate it with the groundtruth response, and use it as the next-token-prediction training objective for the LLM (Brown et al., 2020).

During the inference stage, we use segments with a fixed number of tokens. We set T_c and T_r to 80 and 100 respectively. We use the output format of Step-Audio 2, specifically the *ta4* format, which generates one text token followed by four speech tokens, thus every 100 response tokens contain 20 text tokens and 80 speech tokens. The speech synthesis frame rate is 12.5 Hz, meaning that 80 tokens correspond to 6.4 seconds of speech. This duration is fully sufficient for the LLM to generate 80 think tokens, thereby ensuring the continuous and uninterrupted speech generation of our proposed method. We also attempt to use the same segment division strategy as in the think-incomplete SFT phase, but we find that it does not bring improvement; on the contrary, it introduces uncontrollable latency due to the variable length of each CoT step. We also try using a fixed token count strategy for dividing the CoT during the think-incomplete SFT phase, but it does not yield performance improvements either.

4 Experiments

4.1 Experimental Settings

The LLM backbone used in this paper is Step-Audio 2, and its parameter settings refer to (Wu et al., 2025). The LLMs in Formulation Brain and Articulation brain share the same parameters. To verify the effectiveness of the proposed method on tasks requiring reasoning, we use Spoken-MQA, a mathematical reasoning dataset (Wei et al., 2025). We use accuracy as the evaluation metric. Furthermore, to validate the method's effectiveness on general dialogue tasks, we introduce URO-Bench, which contains several subtasks such as daily dialogue, emotion recognition, paralinguistic information, and question-answering (Yan et al., 2025). For question-answering tasks, we use accuracy as the metric. For other tasks, we use GPT-score, generated by `GPT-4o-mini` and ranging from 0 to 100, to evaluate response quality.

To accommodate the latency requirements of different application scenarios, we implement two distinct MPS paradigms:

- Think-First, denoted as **MPS-thkfirst**: The Formulation Brain LLM first generates T_c think tokens, after which the Articulation Brain LLM generates T_r response tokens and synthesizes speech. Under this setting, the latency is T_c plus the buffer size of streaming TTS, which is significantly lower than the latency required for the TBS structure to generate a complete Chain-of-Thought.

- Speak-First, denoted as **MPS-spkfirst**: The Articulation Brain LLM first generates T_r response tokens, while simultaneously, the Formulation Brain LLM begins generating think tokens. The Formulation Brain LLM completes generating T_c think tokens before the speech synthesized from the T_r response tokens finishes playing. In this configuration, the latency is solely the buffer size of the

Table 1: Test-set accuracy (%) of different methods on the Spoken-MQA benchmark. The evaluated approaches include: the direct response baseline without a thinking process (MPS-wo/*thk*), Think-Before-Speaking (MPS-tbs), Think-First (MPS-thkfirst), and Speak-First (MPS-spkfirst). Results of baseline systems are taken from (Xie et al., 2025c) except that results of Step-Audio 2 are reproduced by ourselves.

Method	Arithmetic			Reasoning			Avg
	Short	Long	Avg	Single	Multi	Avg	
Whisper-Qwen2.5-7B-Instruct	-	-	70.0	-	-	72.5	72.2
Whisper-Qwen2.5-Math-7B-Instruct	-	-	77.3	-	-	86.7	85.6
LLaMA-Omni	40.0	11.0	23.5	29.5	10.5	16.2	16.8
Mini-Omni	5.0	2.3	3.5	0.8	1.9	1.6	1.7
Freeze-omni	43.0	14.5	26.8	69.0	19.8	34.4	33.3
GLM-4-Voice	40.0	22.5	30.1	54.4	28.5	36.2	35.3
Qwen2-Audio-7B-Instruct	43.0	31.2	36.3	55.4	22.5	32.3	32.7
Qwen2.5-Omni-7B	83.0	45.1	61.5	85.2	71.5	75.6	73.6
Qwen2.5-Omni-3B	84.0	43.3	60.1	81.5	57.1	64.4	63.6
Mini-Omni-Reasoner	**92.9**	66.1	77.3	85.9	60.5	68.1	68.6
Step-Audio 2	89.0	52.6	65.9	95.6	90.4	91.9	88.8
MPS-wo/*thk*	71.0	34.1	47.6	88.0	67.8	73.8	70.6
MPS-tbs	90.0	**88.4**	**89.0**	94.4	93.2	93.6	93.0
MPS-thkfirst	89.0	84.9	86.4	95.6	**94.6**	**94.9**	**93.9**
MPS-spkfirst	87.0	71.7	77.3	**96.0**	94.5	**94.9**	92.8

Table 2: The average accuracy of different models with CoT capability on Spoken-MQA, and the extra tokens generated by the model before generating the first response token. The evaluated approaches include: Interleaved Think-While-Speaking (Mini-Omni-Reasoner), Think-Before-Speaking (MPS-tbs), Think-First (MPS-thkfirst), and Speak-First (MPS-spkfirst).

Method	Accuracy	Extra Tokens
Mini-Omni-Reasoner	68.6%	8
MPS-tbs	93.0%	762
MPS-thkfirst	**93.9%**	80
MPS-spkfirst	92.8%	**0**

streaming TTS, meaning the model can be considered to respond directly with near-zero latency.

Additionally, we compare the proposed method with two approaches that use the same LLM backbone as in this paper: Think-Before-Speaking (MPS-tbs) and direct response without thinking (MPS-wo/*thk*), to validate the effectiveness of our proposed think-while-speaking methodology. All results reported are from a single run with the random seed fixed to 42.

4.2 Data Construction

We begin with real-world user queries as our seed set. To ensure topical diversity and sufficient scale, we employ `GPT-4o` (OpenAI, 2024) for transcription and augmentation of these queries. These augmented queries are then used as user prompts to distill dialogue data with native CoT from the `DeepSeek-R1` model (Guo et al., 2025). Finally, the speech is synthesized by `Step-Audio-TTS-3B` (et al., 2025).

However, the raw data generated by `DeepSeek-R1`, a text-centric model, presents two critical challenges for spoken dialogue applications: (1) Text-specific stylizations, such as Markdown formatting and emojis, which are incompatible with speech synthesis. (2) The CoT data reflects complete, turn-based reasoning chains, a format unsuitable for training the model to respond from partial thoughts. When the CoT generated by the LLM exhibits some incomplete, its performance is affected.

To address these challenges, we implement a

Table 3: Performance of different methods on the URO-Bench. The evaluated approaches include: the direct response baseline without a thinking process (MPS-wo/*thk*), Think-Before-Speaking (MPS-tbs), Think-First (MPS-thkfirst), and Speak-First (MPS-spkfirst). Results of baseline systems are taken from (Wu et al., 2025). The results of *Multilingual* of URO-Bench are included in *English*.

Method	Language	Basic				Pro			
		U.	R.	O.	Avg	U.	R.	O.	Avg
GPT-4o Audio	Chinese	89.4	65.5	85.2	78.6	70.6	57.2	70.2	67.1
GPT-Realtime	Chinese	88.8	72.9	90.8	80.6	72.3	62.6	74.2	70.6
Kimi-Audio	Chinese	79.3	64.7	79.8	73.6	60.4	59.3	76.2	66.0
Qwen-Omni	Chinese	59.7	69.7	77.3	69.0	59.0	59.8	58.7	59.1
Step-Audio 2	Chinese	91.1	75.5	86.1	83.3	74.8	63.2	65.1	68.3
MPS-wo/*thk*	Chinese	91.6	77.3	87.7	83.4	75.1	74.7	72.9	74.4
MPS-tbs	Chinese	92.6	82.4	93.8	87.8	75.3	84.2	79.5	79.0
MPS-thkfirst	Chinese	**93.6**	**84.0**	**94.8**	**89.1**	75.2	84.2	**85.2**	**80.5**
MPS-spkfirst	Chinese	92.5	82.5	93.1	87.6	**77.2**	**84.8**	79.0	79.9
GPT-4o Audio	English	90.2	75.9	90.4	84.5	60.7	64.4	78.5	67.5
GPT-Realtime	English	87.4	**84.1**	**94.1**	**88.1**	59.7	74.5	76.1	68.9
Kimi-Audio	English	83.4	42.3	60.4	60.0	50.3	40.6	56.0	49.8
Qwen-Omni	English	66.3	69.6	76.2	70.6	44.5	63.9	49.4	51.0
Step-Audio 2	English	92.7	76.5	84.9	83.9	64.9	67.8	66.3	66.1
MPS-wo/*thk*	English	91.5	68.7	78.8	77.4	73.4	79.2	55.8	65.1
MPS-tbs	English	92.3	81.5	87.5	86.1	76.4	86.4	87.1	83.3
MPS-thkfirst	English	**94.2**	81.4	89.0	87.0	**76.5**	89.3	**89.4**	**85.0**
MPS-spkfirst	English	94.1	78.5	87.5	85.2	76.0	**89.7**	69.9	74.8

fine-grained data processing pipeline:

- **Compatibility Processing:** We discard samples containing Markdown formatting or multi-item lists that cannot be naturally rendered in speech. For samples containing emojis, we employ `Qwen-72B-Instruct` (et al., 2024) to remove these elements while preserving the plain text content.

- **CoT Pruning:** To train the model to respond stably with only partial CoT, we augment the data by randomly deleting some reasoning paragraphs. This operation is performed in a way that generally preserves the overall logic of the CoT. To maintain the stylistic distribution of the original DeepSeek CoT, we neither delete individual sentences within a paragraph nor use an LLM to rewrite the content of the remaining parts. This ensures that the preserved paragraphs are stylistically and distributionally consistent with the source model.

4.3 Results

4.3.1 Evaluation on Reasoning Tasks

Table 1 shows the computational accuracy on Spoken-MQA. It can be seen that the proposed MPS-thkfirst method exceeds the MPS-wo/thk method and all baseline methods, including the think-while-speaking Mini-Omni-Reasoner, in all evaluation tasks. The results proves that the proposed method effectively utilizes the thinking process, achieving more intelligent response. Compared to Mini-Omni-Reasoner, the MPS method maintains semantic coherence and achieves better performance.

Besides, compared to MPS-tbs, the MPS-thkfirst method demonstrates comparable performance, being slightly weaker in arithmetic computation tasks but superior in reasoning tasks. One possible explanation is that reasoning tasks require more textual analysis. The MPS-thkfirst method, by using each think segment to pace the generation of a corresponding response segment, implicitly achieves semantic alignment, enabling the model to better utilize contextual information for response generation.

The MPS-spkfirst method experiences some performance degradation because it initially outputs a response segment without utilizing any think segments. This impact is particularly pronounced in tasks involving direct arithmetic computation. For reasoning tasks, experimental observations indicate that the initial phase of the LLM's CoT content primarily involves analyzing the semantic information of the question, often rewriting the question's content. Consequently, this initial portion of the CoT content has a limited effect on the final response. As a result, MPS-spkfirst is minimally affected in reasoning tasks, and its performance remains nearly identical to that of MPS-thkfirst. Experimental results on Spoken-MQA demonstrate that the proposed method significantly leverages CoT to achieve more intelligent responses. Furthermore, compared to TBS methods, our think-while-speaking approach achieves comparable performance, with significantly lower CoT latency as analysed in Section 4.1.

To demonstrate the latency differences between the proposed method and baseline approaches, we select four models from Table 1 that include a thinking process: Mini-Omni-Reasoner, MPS-tbs, MPS-thkfirst, and MPS-spkfirst. We calculate the number of extra tokens generated by the model from the end of the user's question to the generation of the first response token on Spoken-MQA. The results are shown in Table 2. It can be observed that compared to MPS-tbs, MPS-thkfirst achieves higher accuracy while exhibiting significantly lower response latency. Although the accuracy of MPS-spkfirst is slightly lower than that of MPS-tbs and MPS-thkfirst, its response is without latency. Furthermore, compared to Mini-Omni-Reasoner, which uses interleaved think and response tokens to achieve think-while-speaking, the proposed MPS methods achieve higher accuracy. Notably, the MPS-spkfirst attains this superior accuracy with zero latency. This indicates that MPS-spkfirst can play a more critical role in real-time dialogue scenarios with low-latency requirements. To further demonstrate how the Formulation Brain drives the Articulation Brain to realize the process of thinking while speaking, we present and describe an example of MPS-spkfirst on the SpokenMQA dataset in Appendix B.

4.3.2 Evaluation on Speech-to-speech Conversation

Table 3 shows the results of different methods on URO-Bench. It can be observed that MPS-thkfirst achieves higher performance than MPS-tbs on nearly all tasks and on average, under lower response latency. This may also be related to the implicit semantic alignment performed by MPS-thkfirst. Due to generating an initial response segment without prior thinking, MPS-spkfirst performs slightly worse than MPS-thkfirst, but still significantly outperforms the direct response method MPS-wo/*thk*. It can also be observed that this initial response segment without thinking, primarily affects the performance of MPS-spkfirst on **O**ral Conversation within the *Pro* dataset. In contrast, the performance on **U**nderstanding, **R**easoning, and the subtasks in all the *Basic* dataset is less affected. Nevertheless, MPS-spkfirst features lowest response latency as analysed in Section 4.1, and its response performance remains close to or even better than that of MPS-tbs in some tasks, making it more suitable for scenarios requiring faster feedback. The experimental results demonstrate that the proposed MPS method maintains high performance on dialogue tasks, achieving performance comparable to TBS models while operating at significantly lower latency.

5 Conclusion

This paper proposes the MPS method, which enables SLMs to possess the ability to think while speaking. Inspired by the human thinking and response mechanism, we use a Formulation Brain LLM to continuously generate think segments, pacing the Articulation Brain LLM to utilize historical and current think segments, as well as historical responses, to generate current response segment. Experimental results on mathematical reasoning and speech conversation tasks show that the proposed method significantly outperforms direct response methods and existing think-while-speaking methods. It achieves performance comparable or even better than methods that complete thinking before responding, while greatly reducing response latency. The proposed method breaks through the limitations of existing interleaved thinking and response-based think-while-speaking methods and provides an effective reference for researching real-time dialogue consistent with human thinking and response mechanisms.

Limitations

Despite the breakthroughs achieved in real-time reasoning for SLMs and in enabling LLMs to "think while listening" in a human-like manner, our work presents several limitations:

- Resource Consumption: Our method requires two LLMs for inference. While the utilization of the prefix technique ensures that the total number of inference steps remains same to a single LLM, it doubles the memory usage for two LLMs. Additionally, the dual-model architecture introduces communication overhead between the components.
- Hyperparameter Optimization: The number of tokens assigned to the think and response segments is determined based on empirical results and theoretical time overhead calculations. However, the theoretical optimum for these configurations remains to be fully explored.
- Generalizability: Our investigation is conducted exclusively using the Step-Audio LLM backbone. The generalizability of our approach across other LLM architectures remains to be verified through further experiments.

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Cheng-Han Chiang, Xiaofei Wang, Linjie Li, Chung-Ching Lin, Kevin Lin, Shujie Liu, Zhendong Wang, Zhengyuan Yang, Hung-yi Lee, and Lijuan Wang. 2025. Stitch: Simultaneous thinking and talking with chunked reasoning for spoken language models. *arXiv preprint arXiv:2507.15375*.

Wenqian Cui, Dianzhi Yu, Xiaoqi Jiao, Ziqiao Meng, Guangyan Zhang, Qichao Wang, Yiwen Guo, and Irwin King. 2024. Recent advances in speech language models: A survey. *arXiv preprint arXiv:2410.03751*.

Alexandre Défossez, Laurent Mazaré, Manu Orsini, Amélie Royer, Patrick Pérez, Hervé Jégou, Edouard Grave, and Neil Zeghidour. 2024. Moshi: a speech-text foundation model for real-time dialogue. *arXiv preprint arXiv:2410.00037*.

Ailin Huang et al. 2025. Step-audio: Unified understanding and generation in intelligent speech interaction. *Preprint*, arXiv:2502.11946.

An Yang et al. 2024. Qwen2 technical report. *Preprint*, arXiv:2407.10671.

Qingkai Fang, Yan Zhou, Shoutao Guo, Shaolei Zhang, and Yang Feng. 2025. Llama-omni2: Llm-based real-time spoken chatbot with autoregressive streaming speech synthesis. *arXiv preprint arXiv:2505.02625*.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Pal: Program-aided language models. In *International Conference on Machine Learning*, pages 10764–10799. PMLR.

Sachin Goyal, Ziwei Ji, Ankit Singh Rawat, Aditya Krishna Menon, Sanjiv Kumar, and Vaishnavh Nagarajan. 2023. Think before you speak: Training language models with pause tokens. *arXiv preprint arXiv:2310.02226*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Gregory Hickok and David Poeppel. 2007. The cortical organization of speech processing. *Nature reviews neuroscience*, 8(5):393–402.

Ke Hu, Ehsan Hosseini-Asl, Chen Chen, Edresson Casanova, Subhankar Ghosh, Piotr Żelasko, Zhehuai Chen, Jason Li, Jagadeesh Balam, and Boris Ginsburg. 2025. Efficient and direct duplex modeling for speech-to-speech language model. *arXiv preprint arXiv:2505.15670*.

Peter Indefrey. 2011. The spatial and temporal signatures of word production components: a critical update. *Frontiers in psychology*, 2:255.

Gang Li, Jizhong Liu, Heinrich Dinkel, Yadong Niu, Junbo Zhang, and Jian Luan. 2025. Reinforcement learning outperforms supervised fine-tuning: A case study on audio question answering. *arXiv preprint arXiv:2503.11197*.

Nancy J Nersessian. 2002. *The cognitive basis of model-based reasoning in science*. na.

OpenAI. 2024. Gpt-4o system card. *Preprint*, arXiv:2410.21276.

Jacob Pfau, William Merrill, and Samuel R Bowman. 2024. Let's think dot by dot: Hidden computation in transformer language models. *arXiv preprint arXiv:2404.15758*.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024.

Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

Chengwei Wei, Bin Wang, Jung-jae Kim, and Nancy F Chen. 2025. Towards spoken mathematical reasoning: Benchmarking speech-based models over multi-faceted math problems. *arXiv preprint arXiv:2505.15000*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Cheng Wen, Tingwei Guo, Shuaijiang Zhao, Wei Zou, and Xiangang Li. 2025. Sari: Structured audio reasoning via curriculum-guided reinforcement learning. *arXiv preprint arXiv:2504.15900*.

Boyong Wu, Chao Yan, Chen Hu, Cheng Yi, Chengli Feng, Fei Tian, Feiyu Shen, Gang Yu, Haoyang Zhang, Jingbei Li, and 1 others. 2025. Step-audio 2 technical report. *arXiv preprint arXiv:2507.16632*.

Jingran Xie, Shun Lei, Yue Yu, Yang Xiang, Hui Wang, Xixin Wu, and Zhiyong Wu. 2025a. Leveraging chain of thought towards empathetic spoken dialogue without corresponding question-answering data. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Zhifei Xie, Mingbao Lin, Zihang Liu, Pengcheng Wu, Shuicheng Yan, and Chunyan Miao. 2025b. Audio-reasoner: Improving reasoning capability in large audio language models. *arXiv preprint arXiv:2503.02318*.

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

Jin Xu, Zhifang Guo, Jinzheng He, Hangrui Hu, Ting He, Shuai Bai, Keqin Chen, Jialin Wang, Yang Fan, Kai Dang, and 1 others. 2025. Qwen2. 5-omni technical report. *arXiv preprint arXiv:2503.20215*.

Ruiqi Yan, Xiquan Li, Wenxi Chen, Zhikang Niu, Chen Yang, Ziyang Ma, Kai Yu, and Xie Chen. 2025. Uro-bench: A comprehensive benchmark for end-to-end spoken dialogue models. *arXiv preprint arXiv:2502.17810*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822.

Wenyi Yu, Siyin Wang, Xiaoyu Yang, Xianzhao Chen, Xiaohai Tian, Jun Zhang, Guangzhi Sun, Lu Lu, Yuxuan Wang, and Chao Zhang. 2024. Salmonn-omni: A codec-free llm for full-duplex speech understanding and generation. *arXiv preprint arXiv:2411.18138*.

Aohan Zeng, Zhengxiao Du, Mingdao Liu, Kedong Wang, Shengmin Jiang, Lei Zhao, Yuxiao Dong, and Jie Tang. 2024. Glm-4-voice: Towards intelligent and human-like end-to-end spoken chatbot. *arXiv preprint arXiv:2412.02612*.

A Example of MPS Results

Figure 3 shows an example of MPS-spkfirst on Spoken-MQA. After receiving the user input, the Articulation Brain LLM first generates a response segment $\boldsymbol{S}_1^{res}$. Simultaneously, the Formulation Brain LLM produces the first think segment $\boldsymbol{S}_1^{thk}$. $\boldsymbol{S}_1^{thk}$ is then prefixed to the Articulation Brain LLM along with $\boldsymbol{S}_1^{res}$ to pace the Articulation Brain LLM in generating the second response segment $\boldsymbol{S}_2^{res}$. During this period, the Formulation Brain LLM generates the second think segment $\boldsymbol{S}_2^{thk}$. $\boldsymbol{S}_2^{thk}$ is further prefixed to the Articulation Brain LLM, where $\boldsymbol{S}_1^{thk}$, $\boldsymbol{S}_2^{thk}$, $\boldsymbol{S}_1^{res}$, and $\boldsymbol{S}_2^{res}$ collectively pace the Articulation Brain LLM to produce $\boldsymbol{S}_3^{res}$. This process repeats until the Formulation Brain LLM generates the complete think content, after which the Articulation Brain LLM continues generating content until completion.

B Ethical Considerations

We utilize the publicly available SpokenMQA and URO-bench dataset for testing. The training speech data is synthesized by Step-Audio-TTS-3B, based on the text dialogue data generated by GPT-4o, DeepSeek-R1 and Qwen-72B-Instruct. We have reviewed the dataset documentation which confirms that personal identifiers have been anonymized. We did not attempt to de-anonymize any speakers.

C LLM Usage Statement

We utilized ChatGPT exclusively for grammatical corrections and stylistic polishing to enhance clarity and conciseness. We confirm that the conceptualization, methodology, and data analysis are entirely the original work of the authors.

Question:
"Robin's hair was 16 inches long. He cut off 11 inches. Then it again grew by 12 inches. How long is his hair now?"
Groundtruth:
"17"
Response:
Step1:
Formulation Brain:
""
Articulation Brain:
"Robin's hair started at 16 inches. He cut off 11"

Step1:
Formulation Brain:
"First, the problem states that Robin's hair was initially 16 inches long. He cut off 11 inches. So, after cutting, the length would be 16 inches minus 11 inches. That gives 5 inches. Then"
Articulation Brain:
"Robin's hair started at 16 inches. He cut off 11 inches, so 16 minus 11 is 5 inches. Then, his "

Step2:
Formulation Brain:
"First, the problem states that Robin's hair was initially 16 inches long. He cut off 11 inches. So, after cutting, the length would be 16 inches minus 11 inches. That gives 5 inches. Then, it says his hair grew by 12 inches. So, starting from the 5 inches after cutting, adding 12 inches would be 5 + 12 = 17 inches. The problem says "
Articulation Brain:
"Robin's hair started at 16 inches. He cut off 11 inches, so 16 minus 11 is 5 inches. Then, his hair grew by 12 inches, so 5 plus 12 equals "

Step3:
Formulation Brain:
"First, the problem states that Robin's hair was initially 16 inches long. He cut off 11 inches. So, after cutting, the length would be 16 inches minus 11 inches. That gives 5 inches. Then, it says his hair grew by 12 inches. So, starting from the 5 inches after cutting, adding 12 inches would be 5 + 12 = 17 inches. The problem says \"he cut off 11 inches,\" which clearly indicates cutting, so subtracting is correct. So, the final answer should be 17 inches."
Articulation Brain:
"Robin's hair started at 16 inches. He cut off 11 inches, so 16 minus 11 is 5 inches. Then, his hair grew by 12 inches, so 5 plus 12 equals 17 inches. Therefore, Robin's hair is now 17 inches long. "

Figure 3: An example of the output of MPS-spkfirst on the Spoken-MQA dataset. The Articulation Brain first generates a response segment. Simutaneously, Formulation Brain continuously generates new think segments, and each newly generated think segment is prefixed to the Articulation Brain, pacing it to generate new response segment.