Chronological Thinking in Full-Duplex Spoken Dialogue Language Models

Donghang Wu[1,2,*] **Haoyang Zhang**[1,2,*] **Chen Chen**[1,†] **Tianyu Zhang**[3] **Fei Tian**[2]
Xuerui Yang[2] **Gang Yu**[2] **Hexin Liu**[1] **Nana Hou**[1] **Yuchen Hu**[1] **Eng Siong Chng**[1,†]
[1]Nanyang Technological University [2]StepFun [3]Mila
{chen1436, ASESChng}@ntu.edu.sg

Abstract

Recent advances in spoken dialogue language models (SDLMs) reflect growing interest in shifting from turn-based to full-duplex systems, where the models continuously perceive user speech streams while generating responses. This simultaneous listening and speaking design enables real-time interaction and the agent can handle dynamic conversational behaviors like user barge-in. However, during the listening phase, existing systems keep the agent idle by repeatedly predicting the silence token, which departs from human behavior: we usually engage in lightweight thinking during conversation rather than remaining absent-minded. Inspired by this, we propose *Chronological Thinking*, an on-the-fly conversational thinking mechanism that aims to improve response quality in full-duplex SDLMs. Specifically, chronological thinking presents a paradigm shift from conventional LLM thinking approaches, such as Chain-of-Thought, purpose-built for streaming acoustic input. (1) *Strictly causal*: the agent reasons incrementally while listening, updating internal hypotheses only from past audio with no lookahead. (2) *No additional latency*: reasoning is amortized during the listening window; once the user stops speaking, the agent halts thinking and begins speaking without further delay. Experiments demonstrate the effectiveness of chronological thinking through both objective metrics and human evaluations show consistent improvements in response quality. Furthermore, chronological thinking robustly handles conversational dynamics and attains competitive performance on full-duplex interaction metrics.

1 Introduction

Speech is a natural and fundamental modality for human–computer interaction, offering intuitive, efficient, and expressive communication (Cui et al., 2024; Huang et al., 2025). Reflecting this importance, spoken dialogue language models (SDLMs) have become increasingly central in AI as advanced systems seek to support natural interaction. In academia, SDLMs remain an active area of research (Nguyen et al., 2023; Hu et al., 2025; Ding et al., 2025), with a growing emphasis on end-to-end speech-to-speech dialogue systems that integrate speech understanding and generation within a unified interactive loop.

More recently, full-duplex models have garnered significant attention as a novel SDLM architecture (Défossez et al., 2024; Chen et al., 2025c,b), departing from traditional turn-based interaction by removing rigid listen-then-speak alternation (Veluri et al., 2024; Lin et al., 2022; Liao et al., 2025), as illustrated in Figure 1. In a full-duplex system, the model continually ingests streaming user speech while synthesizing the corresponding response in real time. This "always-on" agent delivers more natural, fluid, and human-like conversations, with the ability to proactively take turns, offer backchannel responses, make timely corrections, and gracefully yield with user barges-in (Défossez et al., 2024).

However, despite active exploration of full-duplex models, we identify a common issue across existing designs: the agent is kept idle during user speaking by repeatedly predicting a "silence token". This practice is problematic for two reasons: (1) in autoregressive models, prolonged repetition of a single token can be harmful, biasing the next-token distribution and reinforcing degeneracy (Zhu et al., 2023; Guan and Huang, 2023; Xu et al., 2022); and (2) it leaves the listening window underutilized, missing opportunities to form intent hypotheses and organize the forthcoming response. In contrast, human listeners perform lightweight, conversational thinking while listening—continually updating beliefs about the speaker's intent and sketching

[*]Equal contribution
[†]Corresponding authors

Proceedings of the 27th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 473–485
August 2–5, 2026.

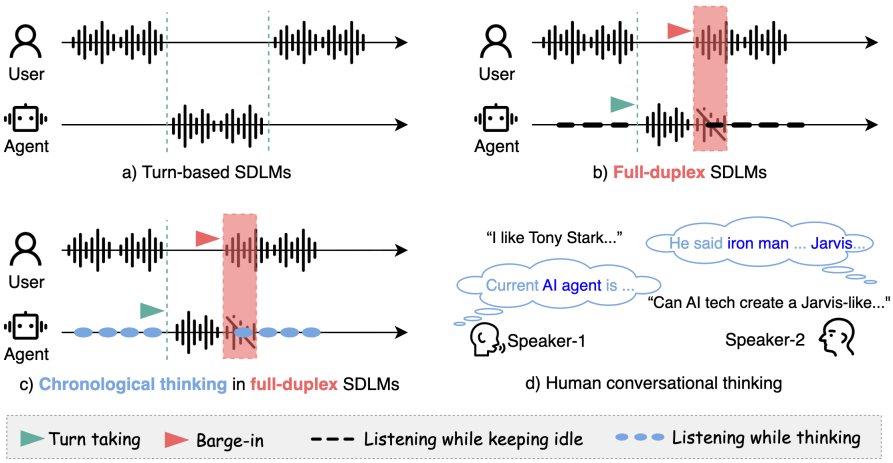

Figure 1: Comparison of a) turn-based SDLMs, b) full-duplex SDLMs c) chronological thinking in full-duplex SDLMs (ours) and d) human conversational thinking patterns.

response structure, as shown in Figure 1. This observation raises a central research question: *Can such on-the-fly thinking be feasible within full-duplex SDLMs?*

Notwithstanding the above, implementing such a mechanism under full-duplex constraints is particularly challenging. First, streaming user speech imposes strict causality: the agent must reason incrementally while listening, updating hypotheses only from past audio without lookahead or access to a complete utterance (Veluri et al., 2024). Second, since user speech can end at any moment, the reasoning process must be preemptible and amortized during listening; once the user stops, the agent should transition to speaking immediately without incurring additional latency (Chiba and Higashinaka, 2025). However, existing "thinking" techniques in LLMs, such as Chain-of-Thought (CoT) (Wei et al., 2022), are typically lengthy and post hoc, and therefore not directly applicable to the full-duplex setting. These limitations motivates the development of a new paradigm.

To address these challenges, we propose CT-SDLM, a full-duplex **SDLM** with a **C**hronological **T**hinking mechanism. Inspired by the Adaptive Control of Thought-Rational (ACT-R) theoretical framework, we propose different node types corresponding to specific modules in the typical ACT-R architecture (Ritter et al., 2019), which replace the repeated silence tokens in conventional full-duplex system. Given the streaming user speech input, these nodes are chronologically predicted based on real-time semantic segments, ensuring causality in the thinking process. This design is consistent with human conversational behavior, where we tend to form associations and generate responses incrementally based on the semantic fragments of the interlocutor's speech, as shown in Figure 1(d). Furthermore, compared with typical long and ad hoc language chain, the structured node representation significantly reduces the token cost of reasoning while retaining useful information in auto-regressive generation, even if the user stops speaking abruptly. Therefore, this compact thinking chain is preemptible, allowing the system to seamlessly switch to response generation without incurring any additional latency. We conduct both objective and subjective evaluations to verify the effectiveness of chronological thinking in full-duplex SDLMs. Across task-oriented dialogue (Si et al., 2023; Yan et al., 2025) and open-domain spoken QA (Nachmani et al., 2023; Berant et al., 2013) benchmarks, CT-SDLM consistently outperforms strong baselines in both A/B tests and quantitative metrics. In addition, evaluations on full-duplex interaction metrics confirm that CT-SDLM introduces no additional latency in turn taking and user barge-in, demonstrating its robustness to conversational dynamics.

Our contributions are summarized as follows: (1) We propose a full-duplex SDLM with chronological thinking—a strictly causal, on-the-fly reasoning mechanism that enables the model to incrementally process semantic segments during user speech. (2) Our design yields a compact, preemptible thinking process that replaces redundant silence tokens without adding latency, and achieves consistent gains over baselines in both subjective and objective evaluations. (3) We demonstrate that chronological thinking serves as a viable new paradigm for full-duplex interaction, with the potential to influence future directions in

real-time dialogue modeling and human-machine communication.

2 Related work

Full-Duplex Spoken Dialogue Systems. Early spoken dialogue systems involved turn-based architectures (Sarikaya et al., 2002), where user speech input and system output occurred sequentially. Recent advances have shifted toward full-duplex dialogue systems (Veluri et al., 2024; Wang et al., 2024; Ma et al., 2025), enabling the agent to simultaneously listen and speak. Research in this area has largely focused on engineering challenges such as streaming ASR (Moritz et al., 2020; He et al., 2019; Yamamoto et al., 2025), incremental TTS (Chiba and Higashinaka, 2025; Skerry-Ryan et al., 2018), and mechanisms for barge-in handling (Chen et al.; Schlangen and Skantze, 2011). In addition, studies on incremental dialogue management (Khouzaimi et al., 2016; Zhang et al., 2025) have explored how conversational agents can respond more naturally in overlapping speech conditions. Nevertheless, during the listening phase, most of the existing systems primarily enforce silence by repeatedly predicting pause or silence tokens. Several approaches implement a streaming "think-while-listening" process by constructing explicit streaming CoTs (Chiang et al., 2025; Arora et al., 2025). These approaches fail to guarantee the causality of the CoT process (Arora et al., 2025; Chiang et al., 2025). Consequently, the gains these explicit methods provide for SDLMs are limited, or their utility is restricted to specific application scenarios.

Reasoning in Language Models. A parallel line of work explores how language models perform reasoning through explicit intermediate steps. CoT (Wei et al., 2022) and its extensions, such as Self-Consistency (Wang et al., 2023) and Tree-of-Thoughts (Yao et al., 2023), have shown that reasoning traces improve performance across arithmetic, logic, and commonsense reasoning. Recent methods such as Program-Aided Language Models (PAL) (Gao et al., 2023) and Toolformer (Schick et al., 2023) further highlight the benefits of externalized or structured reasoning. However, these approaches are designed for static text inputs, assuming access to the full problem before reasoning begins. They often rely on non-causal computation with hypothesis revision, which is incompatible with streaming conversational input.

Incremental and Streaming Reasoning. Another relevant line of work investigates reasoning and generation under streaming or incremental input (Calimeri et al., 2021). In simultaneous machine translation, prefix-to-prefix frameworks (Ma et al., 2019) and monotonic attention models (Ma et al., 2020; Arivazhagan et al., 2019) have been developed to balance accuracy and latency. Similar ideas in incremental decoding (Dalvi et al., 2018) allow models to generate partial outputs while processing incomplete inputs. In the LLM era, recent approaches such as StreamingLLM (Xiao et al., 2024), Medusa decoding (Cai et al., 2024), and attention sink methods (Xiao et al., 2024) have examined how large models can operate efficiently under bounded memory and real-time constraints. These studies illustrate the feasibility of causal reasoning under partial input but rarely consider spoken dialogue. Our work addresses this gap by introducing chronological thinking, a causal reasoning mechanism tailored for full-duplex spoken dialogue, enabling models to think continuously while listening without delaying the onset of response generation.

3 Method

In this section, we introduce the proposed full-duplex SDLM with chronological thinking mechanism. We first describe the overall network architecture, which enables simultaneous input processing and response generation, and then detail the chronological thinking mechanism that replaces redundant silence during periods when the agent listens to the user.

3.1 Model architecture

The network architecture of this paper is illustrated in Figure 2. It accepts two input streams: the user speech stream and the agent speech and text stream. The user's speech stream is input into a streaming speech encoder operating at a frame rate of 12.5Hz, producing continuous embeddings $\boldsymbol{X} \in \mathbb{R}^T$, where T denotes the number of frames. These embeddings are projected by a modality adapter and then summed with the embeddings of agent text tokens before being fed into an LLM backbone. To reduce the prediction burden on the LLM, instead of generating both the text and speech tokens with LLM (Hu et al., 2025), we only set the agent's text tokens $\boldsymbol{Y}^{\text{txt}}$ as the prediction target for

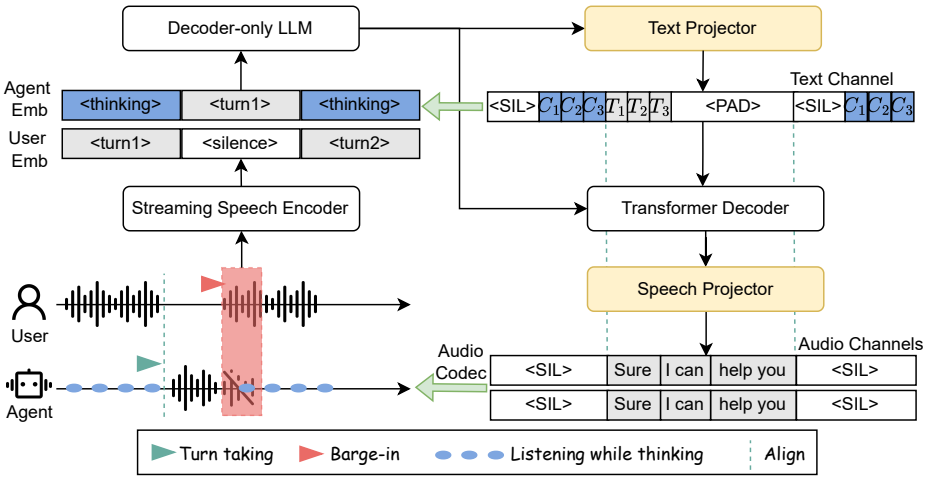

Figure 2: The network architecture of CT-Duplex with **chronological thinking** mechanism. Our model consists of a streaming speech encoder, a speech codec, an LLM backbone and a Transformer decoder. Compared to existing full-duplex SDLMs, chronological thinking is introduced during the listening phase.

the LLM. This autoregressive process can be formulated as:

$$P_{\theta_l}(\boldsymbol{Y}^{\text{txt}}|\boldsymbol{X}) = \prod_{t=1}^{T} P_{\theta_l}(Y_t^{\text{txt}}|\langle \boldsymbol{Y}_{1:t-1}^{\text{txt}}, \boldsymbol{X}_{1:t-1}\rangle), \quad (1)$$

where θ_l denotes the parameters of the LLM backbone. After the LLM backbone generating the agent's text tokens, an autoregressive Transformer decoder is employed to predict the agent's speech tokens $\boldsymbol{Y}^{\text{spc}}$. The inputs to this Transformer decoder include the agent's text tokens $\boldsymbol{Y}^{\text{txt}}$ and previously predicted agent speech tokens $\boldsymbol{Y}^{\text{spc}}$, conditioned on the LLM's last hidden states $\boldsymbol{h}$, which can be formulated as:

$$P_{\theta_d}(\boldsymbol{Y}^{\text{spc}}|\langle \boldsymbol{X}, \boldsymbol{Y}^{\text{txt}}, \boldsymbol{h}\rangle) = \prod_{t=1}^{T} P_{\theta_d}(Y_t^{\text{spc}}|\langle \boldsymbol{Y}_{1:t-1}^{\text{spc}}, \boldsymbol{Y}_{1:t-1}^{\text{txt}}, \boldsymbol{h}_t\rangle), \quad (2)$$

where θ_d represents the parameters of the Transformer decoder and $\boldsymbol{h}_t$ is LLM's hidden state at time step t. We employ Nanocodec (Casanova et al., 2025b) with Finite Scalar Quantization (Casanova et al., 2025a) to obtain agent speech tokens, generating speech codes at 12.5Hz. We jointly finetune the streaming speech encoder, LLM backbone, and autoregressive Transformer decoder using a multi-channel next token prediction training objective (Brown et al., 2020).

3.2 Agent text and speech alignment

Existing methods aligns the agent's text and speech at the token level, prepends the <BOS> token at the beginning of an agent's turn, appends the <EOS> token at the end of the agent turn, and uses text padding tokens <PAD> to pad the gap between text and speech tokens (Défossez et al., 2024; Hu et al., 2025). Furthermore, during the user's turn, most full-duplex models require the LLM to predict silence tokens <SIL> to fill the agent's stream. The expected agent's text tokens in the i-th turn can be represented as:

$$\begin{aligned}\boldsymbol{Y}^{\text{txt}} = [&< \texttt{SIL} >, \cdots, < \texttt{SIL} >, < \texttt{SIL} >, \cdots, < \texttt{SIL} >, \\ &< \texttt{BOS} >, R_{i,1}, R_{i,2}, \cdots, R_{i,T}, \\ &< \texttt{PAD} >, \cdots, < \texttt{PAD} >, < \texttt{EOS} >],\end{aligned} \quad (3)$$

where $R_{i,t}$ is the response token at time step t in turn i. When the agent needs to listen to the user's speech, the LLM repeatedly generates silence tokens <SIL>. When the user finishes speaking and it becomes the agent's turn to respond, the LLM outputs a <BOS> token, marking the beginning of the agent's turn. When the agent's turn ends or is interrupted by the user, the LLM outputs an <EOS> token, after which it resumes repeatedly generating silence tokens. This mechanism of outputting silence segments to maintain a listening state is a common paradigm in end-to-end full-duplex SDLM systems (Défossez et al., 2024; Yu et al., 2024; Hu et al., 2025).

However, forcing an LLM to repeatedly predict the same silence token not only leaves the listening window unexploited but also degrades its performance by biasing the next-token distribution (Xu et al., 2022). In fact, during human conversation, when listening to others speak, the brain does not simply remain idle; instead, it engages in thinking while processing the speaker's input. Similarly, a SDLM should not continuously output silence tokens during the user's turn. Instead, it should per-

form the thinking process as the user's input unfolds streamingly.

3.3 Chronological thinking

A straightforward approach for streaming thinking is to use the streaming ASR transcript text as the thinking content (Moritz et al., 2020; He et al., 2019; Yamamoto et al., 2025). We discuss the comparison between this method and the approach proposed in this paper in Appendix A.1. Inspired by research in human cognitive architecture, particularly the ACT-R theory, which divides human cognition into distinct modules, including the visual module for recognizing entities, the goal module for maintaining current intentions, the declarative module for retrieving knowledge, the manual module for controlling actions, and the production system for managing production rules (Ritter et al., 2019), we define five distinct types of nodes, which form a chain-structured chronological thinking content. The node types and their corresponding relationships with different modules in the ACT-R framework are shown in Table 1.

The chronological thinking chain grows with the user's input. Each time a semantic segment from the user is received, the agent obtains one or more of the five types of nodes. These nodes could be any one of these five node types. There is no fixed order for the nodes. **The type of node generated depends solely on the semantics**. For a full-duplex SDLM, a chain node is formatted as:

`{Node type} Node attributes`

The content within $\{\cdot\}$ represents the node type, followed by the node's attributes. A complete dialogue containing chronological thinking chains is shown in Appendix A.2.

We employ the Qwen2.5-72B-Instruct LLM model to generate chronological thinking chains based on input dialogue data (Team, 2024). After obtaining the chronological thinking chain, we convert it into tokens, denoted as $\boldsymbol{C}$, prepend the starting token <BOC> and append the ending token <EOC>, then put them to the positions originally occupied by silence tokens. Considering causality and latency requirements, we adopt the following strategies of varying lengths of $\boldsymbol{C}$:

We first define the length of chronological thinking chain tokens as M and the original silence tokens's length as S. For cases where M is less than or equal to $S-2$, we replace the last $M+2$ silence tokens with <BOC>, thinking chain tokens, and <EOC>. This ensures that the thinking tokens appear as late as possible, striving to ensure that the thinking tokens corresponding to a semantic segment appear later than the semantic segment in user's speech. Thus, the expected agent text tokens in the turn i can be expressed as:

$$\begin{aligned}\boldsymbol{Y}_i^{\text{txt}} = [&< \texttt{SIL} >, \cdots, < \texttt{SIL} >, \\ &< \texttt{BOC} >, C_{i,1}, \cdots, C_{i,M}, < \texttt{EOC} >, \\ &< \texttt{BOS} >, R_{i,1}, \cdots, R_{i,T}, < \texttt{PAD} >, \cdots, < \texttt{EOS} >],\end{aligned} \tag{4}$$

where $C_{i,m}$ is the m-th chronological thinking chain token in the turn i, and the number of <SIL> equals to $S-(M+2)$.

When M is greater than $S-2$, we first tokenize each chain node into tokens, and denote the number of tokens for each node as $M_1, M_2, \cdots, M_N$, where N is the number of nodes. We then retain the first n nodes such that $M' = \sum_{j=1}^{n} M_j \leq S-2$, and $M'+M_{n+1} > S-2$. We replace the last $M'+2$ silence tokens with chronological thinking chain tokens formed by the first n nodes, as well as <BOC> and <EOC>. We discuss the completeness of thinking chains in Appendix A.3. Finally, we use the text tokens with chronological thinking chains and the speech tokens to train the SDLM with multi-channel next token prediction (Brown et al., 2020).

4 Experiments

4.1 Data generation

Existing real-world conversational datasets, such as Fisher conversation dataset (Cieri et al., 2004), focus mainly on casual conversations, are insufficient to train the SDLM to respond to diverse human inquiries (Défossez et al., 2024; Hu et al., 2025; Chen et al., 2025a). To enhance the model's reasoning capabilities, we generate challenging dialogue data through synthetic methods. We first use seed content and an LLM to generate textual conversations, then convert these conversations into speech using a multi-speaker TTS system with voice cloning capabilities.

To create data for general conversation, we first curate a wide range of topics from sources like Wikipedia, covering general knowledge, common sense, and current events. These topics serve as seeds for Qwen2.5-72B-Instruct LLM (Team, 2024) to generate the textual dialogues, formulating a dataset named *GenConv*.

To train the model's ability in scenarios requiring reasoning, we introduce SpokenWOZ and se-

Table 1: The definitions of the five node types in chronological thinking chains and their corresponding relationships with different modules in the ACT-R theory

Node Type	DESCRIPTION	ACT-R
Entity	Extracts entities from the dialogue.	Visual module
Intent	Represents the user's goal	Goal module
Action	Denotes the agent's executable operation	Manual module
Knowledge	Retrieves factual or procedural knowledge.	Declarative module
Logic	Captures rules or logic generated by the agent	Production system

lect its training set as the seed dataset and create *spokenWOZ-G*. SpokenWOZ encompasses various reasoning scenarios, including those requiring cross-turn information, temporal, mathematical, and semantic reasoning (Si et al., 2023). We prompt Qwen2.5-72B-Instruct to generate topically-related dialogues with SpokenWOZ's format. We then calculate the similarity between the SpokenWOZ's dialogues and generated dialogues using the *thefuzz* python library and discard any generated dialogue with a similarity score over 90%. Similarly, to enhance the model's knowledge base, we create the *Llamaq-G* dataset by applying the same generation scheme to generate dataset with format like LLama Questions (Nachmani et al., 2023). After generating the textual dialogues and their corresponding chronological thinking chains, we synthesize the audio using Step-Audio-TTS-3B (Huang et al., 2025), a large-scale text-to-speech model capable of high-quality voice cloning. We generate $10.5k$ hours of speech for GenConv, $2.0k$ hours for SpokenWOZ-G and $2.7k$ hours for Llamaq-G.

We follow the method in (Hu et al., 2025) to create barge-in events: each dialogue turn has a random 50% probability of cutting off the agent's speech to allow the user to barge in. When a barge-in occurs, a 0.64s delay is enforced before the agent ceases speaking. We introduce a delay of 0.32s between the end of the user's speech and the start of the agent's response to enhance the naturalness of the dialogue. As demonstrated in (Hu et al., 2025), this approach enables the model to effectively learn the barge-in behavior.

4.2 Experimental settings

The model is implemented using the NeMo Toolkit (Kuchaiev et al., 2019) [1] and trained on 8 L40s (48G) GPUs. The LLM backbone is initialized from the Qwen2.5-1.5B-Instruct (Team, 2024). The speech encoder, text tokenizer and speech codec follow the ones in (Hu et al., 2025). The optimizer is AdamW with an inverse Square Root Annealing learning rate schedule. The learning rate starts from 3e-4 with a warm-up of 2500 steps.

[1] https://github.com/NVIDIA-NeMo/NeMo/tree/main/nemo/collections/speechlm2

4.3 Evaluation data and metrics

We utilize SpokenWOZ to validate the model's response quality in scenarios requiring reasoning (Si et al., 2023). Additionally, we employ the Mt-BenchEval from URO-Bench, a multi-turn dialogue evaluation dataset to assess the model's performance in daily conversations without complex reasoning (Yan et al., 2025). We employ the GPT scores generated by `gpt-4o-mini`, ranging from 0 to 5 to evaluate the performance. The prompts used is from URO-Bench (Yan et al., 2025). We also utilize the text BLEU score and Sentence-BERT similarity to evaluate the similarity between the generated responses and the target content (Papineni et al., 2002; Reimers and Gurevych, 2019).

To evaluate the model's factual knowledge capability, we introduce the Llama Questions and Web Questions datasets (Berant et al., 2013; Nachmani et al., 2023). The metrics utilized is accuracy. We use Whisper-large-v3 to transcribe the generated speech into text for calculating scores (Radford et al., 2022).

We follow (Chen et al., 2025a) to evaluate the turn-taking and barge-in performance of the full-duplex SDLM. The metrics include: (1) Turn-taking latency: The delay in the agent's response to the user's query in the first dialogue turn; (2) Barge-in latency: The time between the user's interruption and the agent stopping speech; (3) Barge-in success rate: The percentage of cases where the agent stops speaking within 1.5s after the user interrupts; We employed the *impatient* dataset in (Chen et al., 2025a), where interruptions occur approximately every 2 seconds on average, to evaluate turn-taking and barge-in performance.

Table 2: Performance on SpokenWOZ and MtBenchEval in terms of GPT score, BLEU, and Sentence-BERT. GT-LM is an optimal cascaded system that feeds ground-truth user turns to the LLM. We use "thk" to denote the proposed chronological thinking. The results of SALM-Duplex are reproduced by ourselves.

Method	SpokenWOZ			MtBenchEval		
	GPT score	BLEU	Sentence-BERT	GPT score	BLEU	Sentence-BERT
GT-LM	2.48	7.60	0.55	3.15	10.18	0.73
SALM-Duplex*	2.11	8.73	0.34	2.25	5.31	0.47
CT-Duplex w/o thk	2.40	12.92	0.52	2.39	7.00	0.64
CT-Duplex w/ thk	**2.61**	**16.30**	**0.59**	**2.44**	**7.34**	**0.67**

Table 3: Performance of different methods on Llama Questions and Web Questions benchmark in accuracy (%). Results of baseline systems are taken from (Zeng et al., 2024). The results of SALM-Duplex are reproduced by ourselves. We use "thk" to denote the proposed chronological thinking.

Method	Modality	# Params	Full-duplex	Llama Questions	Web Questions
TWIST	S→S	7B	✗	4.0	1.5
SpeechGPT	S→T	7B	✗	21.6	6.5
Spectron	S→T	1B	✗	21.9	6.1
Moshi	S→S	7B	✓	21.0	9.2
GLM-4-Voice	S→S	9B	✗	50.7	15.9
SALM-Duplex*	S→S	1.5B	✓	15.0	6.7
CT-Duplex w/o thk	S→S	1.7B	✓	30.4	13.2
CT-Duplex w/ thk	S→S	1.7B	✓	**31.4**	**13.3**

4.4 Results

4.4.1 Reasoning quality

We evaluate the performance of the proposed CT-Duplex model with and without the chronological thinking mechanism, denoted as CT-Duplex w/ thk and CT-Duplex w/o thk, respectively. For comparison, we also include the method from (Hu et al., 2025), named SALM-Duplex. The network architecture of SALM-Duplex is nearly identical to ours, with the key difference being that its LLM backbone simultaneously predicts both text tokens and audio tokens, whereas our LLM backbone only predicts text tokens, and an additional Transformer decoder is used to predict audio tokens. Furthermore, by feeding the LLM backbone with the ground-truth text of user inquiries and using the generated text to calculate scores, we establish an optimal cascaded system (GT-LM) to compare with our proposed method. The evaluation result of the reasoning abilities of different methods is shown in Table 2. It can be observed that the integration of chronological thinking has enhanced response quality, **especially in scenarios requiring reasoning**, as evidenced by 8.75% improvements on the SpokenWOZ benchmark. For everyday multi-turn dialogues evaluated on MtBenchEval, the observed gains are relatively modest (2.09%).

Meanwhile, Table 2 shows that the GT-LM method, which uses ground-truth user inquiry texts, performs worse than the proposed chronological thinking method on SpokenWOZ, but better than the method without thinking. However, on MtBenchEval, it achieves the best performance. This is because the output obtained by GT-LM is essentially the result of the text LLM without CoT. Therefore, for scenarios that require certain reasoning, this method performs worse than the thinking-enabled CT-Duplex w/ thk method. For scenarios that do not require reasoning, this method achieves optimal results due to the ideal input.

When compared to SALM-Duplex, both CT-Duplex w/ and w/o thk achieve better performance. This demonstrates the effectiveness of decoupling audio token prediction from the LLM backbone.

4.4.2 Factual knowlege capability

We further evaluate the model's level of factual knowledge. Table 3 shows the accuracy of our proposed method compared to baseline methods on both Llama Questions and Web Questions. It can be observed that when it comes to benchmarks requiring factual knowledge, the proposed choronological thinking method showed negligible improvement. This is because factual knowledge-

Table 4: Evaluation of conversational behaviors of different methods on the *Impatient* dataset proposed in (Chen et al., 2025a), in terms of turn-taking latency, barge-in latency, and barge-in success rate. Results of baseline systems are taken from (Chen et al., 2025a). We use "thk" to denote the proposed chronological thinking. "Suc rate" stands for "Success rate" and "Lat" stands for "Latentcy"

Method	Turn-taking	Barge-in	
	Lat (↓)	Lat (↓)	Suc rate (↑)
Freeze-Omni	1.17	1.20	79.50%
dGSLM	0.57	0.86	85.00%
Moshi	n.a.	0.81	55.10%
ORISE	0.43	0.61	96.80%
SALM-Duplex*	0.92	0.69	87.50%
CT-Duplex w/o thk	**0.45**	**0.53**	88.63%
CT-Duplex w/ thk	0.68	0.54	**94.05%**

based Question-Answering tasks require minimal reasoning, as the model only needs to possess the relevant knowledge to answer questions correctly. Compared to SALM-Duplex, our models still achieve higher accuracy as we alleviate the predictive burden on the LLM. Meanwhile, when compared with other baseline methods, it can be observed that except for GLM-4-Voice, our models outperform all baseline approaches, even though baseline methods such as Moshi have significantly more parameters than our models (7B vs. 1.7B). Although the GLM-4-Voice method achieves much higher accuracy than our proposed models, its 9B-parameter count far exceeds that of our models, and its half-duplex structure is not constrained by real-time and causal requirements.

4.4.3 Turn-taking and barge-in evaluation

Table 4 presents a comparison of the turn-taking and barge-in performance between the proposed method and the baseline method. It can be observed that the results are comparable with and without the thinking mechanism, whether in terms of turn-taking latency or barge-in behavior. These results demonstrate that chronological thinking introduced in this work does not impair the full-duplex SDLM's turn-taking or barge-in abilities. That is because the proposed method generates nodes of the thinking chain chronologically with the input, introducing no additional computational overhead or extra latency. Besides, we only replace the silence tokens in the original full-duplex SDLM with thinking chain tokens, ensuring that the thinking process occurs exclusively during the listening phase without altering any response tokens of the SDLM.

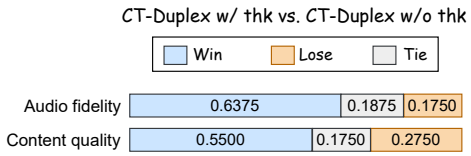

Figure 3: The A/B test results, including evaluations of both audio generation quality and response content quality.

4.4.4 Subjective results

We conduct an A/B test to validate the subjective evaluation results of the proposed chronological thinking method, aiming to assess its performance in terms of audio fidelity and response content quality (Brachmanski). The experimental setups are detailed in Appendix A.4. The test dataset is sourced from SpokenWOZ. We select 10 fluent English speakers to evaluate the audio quality and content of the responses generated by CT-Duplex w/ thk and CT-Duplex w/o thk. The experimental results are shown in Figure 3. It can be observed that the subjective metrics for both audio quality and response content of CT-Duplex w/ thk are superior to those of CT-Duplex w/o thk. Although the proposed thinking method is designed to improve response content quality, the enhancement in response quality leads to a lower loss in agent text prediction. This, in turn, allows the model to learn more from the audio prediction loss, thereby achieving a higher level of audio quality.

5 Conclusion

This paper proposes a chronological thinking mechanism that enables full-duplex SDLMs to possess a human-like thinking-while-listening ability. Inspired by research on human cognitive architecture, we introduce a chronological thinking chain comprising five distinct node types, each corresponding to components of the ACT-R framework. By replacing silence tokens in conventional full-duplex SDLMs with chronological thinking chain tokens, we achieve causal and no-additional-latency thinking during listening phases. Objective and subjective evaluation results demonstrate that the proposed method achieves higher response quality, especially in scenarios requiring reasoning, without compromising the turn-taking and barge-in performance of SDLMs.

Acknowledgements

This research is supported by the National Research Foundation, Singapore under its National Large Language Models Funding Initiative. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not reflect the views of National Research Foundation, Singapore.

References

Naveen Arivazhagan, Colin Cherry, Wolfgang Macherey, Chung-Cheng Chiu, Semih Yavuz, Ruoming Pang, Wei Li, and Colin Raffel. 2019. Monotonic infinite lookback attention for simultaneous machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1313–1323.

Siddhant Arora, Jinchuan Tian, Hayato Futami, Jiatong Shi, Yosuke Kashiwagi, Emiru Tsunoo, and Shinji Watanabe. 2025. Chain-of-thought reasoning in streaming full-duplex end-to-end spoken dialogue systems. *arXiv preprint arXiv:2510.02066*.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1533–1544.

Stefan Brachmanski. Subjective assessment of quality of audio and video signals by means of ab test.

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.

Tianle Cai, Yuhong Li, Zhengyang Geng, Hongwu Peng, Jason D Lee, Deming Chen, and Tri Dao. 2024. Medusa: Simple llm inference acceleration framework with multiple decoding heads. In *International Conference on Machine Learning*, pages 5209–5235. PMLR.

Francesco Calimeri, Giovambattista Ianni, Francesco Pacenza, Simona Perri, and Jessica Zangari. 2021. Stream reasoning with incremental grounding. In *5th Stream Reasoning Workshop*.

Edresson Casanova, Ryan Langman, Paarth Neekhara, Shehzeen Hussain, Jason Li, Subhankar Ghosh, Ante Jukić, and Sang-gil Lee. 2025a. Low frame-rate speech codec: a codec designed for fast high-quality speech llm training and inference. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Edresson Casanova, Paarth Neekhara, Ryan Langman, Shehzeen Hussain, Subhankar Ghosh, Xuesong Yang, Ante Jukić, Jason Li, and Boris Ginsburg. 2025b. Nanocodec: Towards high-quality ultra fast speech llm inference. *arXiv preprint arXiv:2508.05835*.

Chen Chen, Ke Hu, Chao-Han Huck Yang, Ankita Pasad, Edresson Casanova, Weiqing Wang, Szu-Wei Fu, Jason Li, Zhehuai Chen, Jagadeesh Balam, and 1 others. Reinforcement learning enhanced full-duplex spoken dialogue language models for conversational interactions. In *Second Conference on Language Modeling*.

Chen Chen, Ke Hu, Chao-Han Huck Yang, Ankita Pasad, Edresson Casanova, Weiqing Wang, Szu-Wei Fu, Jason Li, Zhehuai Chen, Jagadeesh Balam, and 1 others. 2025a. Reinforcement learning enhanced full-duplex spoken dialogue language models for conversational interactions. In *Second Conference on Language Modeling*.

Junjie Chen, Yao Hu, Junjie Li, Kangyue Li, Kun Liu, Wenpeng Li, Xu Li, Ziyuan Li, Feiyu Shen, Xu Tang, and 1 others. 2025b. Fireredchat: A pluggable, full-duplex voice interaction system with cascaded and semi-cascaded implementations. *arXiv preprint arXiv:2509.06502*.

Qian Chen, Yafeng Chen, Yanni Chen, Mengzhe Chen, Yingda Chen, Chong Deng, Zhihao Du, Ruize Gao, Changfeng Gao, Zhifu Gao, and 1 others. 2025c. Minmo: A multimodal large language model for seamless voice interaction. *arXiv preprint arXiv:2501.06282*.

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. Shanks: Simultaneous hearing and thinking for spoken language models. *arXiv preprint arXiv:2510.06917*.

Yuya Chiba and Ryuichiro Higashinaka. 2025. Investigating the impact of incremental processing and voice activity projection on spoken dialogue systems. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 3687–3696.

Christopher Cieri, David Miller, and Kevin Walker. 2004. The fisher corpus: A resource for the next generations of speech-to-text. In *LREC*, volume 4, pages 69–71.

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*.

Fahim Dalvi, Nadir Durrani, Hassan Sajjad, and Stephan Vogel. 2018. Incremental decoding and training methods for simultaneous translation in neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the*

Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), pages 493–499.

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*.

Ding Ding, Zeqian Ju, Yichong Leng, Songxiang Liu, Tong Liu, Zeyu Shang, Kai Shen, Wei Song, Xu Tan, Heyi Tang, and 1 others. 2025. Kimi-audio technical report. *arXiv preprint arXiv:2504.18425*.

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.

Jian Guan and Minlie Huang. 2023. Mitigating the learning bias towards repetition by self-contrastive training for open-ended generation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 6897–6909.

Yanzhang He, Tara N Sainath, Rohit Prabhavalkar, Ian McGraw, Raziel Alvarez, Ding Zhao, David Rybach, Anjuli Kannan, Yonghui Wu, Ruoming Pang, and 1 others. 2019. Streaming end-to-end speech recognition for mobile devices. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6381–6385. IEEE.

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*.

Ailin Huang, Boyong Wu, Bruce Wang, Chao Yan, Chen Hu, Chengli Feng, Fei Tian, Feiyu Shen, Jingbei Li, Mingrui Chen, and 1 others. 2025. Step-audio: Unified understanding and generation in intelligent speech interaction. *arXiv preprint arXiv:2502.11946*.

Hatim Khouzaimi, Romain Laroche, and Fabrice Lefèvre. 2016. Reinforcement learning for turn-taking management in incremental spoken dialogue systems. In *IJCAI*, pages 2831–2837.

Oleksii Kuchaiev, Jason Li, Huyen Nguyen, Oleksii Hrinchuk, Ryan Leary, Boris Ginsburg, Samuel Kriman, Stanislav Beliaev, Vitaly Lavrukhin, Jack Cook, and 1 others. 2019. Nemo: a toolkit for building ai applications using neural modules. *arXiv preprint arXiv:1909.09577*.

Borui Liao, Yulong Xu, Jiao Ou, Kaiyuan Yang, Weihua Jian, Pengfei Wan, and Di Zhang. 2025. Flexduo: A pluggable system for enabling full-duplex capabilities in speech dialogue systems. *arXiv preprint arXiv:2502.13472*.

Ting-En Lin, Yuchuan Wu, Fei Huang, Luo Si, Jian Sun, and Yongbin Li. 2022. Duplex conversation: Towards human-like interaction in spoken dialogue systems. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3299–3308.

Mingbo Ma, Liang Huang, Hao Xiong, Renjie Zheng, Kaibo Liu, Baigong Zheng, Chuanqiang Zhang, Zhongjun He, Hairong Liu, Xing Li, and 1 others. 2019. Stacl: Simultaneous translation with implicit anticipation and controllable latency using prefix-to-prefix framework. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3025–3036.

Xutai Ma, Juan Miguel Pino, James Cross, Liezl Puzon, and Jiatao Gu. 2020. Monotonic multihead attention. In *International Conference on Learning Representations*.

Ziyang Ma, Yakun Song, Chenpeng Du, Jian Cong, Zhuo Chen, Yuping Wang, Yuxuan Wang, and Xie Chen. 2025. Language model can listen while speaking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 24831–24839.

Niko Moritz, Takaaki Hori, and Jonathan Le. 2020. Streaming automatic speech recognition with the transformer model. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6074–6078. IEEE.

Eliya Nachmani, Alon Levkovitch, Roy Hirsch, Julian Salazar, Chulayuth Asawaroengchai, Soroosh Mariooryad, Ehud Rivlin, RJ Skerry-Ryan, and Michelle Tadmor Ramanovich. 2023. Spoken question answering and speech continuation using spectrogram-powered llm. *arXiv preprint arXiv:2305.15255*.

Tu Anh Nguyen, Eugene Kharitonov, Jade Copet, Yossi Adi, Wei-Ning Hsu, Ali Elkahky, Paden Tomasello, Robin Algayres, Benoit Sagot, Abdelrahman Mohamed, and 1 others. 2023. Generative spoken dialogue language modeling. *Transactions of the Association for Computational Linguistics*, 11:250–266.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2022. Robust speech recognition via large-scale weak supervision. *arXiv preprint*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Frank E Ritter, Farnaz Tehranchi, and Jacob D Oury. 2019. Act-r: A cognitive architecture for modeling

cognition. *Wiley Interdisciplinary Reviews: Cognitive Science*, 10(3):e1488.

Ruhi Sarikaya, Yuqing Gao, Hakan Erdogan, and Michael Picheny. 2002. Turn-based language modeling for spoken dialog systems. In *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages I–781. IEEE.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36:68539–68551.

David Schlangen and Gabriel Skantze. 2011. A general, abstract model of incremental dialogue processing. *Dialogue & Discourse*, 2(1):83–111.

Shuzheng Si, Wentao Ma, Haoyu Gao, Yuchuan Wu, Ting-En Lin, Yinpei Dai, Hangyu Li, Rui Yan, Fei Huang, and Yongbin Li. 2023. Spokenwoz: A large-scale speech-text benchmark for spoken task-oriented dialogue agents. *Advances in Neural Information Processing Systems*, 36:39088–39118.

RJ Skerry-Ryan, Eric Battenberg, Ying Xiao, Yuxuan Wang, Daisy Stanton, Joel Shor, Ron Weiss, Rob Clark, and Rif A Saurous. 2018. Towards end-to-end prosody transfer for expressive speech synthesis with tacotron. In *international conference on machine learning*, pages 4693–4702. PMLR.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Bandhav Veluri, Benjamin N Peloquin, Bokai Yu, Hongyu Gong, and Shyamnath Gollakota. 2024. Beyond turn-based interfaces: Synchronous llms as full-duplex dialogue agents. *arXiv preprint arXiv:2409.15594*.

Peng Wang, Songshuo Lu, Yaohua Tang, Sijie Yan, Wei Xia, and Yuanjun Xiong. 2024. A full-duplex speech dialogue scheme based on large language model. *Advances in Neural Information Processing Systems*, 37:13372–13403.

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*.

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.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2024. Efficient streaming language models with attention sinks. In *The Twelfth International Conference on Learning Representations*.

Jin Xu, Xiaojiang Liu, Jianhao Yan, Deng Cai, Huayang Li, and Jian Li. 2022. Learning to break the loop: Analyzing and mitigating repetitions for neural text generation. *Advances in Neural Information Processing Systems*, 35:3082–3095.

Kenta Yamamoto, Ryu Takeda, and Kazunori Komatani. 2025. Analysis of voice activity detection errors in api-based streaming asr for human-robot dialogue. In *Proceedings of the 15th International Workshop on Spoken Dialogue Systems Technology*, pages 245–253.

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*.

Hao Zhang, Weiwei Li, Rilin Chen, Vinay Kothapally, Meng Yu, and Dong Yu. 2025. Llm-enhanced dialogue management for full-duplex spoken dialogue systems. *arXiv preprint arXiv:2502.14145*.

Wenhong Zhu, Hongkun Hao, and Rui Wang. 2023. Penalty decoding: Well suppress the self-reinforcement effect in open-ended text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1218–1228.

A Appendix

A.1 Discussion about streaming ASR and chronological thinking

Using streaming ASR results as thinking content is an intuitive implementation approach to enable SDLMs with the thinking-while-listening ability (Moritz et al., 2020; He et al., 2019; Yamamoto et al., 2025). Under the task scenarios and experimental setup of this paper, the performance upper

Round1:
User:
Help me order a restaurant to celebrate my birthday this weekend.
Agent:
###
[Help me]{INTENT} Request assistance
[order a restaurant]{ACTION} Initiate booking process
[to celebrate my birthday]{LOGIC} Purpose: birthday celebration
[this weekend]{ENTITY} Timeframe: weekend
{KNOWLEDGE} Birthday: Decorations, discounts, or special perks
###
Certainly! For a birthday celebration, I recommend a place that offers a special dining experience. Do you have a preferred cuisine or location in mind?

Round2:
User:
I'd like to try Italian food, and the place should be in the city center.
Agent:
###
[I'd like to try]{INTENT} Express cuisine
[Italian food]{ENTITY} Cuisine:Italian
[and the place should be in the city center]{LOGIC} Location:city center
###
Great choice! How about Trattoria Bella, located right in the heart of the city? It's known for its authentic Italian dishes and cozy atmosphere.

Round3:
User:
That sounds perfect. Can you book a table for me and three of my friends?
Agent:
###
[That sounds perfect]{INTENT} Confirmation
[Can you book a table]{ACTION} Book table
[for me]{ENTITY} 1 person
[and three of my friends?]{ENTITY} 3 people
{LOGIC} 4 people
{ENTITY} Restaurant@ctx:Trattoria Bella
###
Sure, I can book a table for four people at Trattoria Bella. What day this weekend would you like to visit?

Figure 4: A dialogue example with chronological thinking chains.

bound of this streaming ASR method corresponds to the performance of the text LLM when fed with the ground-truth user inquiry text. This output reflects the LLM's response without any thinking process; when the task requires reasoning, its performance is significantly affected. In contrast, the chronological thinking method proposed in this paper, which includes analysis of user intent, contextual entity binding, logical reasoning, prerequisite knowledge extraction, and prediction of agent actions, achieves superior reasoning performance. In fact, the comparison between the GT-LM results and our method in Table 2 demonstrates this conclusion. Additionally, using streaming ASR results as thinking content is also affected by ASR accuracy. In experiments, we observe that streaming ASR models often insert padding tokens between adjacent text tokens, which disrupts semantic consistency and degrades SDLM's performance.

A.2 Example of chronological thinking chains

Figure 4 displays a three-turn dialogue example incorporating chronological thinking chains. It includes simple semantic reasoning (inferring the preferred restaurant for a birthday celebration) and mathematical reasoning (me + 3 people = 4 people), as well as cross-turn entity tracking (Trattoria Bella). In the design of the thinking chain, we use "@ctx:ID" to bind and track entities across dialogue turns, with this functionality implemented within the ENTITY nodes. The usage of "@ctx:ID" is demonstrated in the last line of the thinking chain in the third dialogue turn. The transcribed speech content within $[\cdot]$ is only used during the data generation phase to control the generation of chronological thinking chains and is not included in the input and output stream of the full-duplex SDLM.

Table 5: The ratio of dialogue turns where the thinking chain tokens being less than the frame count of user utterances.

Dataset	Ratio
GenConv	98.91%
SpokenWOZ-G	96.78%
Llamaq-G	94.69%

A.3 Completeness of thinking chains

Although in Section 3.3 we employ truncation to ensure that thinking tokens do not occupy the original response tokens, the proposed method significantly shortens the length of the generated thinking content by replacing natural language with structured thinking chain nodes. This maximizes the completeness of the thinking process within a given time duration. To verify the completeness of the generated thinking chains during training process, we statistically analyze the frame count of each user utterance and the corresponding token count of the thinking chains, across three training datasets. We then calculate the ratio of dialogue turns where the thinking chain tokens being less than the frame count of user utterances, as summarized in Table 5. The results demonstrate that for the majority of cases across all three datasets, the token count of the thinking chains remains lower than the frame count of user utterances, indicating that most thinking chains are fully preserved.

A.4 Subjective experiments

To ensure a fair comparison, we conduct a blind A/B test where human evaluators are presented with paired responses from the full-duplex SDLM with and without chronological thinking in a random order. Evaluators are asked to select the preferred response based on audio fidelity and response content quality, or mark them as a tie if no significant difference is observed.

We recruit 10 fluent English speakers as evaluators, where each participant assesses 20 audio samples selected randomly from the test dataset. These subjects are either from English-speaking countries or possess over seven years of speaking experience, ensuring high proficiency. To ensure precise evaluation, evaluators are allowed to replay the audio repeatedly but are required to listen to each sample at least three times before providing a rating.

A.5 LLM usage statement

We used ChatGPT only for minor language editing to improve clarity and conciseness. No part of the research idea, methodology, or analysis was generated by LLMs.