

# Think, Verbalize, then Speak: Bridging Complex Thoughts and Comprehensible Speech

Sang Hoon Woo\*    Sehun Lee\*    Kang-wook Kim    Gunhee Kim  
Seoul National University

tonyswoo@gmail.com    shlee@vision.snu.ac.kr    full324@snu.ac.kr    gunhee@snu.ac.kr

## Abstract

Recent spoken dialogue systems leverage large language models (LLMs) for advanced reasoning. However, a mismatch between optimal textual and verbal delivery limits their effectiveness in spoken communication. While some approaches adapt LLMs for speech-friendly outputs, their impact on reasoning remains under-explored. We propose **THINK-VERBALIZE-SPEAK**, a framework that separates reasoning from spoken delivery to preserve the full reasoning capacity of LLMs. Central to our method is *verbalizing*, an intermediate step that translates thoughts into natural, speech-ready text. We also introduce **REVERT**, a latency-efficient verbalizer based on incremental and asynchronous summarization. Experiments across multiple benchmarks show that our method enhances speech naturalness and conciseness with minimal impact on reasoning. We release both the dataset and pipeline to support future research.

## 1 Introduction

Humans inherently differentiate between what to think internally and what to say externally. That is, individuals can easily reformulate their thought processes into a structure that is more appropriate for verbal communication (Levelt, 1993; Indefrey and Levelt, 2004; Sahin et al., 2009). However, current spoken dialogue systems do not incorporate mechanisms that emulate this process. This limitation is significant given the increasing popularity of reasoning models that often produce a long chain-of-thought to address complex problems (Wei et al., 2022; OpenAI, 2024; Guo et al., 2025).

Current spoken dialogue systems typically employ a two-stage framework, herein referred to as the THINK-SPEAK framework (Ji et al., 2024; Dongre et al., 2024; Xu et al., 2025; Fang et al., 2025). In this approach, the system first constructs the

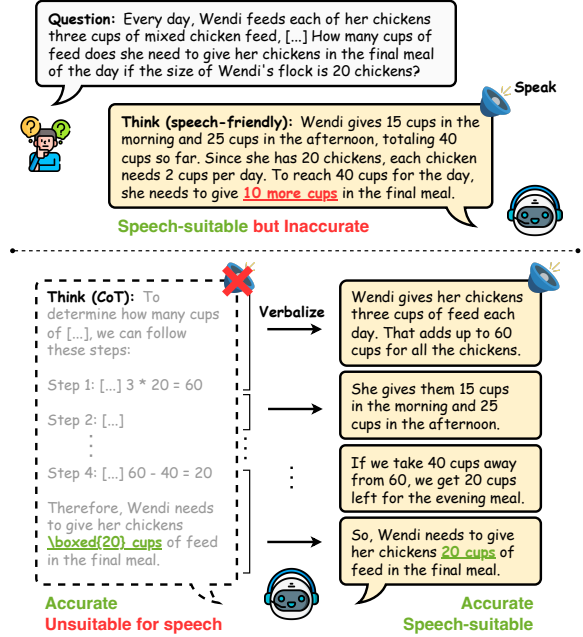


Figure 1: To produce both speech-friendly and accurate responses, we decouple thinking from verbalizing. A chain-of-thought process (verbose, structured, or in technical formats such as LaTeX) is unsuitable for spoken delivery. Conversely, generating a speech-friendly answer without underlying reasoning may be fast but often results in inaccurate responses. Moreover, waiting for the thinking to complete leads to severe latency. By verbalizing internal thoughts incrementally, we achieve accuracy, speech-suitability, and low latency.

content of the speech (THINK), and then generates the corresponding spoken output (SPEAK). However, large language models (LLMs), which are commonly used in the THINK stage, combined with test-time computing methods, such as chain-of-thought reasoning, often yield responses that are not suitable for spoken dialogue. Some studies (Cho et al., 2024; Hyeon et al., 2025) attempt to address this issue by guiding the model to produce speech-friendly outputs, either through fine-tuning or prompting. Nevertheless, enforcing a speech-friendly thought format may substantially deteriorate the performance of reasoning processes. An

\*Equal contribution.

illustrative example is provided in Figure 1.

In this work, we propose the THINK-VERBALIZE-SPEAK framework, which introduces an intermediate verbalization stage to translate raw model reasoning into speech-friendly, comprehensible utterances. By employing a dedicated verbalization model—while keeping the reasoning model fixed—our system produces natural, concise speech output without sacrificing problem-solving capabilities. To address the latency incurred by a naive two-stage implementation, we present the **RE**asoning to **VER**bal Text (REVERT) model, which enables efficient, incremental verbalization and reduces response time by up to 66% compared to sequential approaches. Extensive automatic and human evaluations confirm that our method generates speech output that is both natural and accurate, with minimal loss in reasoning performance, maintaining robust performance even with smaller verbalization models.

Our key contributions to the field are as follows:

- We introduce the THINK-VERBALIZE-SPEAK framework, which enhances the speech-friendliness of generated utterances while preserving the problem-solving capabilities of the underlying reasoning model.
- We propose REVERT, a latency-efficient verbalization model that significantly reduces system latency by initiating verbal output before the underlying reasoning process is complete.
- We develop the solve-summarize-scatter data pipeline, which produces reasoning data with incremental speech-friendly summaries from QA datasets. We publicly release the dataset used to train the REVERT model.

## 2 Related Work

**Reasoning in LLMs** While LLMs have achieved significant progress through scaling model and dataset sizes, these advancements alone remain insufficient for addressing complex tasks such as arithmetic and commonsense reasoning (Cobbe et al., 2021; Ho et al., 2020; Wang et al., 2024a,c). The introduction of chain-of-thought (CoT) prompting (Wei et al., 2022) has enabled LLMs to demonstrate enhanced reasoning abilities. The models specialized in reasoning incorporate non-linear processes such as reflection and backtracking. However, these enhanced reasoning processes are lengthy and verbose, making it difficult

for users to follow in real time or stay engaged during spoken interactions.

**Spoken Dialogue Systems** Spoken dialogue systems are typically categorized as cascaded or end-to-end (Ji et al., 2024). Cascaded systems employ a pipeline architecture comprising automatic speech recognition (ASR), a dialogue model, and a text-to-speech (TTS) component, using text as the intermediate representation. This modular approach allows for the integration of state-of-the-art components at each stage. However, LLM-based dialogue models within these systems often produce outputs optimized for reading—such as bullet points, sentence fragments, or formatted equations—rather than for spoken communication, which can undermine the naturalness of speech-based interactions.

End-to-end systems eliminate the dependency on intermediate text, thereby preserving paralinguistic cues and facilitating more natural speech generation. Recent work includes fully textless models (Lakhotia et al., 2021; Zhang et al., 2023; Défossez et al., 2024), text-speech interleaved architectures (Zeng et al., 2024; Wang et al., 2024b), and parallel decoding approaches (Xie and Wu, 2024; Gao et al., 2025; Xu et al., 2025). While end-to-end systems are more effective at generating speech-friendly outputs, they typically exhibit weaker reasoning capabilities compared to conventional LLMs.

**Speech-Suitable Text Generation** Recent work on speech-suitable text can be divided into two main perspectives. The first is normalization, which converts input text into a form suitable for direct spoken delivery. For example, Math-Reader (Hyeon et al., 2025) translates LaTeX mathematical expressions into English, enabling spoken rendering crucial as LLMs often output LaTeX when solving arithmetic problems. Beyond normalization, a second perspective considers how content should be verbalized for effective spoken communication. As discussed by Cho et al. (2024), "speechworthiness" refers to properties that make text well-suited for spoken presentation, including clarity, utterance length, and information density. Additionally, unlike text, audio requires listeners to engage with the content sequentially, without the ability to selectively skip sections.

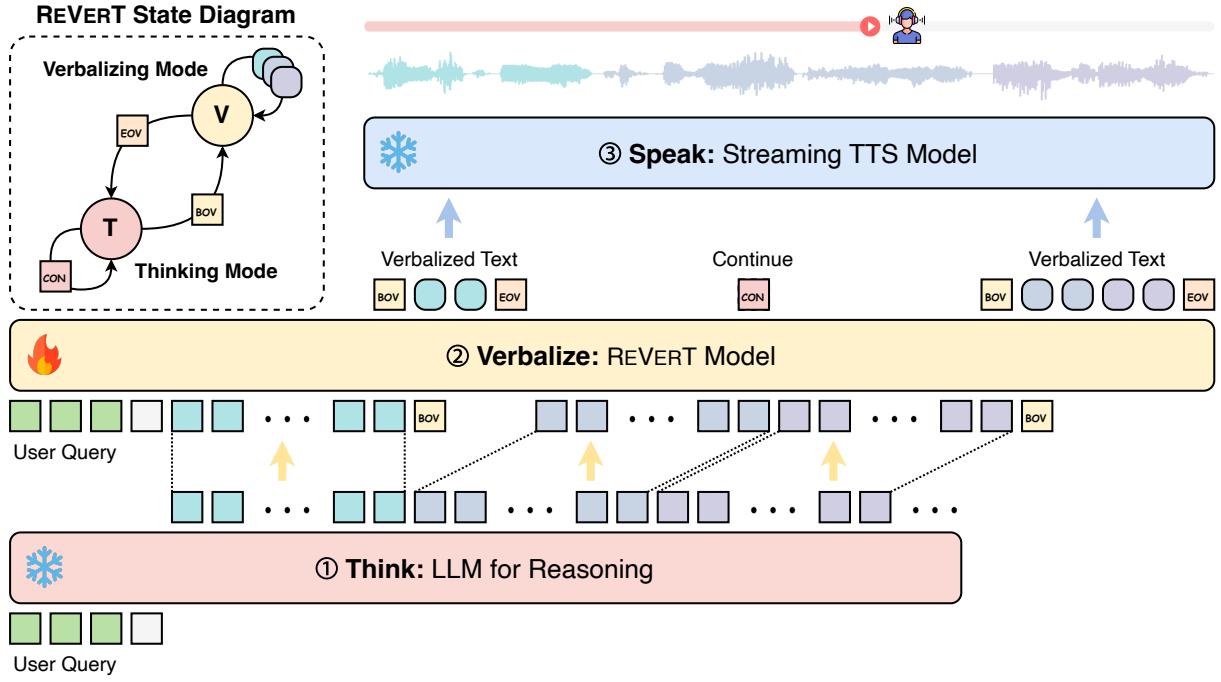


Figure 2: Overall framework of THINK-VERBALIZE-SPEAK. For a given user query, (1) a reasoning LLM generates a step-by-step chain-of-thoughts in text. (2) The REVERT model verbalizes the intermediate reasoning outputs into speech-friendly text incrementally, reducing the latency. (3) A TTS model converts the verbalized text into synthesized speech output in a streaming manner. REVERT model operates in two modes: *thinking mode* ( $S_T$ ), where it receives and accumulates reasoning chunks, and *verbalizing mode* ( $S_V$ ), where it translates accumulated reasoning into speech-friendly text. Please refer to § 3.2 for the usage of special tokens  $\langle \text{bov} \rangle$ ,  $\langle \text{con} \rangle$ , and  $\langle \text{eov} \rangle$ .

### 3 THINK-VERBALIZE-SPEAK

Our framework, THINK-VERBALIZE-SPEAK, modifies the traditional cascaded system by generating response content in two stages; a reasoning stage ensures response accuracy (THINK) and a translation stage converts the reasoning into a verbal response (VERBALIZE). The resulting response is then synthesized into speech (SPEAK). Figure 2 provides an overview of our approach utilizing REVERT as the verbalizer. We employ an off-the-shelf reasoning LLM and a streaming TTS model, both of which remain frozen; only the REVERT model undergoes training under our framework.

#### 3.1 THINK

In the THINK stage, we leverage the problem-solving abilities of a reasoning LLM. Upon receiving a user query, the LLM solves the query using chain-of-thought reasoning. The reasoning output is then streamed to the subsequent stage.

#### 3.2 VERBALIZE

In the VERBALIZE stage, the system receives the streaming reasoning output from the THINK stage and translates it into speech-friendly utterances. A

naive method is the sequential approach, in which the system completes the reasoning stage before generating speech-friendly translations of the output. However, this causes significant latency.

To address this issue, we propose REVERT, a latency-efficient verbalizer. As described in Algorithm 1, the REVERT model operates asynchronously with the reasoning LLM from the THINK stage, incrementally generating speech-friendly utterances based on partial reasoning outputs.

The REVERT model operates in two distinct modes: *thinking mode* ( $S_T$ ) and *verbalizing mode* ( $S_V$ ). In thinking mode, REVERT receives and processes the outputs of the reasoning model. While the reasoning LLM emits output token by token, REVERT processes these outputs in segments, defined by a predetermined set of delimiters. This chunk-based processing enables more efficient computation through hardware parallelism.

After processing each segment, REVERT determines whether to initiate verbalization via single token generation. If the next token is  $\langle \text{con} \rangle$ , REVERT continues processing additional reasoning segments. If the next token is  $\langle \text{bov} \rangle$ , the model transitions to verbalizing mode, where REVERT

---

**Algorithm 1** THINK-VERBALIZE-SPEAK with REVERT

---

**Require:** a trained REVERT  $p_\theta$ , a reasoning model  $q$ , user query tokens  $\mathcal{Q}$ , a set of delimiters  $\mathcal{D}$ .

```
1: function THINK ( $q, \mathcal{Q}$ )
2:   initialize  $i \leftarrow 0$ 
3:   repeat
4:     generate  $r_i \sim q(\cdot \mid \mathcal{Q}, r_{<i})$ 
5:     send  $r_i$  to the verbalizer
6:      $i \leftarrow i + 1$ 
7:   until  $r_{i-1} = \langle \text{eos} \rangle$ 
8: end function

9: function VERBALIZE ( $p_\theta, \mathcal{Q}$ )
10:  set the current state  $\mathcal{S}$  as thinking mode  $\mathcal{S}_T$ 
11:  initialize a context  $\mathcal{C} \leftarrow \mathcal{Q}$ 
12:  while reasoning is not complete do
13:    receive texts from the reasoning model.
14:    process these texts into segment  $\mathcal{R}$  with  $\mathcal{D}$ .
15:    if  $\mathcal{S}$  is in thinking mode ( $\mathcal{S}_T$ ) then
16:      update  $\mathcal{C} \leftarrow (\mathcal{C}, \mathcal{R})$ .
17:      sample  $s \sim p_\theta(\cdot \mid \mathcal{C}) \triangleright \langle \text{con} \rangle$  or  $\langle \text{bov} \rangle$ 
18:      if  $s = \langle \text{bov} \rangle$  then
19:        transition state  $\mathcal{S}$  to verbalizing mode.
20:      end if
21:    end if
22:    if  $\mathcal{S}$  is in verbalizing mode ( $\mathcal{S}_V$ ) then
23:      update  $\mathcal{C} \leftarrow (\mathcal{C}, \langle \text{bov} \rangle) \triangleright$  Begin verbalization
24:      initialize the verbalization buffer  $\mathcal{V} \leftarrow ()$ .
25:      repeat
26:        generate  $v \sim p_\theta(\cdot \mid \mathcal{C})$ .
27:        update context:  $\mathcal{C} \leftarrow (\mathcal{C}, v)$ .
28:         $\mathcal{V} \leftarrow (\mathcal{V}, v)$ 
29:      until  $v = \langle \text{eov} \rangle \triangleright$  End of verbalization
30:      transition state  $\mathcal{S}$  to thinking mode.
31:      send  $\mathcal{V}$  to the TTS model.
32:    end if
33:  end while
34: end function
```

---

translates the accumulated reasoning segments into speech-friendly output tokens. The model continues generating verbalized text until it produces the  $\langle \text{eov} \rangle$  token, at which point it forwards the generated text to the subsequent stage, returns to thinking mode, and resumes processing reasoning segments. The state diagram of REVERT is included in Figure 2.

In summary, REVERT functions as an incremental, asynchronous, speech-oriented summarizer of the reasoning output. Since REVERT performs no reasoning itself, it can be implemented as a more compact model compared to the reasoning LLM.

### 3.3 SPEAK

In the SPEAK stage, we convert the utterances to speech using a TTS model. Specifically, we adopt a TTS model that supports both streaming input and output, allowing the system to process streaming outputs from the VERBALIZE stage and play the generated speech with minimal delay for the user.

### 3.4 REVERT Training

Since the reasoning LLM and streaming TTS models remain frozen, we describe only the training procedure for the REVERT model. We below discuss the training data format, the dataset construction pipeline, and the training objective.

**Training Data** Each training example comprises a user query  $\mathcal{Q}$  and the corresponding response  $\mathcal{X}$ . Since REVERT performs incremental summarization of reasoning steps, the training data must be structured such that summaries are interleaved with their respective reasoning segments. Formally,  $\mathcal{X}$  is represented as

$$\mathcal{X} = [\mathcal{X}_1, \dots, \mathcal{X}_n] \quad (1)$$

$$\mathcal{X}_k = [\mathcal{R}_k \langle \text{bov} \rangle \mathcal{V}_k \langle \text{eov} \rangle], \quad (2)$$

where  $\mathcal{R}_k$  is the segments of the  $k$ -th reasoning step, and  $\mathcal{V}_k$  is the verbalized text, enclosed by  $\langle \text{bov} \rangle$  and  $\langle \text{eov} \rangle$  tokens, as a speech-friendly summary of  $\mathcal{R}_k$ . Sometimes,  $\mathcal{R}_k$  consists of multiple reasoning segments, denoted as  $\mathcal{R}_k = [\mathcal{R}_k^1, \dots, \mathcal{R}_k^{m_k}]$ , where each segment is separated by delimiters  $\mathcal{D}$  (i.e.,  $\backslash n$ ), and  $m_k$  indicates the total number of segments.

**Dataset Construction Pipeline** Because no publicly available datasets conform to the required format, we propose a simple LLM-based pipeline to generate a dataset in our desired format with a standard QA dataset as input. Figure 3 presents an overview of the proposed pipeline. The pipeline consists of three steps: *solve*, *summarize*, and *scatter*. In the solve step, the reasoning LLM solves the user query using a standard chain-of-thought. In the summarize step, we generate a speech-friendly summary for the generated reasoning output. In the scatter step, we scatter the summaries across the reasoning process such that each summary appears immediately after its associated reasoning step, along with  $\langle \text{bov} \rangle$  and  $\langle \text{eov} \rangle$  tokens. We use the output of the scatter step as the training data. For all three steps, we employ gpt-4.1-mini-2025-04-11 as the processing model. More detailed procedures and prompts are provided in Appendix A.

**Objective** The training procedure for REVERT closely follows standard LLM finetuning. We begin by initializing REVERT with a pretrained LLM and finetune it using cross-entropy loss on the next-token prediction task, applied selectively to the training data described above. Importantly, since



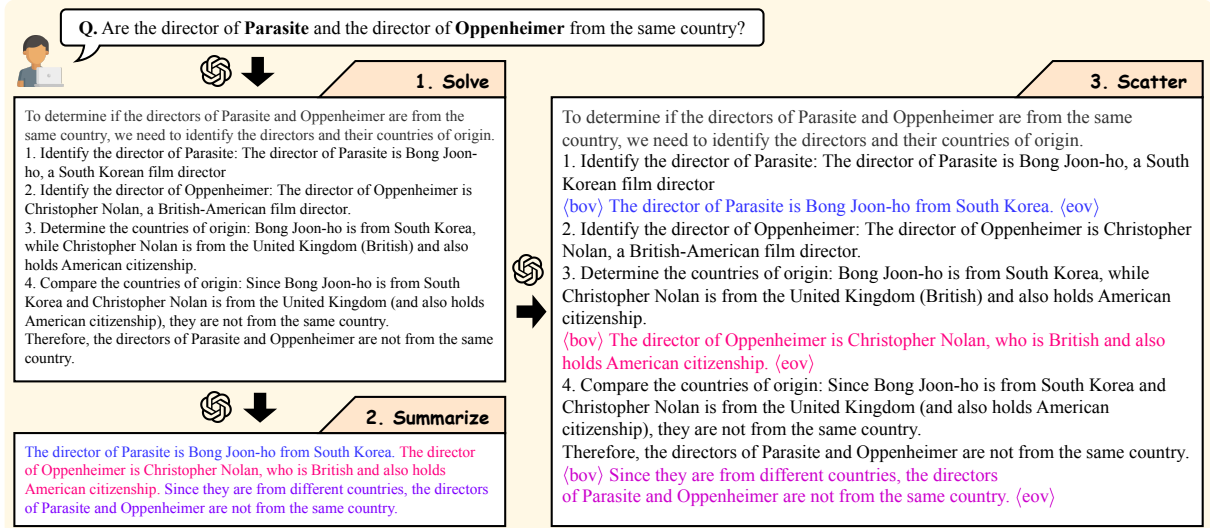


Figure 3: Data construction pipeline for training REVERT. Starting from a question, (1) **Solve**: we first generate a step-by-step reasoning process to derive the answer. (2) **Summarize**: we summarize key parts of the reasoning as speech-friendly utterances. (3) **Scatter**: we insert each utterance right after the reasoning segment it reflects, producing a sequence where internal reasoning and verbal explanations are interleaved.

REVERT is not required to perform the reasoning process itself, we compute the loss only within the verbalization segments of each sequence. For positions outside these verbalization segments, the model is trained to predict a special  $\langle \text{con} \rangle$  token, signaling that it is still in the reasoning phase.

Formally, let  $\mathcal{I}_{\text{VERBAL}}$  denote the set of token positions within verbalization segments, the set of token positions spanning from  $\langle \text{bov} \rangle$  to  $\langle \text{eov} \rangle$ , inclusive. Conversely, let  $\mathcal{I}_{\text{THINK}}$  represent the set of token positions outside  $\mathcal{I}_{\text{VERBAL}}$ , corresponding to the tokens used for LLM reasoning. The total loss is:

$$\mathcal{L}(\theta) = - \sum_{i \in \mathcal{I}_{\text{VERBAL}}} \log p_{\theta}(x_i | \mathcal{Q}, x_{<i}) \quad (3)$$

$$- \sum_{i \in \mathcal{I}_{\text{THINK}}} \log p_{\theta}(\langle \text{con} \rangle | \mathcal{Q}, x_{<i}). \quad (4)$$

Here,  $x_i$  is the  $i$ -th token in the response sequence  $\mathcal{X}$ , and  $p_{\theta}$  is the model’s output probability.

## 4 Experiments

We evaluate the effectiveness of our THINK-VERBALIZE-SPEAK framework and the verbalizer model across multiple experimental settings. Additional details are provided in Appendix B.2.

### 4.1 Models

We evaluate two versions of THINK-VERBALIZE-SPEAK: (1) **TVS+SEQ**, which performs reasoning followed by verbalization sequentially; and (2)

**TVS+REVERT**, in which the REVERT model incrementally verbalizes the reasoning outputs.

For comparison, we include several baselines based on the two-stage framework: (1) Chain-of-Thought (**CoT**) employs a standard zero-shot chain-of-thought prompting technique to elicit step-by-step reasoning. (2) Speech-Friendly Prompting (**SFP**) applies prompting strategies to encourage the model to generate concise, speech-appropriate outputs, following the guidelines established by Cho et al. (2024). (3) Speech-Friendly Finetuning (**SFF**) uses a finetuned model to directly produce speech-friendly responses. For finetuning, we use the same dataset as our model, but replace the output of the scatter step with that of the summarize step. Additionally, we include Qwen2.5-Omni-7B (Xu et al., 2025), an end-to-end spoken dialogue system finetuned to produce speech-friendly outputs, as a baseline for comparative analysis.

For the think model, we experiment with multiple LLMs, specifically Qwen2.5-7B-Instruct (Yang et al., 2024), Llama-3.1-8B-Instruct (Grattafiori et al., 2024), and gpt-4o-mini-2024-07-18 (OpenAI et al., 2024). For SEQ and REVERT model, we use Qwen2.5-3B-Instruct (Yang et al., 2024) as the base model and fine-tune it. For all models except Qwen2.5-Omni-7B, we employ gpt-4o-mini-tts (OpenAI, 2025) as the speak model to convert textual responses into speech.

## 4.2 Datasets

We consider the following three datasets for our evaluation setup: (1) GSM8K (Cobbe et al., 2021) focuses on arithmetic reasoning, based on grade-school level math problems. The solutions are generally straightforward and linear, involving simple, easy-to-follow steps without complex mathematical elements. (2) 2WikiMultiHopQA (Ho et al., 2020) requires multi-hop retrieval of Wikipedia documents to answer a question. While the dataset is not primarily designed to assess complex reasoning, multi-hop QA in a closed-book setting demands step-by-step reasoning abilities. (3) SciBench (Wang et al., 2024a) assesses college-level scientific problem-solving abilities. The solutions are often involve complex equations, formulas, and other components that are not easily communicated verbally.

We construct the training set of REVERT as a subset of examples from the GSM8K and 2WikiMultiHopQA training sets. SciBench remains unseen during training and serves to evaluate the model’s out-of-domain generalization capability.

## 4.3 Evaluation Procedure and Measures

**Automatic Reasoning Evaluation** We evaluate the reasoning capabilities of dialogue systems. Each system generates responses to the provided questions, and we assess the correctness of the final outputs using an LLM-as-a-judge framework. We report the accuracy for this evaluation.

**Automatic Speech-Friendliness Evaluation** We evaluate whether the responses from each system are suitable for verbal delivery. We adopt the four metrics also used by Cho et al. (2024). (1) Word count (WC) measures the overall conciseness of the response and is computed using simple whitespace delimitation. (2) Flesch Reading Ease (FRE) score assesses text readability based on the number of syllables per word and words per sentence. Although not directly related to speech, the FRE score is correlated with listenability. (3) Dependency depth (DD) is the maximum depth of the response dependency tree computed by Spacy dependency parser<sup>1</sup>. DD helps assess the sentence complexity. (4) Nonvocalizable character count (NV) evaluates the appropriateness of the response for verbal delivery by identifying the presence of nonvocalizable content.

<sup>1</sup><https://spacy.io/api/dependencyparser>

Criteria
<b>Naturalness:</b> Whether the response sounds like something a human would naturally say in conversation.
<b>Conciseness:</b> Whether the response delivers essential information without unnecessary verbosity.
<b>Understandability:</b> How easily the response can be comprehended when spoken aloud.
<b>Overall Quality:</b> Overall impression of the response’s quality and suitability for spoken delivery.

Table 1: Criteria for human evaluation of spoken responses. More details are in Appendix C.

**REVERT Latency Evaluation** We measure the time-to-response of THINK-VERBALIZE-SPEAK and evaluate the effectiveness of REVERT in reducing latency. Since we use a streaming TTS model, we focus on the time required to generate the first spoken output. Specifically, ( $T_1$ ) the time taken for the system to enter the verbalizing mode after receiving the user’s query, and ( $T_2$ ) the additional time required to produce the first verbalized segment after verbalization has started. We report latencies at the 50th percentile with Qwen2.5-3B-Instruct as the verbalizer. All experiments are conducted on the GSM8K dataset using the PyTorch transformers library with bfloat16 precision on an NVIDIA A6000 GPU.

**Human Evaluation** We conduct a human evaluation in which annotators on Amazon Mechanical Turk rate system responses on a 5-point Likert scale according to four criteria: naturalness, conciseness, understandability, and overall quality. Table 1 provides definitions for each criterion. We randomly sample 60 examples, 20 from each dataset, and collect annotations from three independent raters per example. Unlike previous evaluations that rely on textual assessment, this evaluation is *speech-based*.

## 5 Results and Discussion

### 5.1 Does speech-friendliness compromise models’ reasoning capabilities?

Table 2 presents the results of the automatic evaluations for the THINK-VERBALIZE-SPEAK model and the baseline systems. In most cases, the chain-of-thought strategy achieves the highest reasoning benchmark accuracies within each THINK model category, but demonstrates the lowest performance in speech-suitability evaluations. This indicates that the chain-of-thought strategy exhibits highly polarized performance with respect to reasoning

Models	(a) Accuracy (%)			(b) Speech-suitability				(c) Latency (s)	
	GSM8K	2MHQA	SciBench	WC(↓)	FRE(↑)	DD(↓)	NV(↓)	T <sub>1</sub>	T <sub>2</sub>
<b>Qwen2.5-Omni-7B</b>	84.53	14.30	20.95	101.7	90.90	5.24	0.78	-	-
<b>Qwen2.5-7B-Instruct</b>									
Chain-of-Thought	92.72	30.00	50.72	153.5	69.22	6.23	67.11	0.0	0.64
Speech-Friendly Prompting	87.57	26.60	45.09	87.11	84.97	5.45	11.04	0.0	0.46
Speech-Friendly Finetuning	68.69	32.70	21.97	44.90	88.32	4.28	0.035	0.0	0.47
<b>TVS (SEQ)</b>	93.18	29.75	47.40	42.15	88.71	4.23	0.005	8.08	0.43
<b>TVS (REVERT)</b>	92.65	30.00	47.25	44.02	88.40	4.21	0.024	2.72	0.45
<b>Llama-3.1-8B-Instruct</b>									
Chain-of-Thought	85.44	17.95	28.32	194.5	69.86	6.42	9.079	0.0	0.77
Speech-Friendly Prompting	83.70	16.45	22.83	101.7	87.99	5.33	2.887	0.0	0.34
Speech-Friendly Finetuning	65.13	42.50	14.02	48.05	88.43	4.25	0.034	0.0	0.35
<b>TVS (SEQ)</b>	85.44	22.25	26.01	43.94	88.72	4.27	0.026	7.19	0.44
<b>TVS (REVERT)</b>	85.29	19.10	27.80	44.95	88.89	4.20	0.043	2.67	0.44
<b>gpt-4o-mini-2024-07-18</b>									
Chain-of-Thought	94.84	39.60	55.64	175.4	67.40	6.37	74.69	-	-
Speech-Friendly Prompting	87.26	34.40	34.54	73.09	82.45	5.14	0.215	-	-
<b>TVS (SEQ)</b>	94.77	39.75	53.26	43.83	88.48	4.27	0.008	-	-
<b>TVS (REVERT)</b>	94.69	39.55	53.32	45.92	88.39	4.25	0.019	-	-

Table 2: Main results comparing different baselines and our proposed method (TVS (SEQ) or (REVERT) ) across three base THINK models. We report (a) task accuracy on GSM8K, 2WikiMultiHopQA, and SciBench; (b) speech-suitability scores using word count (WC), Flesch Reading Ease (FRE), dependency depth (DD), and number of non-vocal characters (NV); and (c) generation latency (T<sub>1</sub>, T<sub>2</sub>) at the 50<sup>th</sup> percentile. Speech-suitability scores and latencies are computed on the GSM8K test set. By decoupling thinking and verbalizing (TVS), we preserve reasoning capabilities while enhancing speech-friendliness. Furthermore, the introduction of the REVERT model significantly reduces latency. Results on other datasets are presented in Appendix D.

capabilities and speech-friendliness.

Therefore, we apply the two most widely used solutions to these issues: prompting and finetuning. While the speech-friendly prompting yields only a minimal decrease in reasoning benchmark accuracies, it resorts to chain-of-thought reasoning when faced with challenging questions, which in turn harms its speech-suitability scores. An example in Table 8 and human evaluation in Table 3 reveal similar issues. Despite receiving the highest overall scores on 2WikiMultiHopQA, its scores, especially the conciseness score, drop significantly on GSM8K and SciBench. Qwen2.5-Omni-7B also exhibits a similar trend, where its speech-friendliness diminishes with rising problem difficulty.

In contrast, the speech-friendly finetuning system receives high speech-friendliness scores but low reasoning benchmark scores. In other words, it yields highly intelligible responses but not intelligent ones. Notably, the system achieves the highest scores on the 2WikiMultiHopQA dataset. We attribute this to the model acquiring additional knowledge during training, as the dataset does not strictly separate train set and development set knowledge bases. Therefore, the high score is likely unrelated to the system’s reasoning capabilities.

These findings highlight a fundamental trade-off within the two-stage paradigm: optimizing for reasoning capability tends to degrade speech-suitability, and vice versa.

## 5.2 How does the explicit verbalization stage affect performance?

While this framework, by design, should mirror the accuracy scores of the THINK model’s chain-of-thought strategy, we observe a slight decrease in accuracy on the SciBench dataset. We attribute this to two possible factors: (1) out-of-domain characteristics and (2) inherent task difficulty. However, even with the drop in accuracy, both versions of our framework vastly outperform other baselines.

We also observe an anomalous result on the 2WikiMultiHopQA dataset for the Llama-3.1-8B-Instruct think model, where both the SEQ and REVERT variants outperform the chain-of-thought strategy. We attribute this to the same factor identified in the speech-friendly finetuning strategy issue, as all three systems share the same target text in the training data.

For speech-suitability measures, both SEQ and REVERT outperform all other baselines in automatic evaluation. In human evaluation, we analyze the results for each dataset. On 2WikiMultiHopQA,

Method	GSM8K				2WikiMultiHopQA				SciBench			
	Natu.	Conc.	Unde.	Over.	Natu.	Conc.	Unde.	Over.	Natu.	Conc.	Unde.	Over.
CoT	4.55 $\pm$ 0.09	3.72 $\pm$ 0.11	4.48 $\pm$ 0.09	4.28 $\pm$ 0.09	3.92 $\pm$ 0.16	2.92 $\pm$ 0.18	4.40 $\pm$ 0.11	3.53 $\pm$ 0.15	4.25 $\pm$ 0.12	3.10 $\pm$ 0.16	3.75 $\pm$ 0.12	4.08 $\pm$ 0.10
SFP	4.32 $\pm$ 0.12	3.67 $\pm$ 0.15	4.52 $\pm$ 0.12	4.17 $\pm$ 0.12	4.47 $\pm$ 0.10	4.23 $\pm$ 0.12	4.65 $\pm$ 0.08	4.35 $\pm$ 0.10	4.20 $\pm$ 0.11	3.02 $\pm$ 0.18	3.90 $\pm$ 0.11	4.03 $\pm$ 0.09
SFF	4.62 $\pm$ 0.09	4.33 $\pm$ 0.11	4.52 $\pm$ 0.10	4.33 $\pm$ 0.10	4.47 $\pm$ 0.12	4.17 $\pm$ 0.13	4.60 $\pm$ 0.10	4.32 $\pm$ 0.12	4.23 $\pm$ 0.15	4.00 $\pm$ 0.12	3.80 $\pm$ 0.12	4.10 $\pm$ 0.10
Qwen2.5-Omni-7B	3.40 $\pm$ 0.14	3.50 $\pm$ 0.14	3.72 $\pm$ 0.15	3.50 $\pm$ 0.14	4.03 $\pm$ 0.14	4.18 $\pm$ 0.10	4.47 $\pm$ 0.10	4.10 $\pm$ 0.11	4.00 $\pm$ 0.16	3.48 $\pm$ 0.15	3.70 $\pm$ 0.14	3.93 $\pm$ 0.14
TVS (SEQ)	4.62 $\pm$ 0.09	4.40 $\pm$ 0.10	4.55 $\pm$ 0.09	4.37 $\pm$ 0.09	4.35 $\pm$ 0.12	4.10 $\pm$ 0.11	4.58 $\pm$ 0.10	4.33 $\pm$ 0.10	4.40 $\pm$ 0.10	4.02 $\pm$ 0.11	3.82 $\pm$ 0.13	4.28 $\pm$ 0.09
TVS (REVERT)	4.58 $\pm$ 0.10	4.45 $\pm$ 0.11	4.23 $\pm$ 0.13	4.23 $\pm$ 0.12	4.39 $\pm$ 0.13	4.21 $\pm$ 0.13	4.63 $\pm$ 0.10	4.18 $\pm$ 0.11	4.45 $\pm$ 0.09	4.07 $\pm$ 0.12	3.87 $\pm$ 0.14	4.28 $\pm$ 0.09

Table 3: Human-annotated scores on spoken responses. "Natu.", "Conc.", "Unde.", and "Over." denote Naturalness, Conciseness, Understandability, and Overall Quality, respectively. Each score represents the mean and standard error of a 5-point Likert rating across three datasets. **Bold** indicates the highest score in each column, and underline indicates the lowest.

all systems score highly. In GSM8K and SciBench, all models lose points in conciseness and understandability. Regardless, SEQ and REVERT remain the top two models in terms of naturalness, conciseness, and overall quality.

In summary, the introduction of the VERBALIZE stage in THINK-VERBALIZE-SPEAK enables exceptional speech-friendliness with minimal compromise in the reasoning capabilities of the THINK model.

### 5.3 When should I use REVERT over SEQ?

As stated in Section 5.2, both SEQ and REVERT perform well across different datasets, with minimal differences in their effectiveness as verbalizers. The primary distinction between the two models lies in their latency. Specifically, SEQ waits for the reasoning process to complete before verbalizations, which requires approximately 8.08 seconds, as shown in Table 2(c). Such latency is unsuitable for real-time spoken conversation settings.

Conversely, REVERT incrementally processes verbalizable segments before the reasoning process is complete, receiving the first segment in an average of 2.72 seconds, a 66% reduction in latency compared to SEQ. In voice-interface conversations, this latency can be effectively masked by brief filler phrases such as "Let me think," making it acceptable for real-time applications. Therefore, REVERT achieves performance comparable to SEQ while significantly reducing latency, suggesting that REVERT is preferable for most real-time applications.

### 5.4 Does size matter?

We discuss the effect of the size of the REVERT model on its performance. Table 4 illustrates the performance of three differently sized REVERT models: Qwen2.5-0.5B-Instruct, Qwen2.5-1.5B-

REVERT size	Accuracy (%)		Speech-suitability	
	GSM8K	SciBench	WC ( $\downarrow$ )	FRE ( $\uparrow$ )
7B	92.7	50.7	–	–
3B	92.7	47.3	44.0	88.4
1.5B	92.7	46.8	45.3	88.9
0.5B	91.4	42.1	44.2	88.6

Table 4: Comparison of verbalization abilities across different REVERT model sizes. Speech-suitability scores are calculated on GSM8K.

Instruct, and Qwen2.5-3B-Instruct. The results indicate that the performance loss from decreasing model size is more pronounced for SciBench than for GSM8K, likely due to differences in task difficulty between the datasets. Notably, the speech suitability scores remain stable despite reductions in model size.

In conclusion, although model size affects REVERT’s performance, the degradation is not substantial. This suggests that smaller REVERT models remain a viable option in low-resource settings.

## 6 Conclusion

In this work, we address a critical gap between reasoning capability and speech-friendliness in spoken dialogue systems. We present the THINK-VERBALIZE-SPEAK framework, which decouples reasoning from verbalization to achieve both accuracy and naturalness in speech. Extensive automatic and human evaluations show that our framework enhances speech suitability with minimal compromise of reasoning capability across most benchmarks. Additionally, we introduce the REVERT model for incremental verbalization, which reduces latency compared to the sequential approach. Although our focus is on single-turn conversations, extending the framework to multi-turn or full-duplex interactions presents a promising avenue for future research.



## 7 Limitations

While our framework shows promising results, it has several limitations. First, it focuses on single-turn conversational settings and does not support multi-turn or full-duplex interactions, where reasoning and verbalization may need to occur in parallel with multiple user interactions. Extending the framework to handle such interactive scenarios remains an important direction for future work. Second, the current verbalization model does not provide control over the level of explanation detail. Supporting adjustable granularity—from brief summaries to step-by-step explanations—could improve adaptability to different user needs. Third, our work focuses on chain-of-thought reasoning, but extending it to other test-time computation methods with intermediate traces, such as multi-step retrieval or tool use, could broaden its applicability.

## 8 Potential Risks

Our framework introduces no additional epistemic or safety risks beyond those already present in the underlying reasoning model. This is because the verbalization model is designed solely to rephrase the outputs of a frozen, pretrained reasoning LLM into speech-friendly language without altering their content or logic. It performs no independent reasoning, decision-making, or content generation beyond linguistic reformulation. Consequently, factual inaccuracies, biases, or harmful outputs originate entirely from the reasoning model. The verbalization stage merely translates those outputs into a form more suitable for spoken communication. Thus, the overall risk profile of the system is bounded by that of the underlying reasoning model, and our model introduces no novel vulnerabilities.

## References

- Hyundong Justin Cho, Nicolaas Paul Jedema, Leonardo F. R. Ribeiro, Karishma Sharma, Pedro Szekely, Alessandro Moschitti, Ruben Janssen, and Jonathan May. 2024. Speechworthy instruction-tuned language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 10652–10670, Miami, Florida, USA. Association for Computational Linguistics.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.
- 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*. Technical report.
- Vardhan Dongre, Xiaocheng Yang, Emre Can Acikgoz, Suvodip Dey, Gokhan Tur, and Dilek Hakkani-Tür. 2024. Respect: Harmonizing reasoning, speaking, and acting towards building large language model-based conversational ai agents. *CoRR*.
- Qingkai Fang, Shoutao Guo, Yan Zhou, Zhengrui Ma, Shaolei Zhang, and Yang Feng. 2025. LLaMA-Omni: Seamless speech interaction with large language models. In *The Thirteenth International Conference on Learning Representations*.
- Heting Gao, Hang Shao, Xiong Wang, Chaofan Qiu, Yunhang Shen, Siqi Cai, Yuchen Shi, Zihan Xu, Zuwei Long, Yike Zhang, et al. 2025. Lucy: Linguistic understanding and control yielding early stage of her. *arXiv preprint arXiv:2501.16327*.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.
- Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Sieun Hyeon, Kyudan Jung, Nam-Joon Kim, Hyun Gon Ryu, and Jaeyoung Do. 2025. *Mathreader : Text-to-speech for mathematical documents*. *Preprint*, arXiv:2501.07088.
- Peter Indefrey and Willem JM Levelt. 2004. The spatial and temporal signatures of word production components. *Cognition*, 92(1-2):101–144.
- Shengpeng Ji, Yifu Chen, Minghui Fang, Jialong Zuo, Jingyu Lu, Hanting Wang, Ziyue Jiang, Long Zhou, Shujie Liu, Xize Cheng, et al. 2024. Wavchat: A survey of spoken dialogue models. *arXiv preprint arXiv:2411.13577*.
- Kushal Lakhotia, Eugene Kharitonov, Wei-Ning Hsu, Yossi Adi, Adam Polyak, Benjamin Bolte, Tu-Anh Nguyen, Jade Copet, Alexei Baevski, Abdelrahman Mohamed, et al. 2021. On generative spoken language modeling from raw audio. *Transactions of the Association for Computational Linguistics*, 9:1336–1354.

- Willem JM Levelt. 1993. *Speaking: From intention to articulation*.
- OpenAI. 2024. [Openai o1 system card](#). *Preprint*, arXiv:2412.16720.
- OpenAI. 2025. <https://platform.openai.com/docs/models/gpt-4o-mini-tts>. Accessed: 2025-05-16.
- Aaron OpenAI, Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.
- Ned T Sahin, Steven Pinker, Sydney S Cash, Donald Schomer, and Eric Halgren. 2009. Sequential processing of lexical, grammatical, and phonological information within broca’s area. *Science*, 326(5951):445–449.
- Xiaoxuan Wang, Ziniu Hu, Pan Lu, Yanqiao Zhu, Jieyu Zhang, Satyen Subramaniam, Arjun R. Loomba, Shichang Zhang, Yizhou Sun, and Wei Wang. 2024a. SciBench: Evaluating College-Level Scientific Problem-Solving Abilities of Large Language Models. In *Proceedings of the Forty-First International Conference on Machine Learning*.
- Xiong Wang, Yangze Li, Chaoyou Fu, Yunhang Shen, Lei Xie, Ke Li, Xing Sun, and Long Ma. 2024b. Freeze-omni: A smart and low latency speech-to-speech dialogue model with frozen llm. *arXiv preprint arXiv:2411.00774*.
- Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyang Jiang, et al. 2024c. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.
- Zhifei Xie and Changqiao Wu. 2024. Mini-omni: Language models can hear, talk while thinking in streaming. *arXiv preprint arXiv:2408.16725*.
- Jin Xu, Zhifang Guo, Jinzheng He, Hangrui Hu, Ting He, Shuai Bai, Keqin Chen, Jialin Wang, Yang Fan, Kai Dang, et al. 2025. Qwen2. 5-omni technical report. *arXiv preprint arXiv:2503.20215*.
- An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.
- 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](#). *Preprint*, arXiv:2412.02612.
- Dong Zhang, Shimin Li, Xin Zhang, Jun Zhan, Pengyu Wang, Yaqian Zhou, and Xipeng Qiu. 2023. [Speechgpt: Empowering large language models with intrinsic cross-modal conversational abilities](#). *Preprint*, arXiv:2305.11000.

## A Details of dataset

This section provides a detailed procedure on how our training dataset was generated. First, we source a set of raw question-answer pairs from the GSM8K and 2WikiMultiHopQA.

From the GSM8K training set, we use all 7473 examples. From the 2WikiMultiHopQA dataset, we sample 1,000 examples from each of the 4 data types: inference, comparison, bridge\_comparison, and compositional.

### A.1 Solve, Summarize, Scatter

**Solve** In this step, we simply induce step-by-step reasoning process, using standard zero-shot chain-of-thought prompting.

**Summarize** In this step, we generate a summary of the reasoning process from *solve*. We impose the following constraints on the resulting summary:

- The summary must contain all essential information from the reasoning process.
- The summary must follow the same logical progression as the reasoning process.
- The summary must not repeat information provided in the question.
- The summary must be speech-friendly and free of complex sentences or hard-to-read words.

Because enforcing all constraints simultaneously in a single instruction yields suboptimal results, we adopt a progressive approach, providing the language model with one constraint at a time.

**Scatter** In this step, we distribute the summary throughout the reasoning process, placing each summary segment immediately after its corresponding reasoning segment. To encourage fine-grained control over the placement of summary segments, we manually label 16 samples and use them as few-shot examples.

## B Details of Experimental Setup

### B.1 Prompts

This section outlines the specific prompts used in our experiments, including those for baseline methods and our proposed verbalizer. For the chain-of-thought (CoT) reasoning experiments, we adopt the system prompt illustrated in Figure 4. In the

case of speech-friendly prompting and finetuning, we follow the instruction template shown in Figure 6. Our proposed verbalizer (both the SEQ and REVERT) uses the prompt presented in Figure 5. For experiments involving Qwen2.5-Omni-7B, we employ the default system prompt provided by the model.

#### Chain-of-Thought Prompting (CoT)

You are a helpful assistant that provides a step-by-step reasoning process before arriving at the final answer.

Figure 4: A system prompt designed for chain-of-thought (CoT) prompting.

#### Speech-Friendly Prompting (SFP)

You are a voice assistant that responds in a way that is easy to understand when spoken aloud. Your responses should be concise, clear, and listener-friendly. Avoid using equations, LaTeX, or complex symbols that are hard to pronounce or understand in speech.

Figure 5: A system prompt designed for speech-friendly prompting (SFP) or finetuning (SFF).

#### REVERT

Your task is to provide step-by-step reasoning with incremental, speech-friendly summaries. You will be given a question and must reason through it step by step. Each time you generate the token <bov>, provide a clear and concise summary of the reasoning so far, suitable for spoken delivery. Do not include equations, LaTeX, or complex symbols in these summaries. Conclude each summary with the token <eov>. Ensure each summary connects naturally to the one before it.

Figure 6: A system prompt designed for our verbalizer.

### B.2 Details of Training

We finetune our verbalization model for SEQ and REVERT model using Qwen2.5-3B-Instruct with full-parameter optimization. All models are trained for one epoch with 4 A6000GPU, totaling 1.3k steps (within 1 hour) with a batch size of 8. For optimization, we employ the AdamW optimizer with a learning rate of  $2 \times 10^{-5}$ , a cosine learning rate

Prompts for Data Construction	
<b>Solve</b>	Provide a step-by-step reasoning process before arriving at the final answer.
<b>Summarize</b>	<p>Write a clear, concise, and speech-friendly summary of the provided analysis. Ensure the summary flows naturally when read aloud, avoiding complex sentences, mathematical equations or awkward phrasing. Follow the analysis’s logical progression, presenting key points in the same order and context. Include only essential steps, omitting unnecessary details, boilerplate text, or repetitions. The general tone of the summary must match that of the original analysis. The summary must be appropriate as the response to the original question. Do not apply LaTeX or markdown formatting in your summary.</p> <p>First, extract the essential information from the analysis.</p> <p>Now, make sure the logical progression of the summary matches the order of the analysis, without adding or removing information.</p> <p>Now, remove all parts that are repeating the information from the original question.</p> <p>Now, make sure the content is speech-friendly by breaking up complex sentences and avoiding long and hard-to-read words.</p>
<b>Scatter</b>	<p>Combine an analysis with its corresponding summary by interleaving summary segments with relevant analysis portions.</p> <p>Ensure the summary segments immediately follow the equivalent content within the analysis and are enclosed with ‘&lt;bov&gt;’ and ‘&lt;eov&gt;’ tags.</p> <p># Steps</p> <ol style="list-style-type: none"> <li>1. <b>Receive analysis and summary</b>: Identify sections in both the analysis and the summary.</li> <li>2. <b>Match content segments</b>: Pair each summary segment with the corresponding analysis segment to ensure logical flow.</li> <li>3. <b>Interleave content</b>: Insert summary segments after the matching sections of the analysis, surrounded by ‘&lt;bov&gt;’ and ‘&lt;eov&gt;’ tags.</li> </ol> <p># Output Format</p> <p>The output should maintain the structure of the analysis, with summary segments appropriately interleaved. Each summary segment must be enclosed in ‘&lt;bov&gt;’ and ‘&lt;eov&gt;’ tags immediately after the equivalent analysis content.</p> <p># Notes</p> <ul style="list-style-type: none"> <li>- Maintain the logical sequence of both the analysis and summary.</li> <li>- Ensure clarity in how summary elements align with the analysis.</li> <li>- Avoid rephrasing the given segments; use them verbatim for consistency and accuracy.</li> <li>- The summary and analysis must not change from the original text.</li> </ul>

Table 5: Prompts designed for data construction.

scheduler, and a warmup ratio of 0.1. The optimizer parameters are set to  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$ , and a weight decay of 0.1. For speech-friendly finetuning (SFF), we finetune Qwen2.5-7B-Instruct and Llama-3.1-8B-Instruct using LoRA with  $r = 16$  and  $\alpha = 16$ . and other training configurations are kept identical to those described above.

### B.3 Inference

We use top- $p$  sampling with a temperature of 0.1 and a nucleus probability  $p = 0.95$  for all response generation for inference. For the REVERT model, we employ greedy decoding for next-token prediction to determine whether to initiate verbalization (i.e., generate the <bov> or <con> token). Upon receiving the final reasoning segment, the verbalizer is manually appended with the <bov> token rather than relying on sampling.

### B.4 LLM-based answer verification

We utilize an LLM-based answer verification method to overcome the limitations of rule-based evaluation. In the context of speech-friendly responses, model outputs are expected to be clear and easily understandable, but they often deviate from structured formats and may normalize numerical

answers (e.g., \$ 25  $\rightarrow$  twenty-five dollars). These characteristics make exact matching and rule-based answer extraction unreliable.

Therefore, we use gpt-4.1-mini-2025-04-11 to automatically assess answer correctness. As illustrated in Figure 7, the verifier is prompted with the question, a model-generated response, and the corresponding ground-truth answer.

## C Details of Human Evaluation

In this section, we provide comprehensive details regarding our human evaluation protocol.

### C.1 Datasets and Models.

We evaluate 60 examples, with a random sample of 20 from the three target datasets: GSM8K, 2Wiki-MultiHopQA, and SciBench. Each example is evaluated independently by multiple annotators. We use the Qwen2.5-7B-Instruct model as the thinking LLM across all evaluated systems, including CoT, SFP, SFF, Qwen2.5-Omni-7B for THINK-SPEAK baselines, and our proposed approaches.

### C.2 Evaluation Criteria.

Each output is evaluated along four key dimensions. We provided annotators with the following, more



### Answer Verification Prompt

**[SYSTEM]**

Given a ground-truth answer and a submitted answer, tell me if the submitted answer is correct. Disregard any units, formatting, etc. In your explanation, first extract the final answer of the submitted answer and then compare it with the ground truth.

You must only use the ground truth answer to determine the correctness of the submitted answer. The validity of the ground truth answer must not be questioned. Note that the submitted answer must directly and explicitly answer the question. Any implicit answering should be considered incorrect.

**[USER]**

Question:

{question}

Ground-Truth Answer:

{ground\_truth}

Submitted Answer:

{submitted\_answer}

Figure 7: A prompt designed for LLM-based answer verification.

detailed definitions for each factor, which offer additional guidance beyond the brief descriptions in Table 1.

- **Naturalness:** measures whether the response sounds like something a real person would say in a conversation. This is NOT a measure of acoustic quality—focus on the wording and phrasing, not the voice.
- **Conciseness:** measures whether the response gets to the point without including unnecessary or excessive information. Focus on whether the response is brief and relevant, or if it feels too long or contains details that aren’t needed.
- **Understandability:** measures how clearly the response communicates its meaning. Focus on whether you can easily grasp what the response is trying to say, without getting lost or confused by the way the information is presented.
- **Overall Quality:** measures your general impression of the response, taking into account all aspects such as clarity, naturalness, and conciseness. Focus on how well the response works as a whole.

**Annotation Procedure.** We recruited annotators via Amazon Mechanical Turk (MTurk). For each data point, we collected ratings from three independent workers to mitigate subjectivity. Annotators were instructed as follows:

- Carefully read the question and listen to the speech-based response before rating.
- Rate each evaluation criterion on a 1–5 Likert scale, where 1 represents the lowest and 5 the highest quality.
- For each criterion, provide a brief explanation to justify the assigned score.

Compensation was set at 0.5\$ per example for GSM8K and 2WikiMultiHopQA, and 0.7\$ per example for SciBench, reflecting the varying complexity and required annotation effort. Explanations were manually reviewed to filter out low-effort or inconsistent responses.

## D Further Analysis

In this section, we present additional analyses of our experimental results to complement the main findings discussed in the paper. We provide qualitative examples for each dataset and method, along with detailed dataset-wise results of speech-suitability scores and human evaluation outcomes.

### D.1 Qualitative Results

To provide deeper insight into our framework THINK-VERBALIZE-SPEAK and the REVERT model, we present representative qualitative examples from each evaluation dataset. All examples use Qwen2.5-7B-Instruct as the reasoning model. Specifically, Table 7 presents results on

Models	WC(↓)			FRE(↑)			DD(↓)			NV(↓)		
	2MHQA	GSM8K	SciBench	2MHQA	GSM8K	SciBench	2MHQA	GSM8K	SciBench	2MHQA	GSM8K	SciBench
<b>Qwen2.5-Omni-7B</b>	49.0	101.7	138.1	74.8	90.9	72.6	5.51	5.24	5.95	0.004	0.78	30.8
<b>Qwen2.5-7B-Instruct</b>												
CoT	149.4	194.5	436.0	51.0	69.9	61.6	6.51	6.42	7.18	3.737	9.079	327.2
SFP	53.2	101.7	326.5	59.2	88.0	68.9	6.44	5.33	6.22	0.449	2.887	122.6
SFF	40.6	48.0	80.8	70.8	88.4	75.8	4.48	4.25	5.13	0.011	0.034	5.6
<b>TVS (SEQ)</b>	46.6	43.9	90.0	71.7	88.7	74.7	4.80	4.27	5.49	0.021	0.026	5.7
<b>TVS (REVERT)</b>	30.5	44.9	78.1	69.5	88.9	74.6	4.79	4.20	5.37	0.075	0.043	3.2
<b>Llama-3.1-8B-Instruct</b>												
CoT	145.6	153.4	296.2	58.4	69.2	61.8	5.85	6.23	7.08	14.604	67.109	384.7
SFP	27.9	87.1	212.4	61.8	85.0	68.4	5.70	5.45	6.69	0.155	11.043	243.0
SFF	39.9	44.9	84.6	71.7	88.3	75.0	4.48	4.28	5.29	0.015	0.035	12.5
<b>TVS (SEQ)</b>	45.9	42.1	79.3	70.9	88.7	72.9	4.62	4.23	5.52	0.019	0.005	4.8
<b>TVS (REVERT)</b>	41.6	44.0	76.9	71.1	88.4	72.8	4.55	4.21	5.43	0.173	0.024	1.9
<b>gpt-4o-mini-2024-07-18</b>												
CoT	104.6	175.4	291.0	53.8	67.4	63.5	5.85	6.37	6.55	11.463	74.687	352.6
SFP	20.4	73.1	213.8	58.4	82.5	57.3	5.79	5.14	6.37	0.011	0.215	54.0
<b>TVS (SEQ)</b>	43.1	43.8	79.4	69.7	88.5	72.8	4.63	4.27	5.52	0.014	0.008	6.4
<b>TVS (REVERT)</b>	32.2	45.9	74.8	70.6	88.4	73.1	4.63	4.25	5.52	0.041	0.019	2.3

Table 6: Additional speech-suitability scores across all three datasets, comparing various approaches: chain-of-thought (CoT), speech-friendly prompting (SFP), finetuning (SFF), Qwen2.5-Omni-7B, and our proposed methods (TVS+SEQ, TVS+REVERT). Results show consistent trends across all datasets. As we move from 2WikiMultiHopQA (denoted as 2MHQA) to GSM8K and then SciBench, the tasks increasingly demand stronger reasoning capabilities. Correspondingly, the length of test-time reasoning grows and overall speech-suitability decreases.

GSM8K, Table 8 on 2WikiMultiHopQA, and Tables 9 and 10 on SciBench. These examples highlight the strengths and limitations of various approaches in terms of both reasoning capability and speech-friendliness. Our framework demonstrates balanced performance, producing outputs that are logically sound and well-suited for verbal communication.

## D.2 Speech-suitability Scores

We provide additional speech-suitability scores for the remaining datasets in Table 6. Consistent patterns are observed across datasets: models explicitly guided to generate speech-friendly outputs (SFP, SFF, Qwen2.5-Omni-7B) tend to produce less verbose responses (WC↓), exhibit improved readability (FRE↑), and reduce the use of non-vocal elements (NV↓), such as LaTeX expressions and equations, relative to standard chain-of-thought generations.

Nevertheless, we find that applying the speech-friendly prompt (Figure 6) to standard text-based language models does not consistently yield outputs suitable for spoken delivery. This limitation is particularly pronounced in SciBench, where tasks involve complex reasoning and numerical computation. In such cases, models often fail to con-

form to the intended spoken style, suggesting that prompt-based approaches alone may be insufficient for speech adaptation in highly technical domains.

Sample text responses for various methods on GSM8K	
Question	Each class in a school has 20 students. There are 3 classes. The school as a whole is 50% boys and 50% girls. The first class has 15 girls. The second class has 12 girls. How many boys are in the third class?
Answer	17
Chain-of-Thought	<p>To determine the number of boys in the third class, we need to follow these steps:</p> <p><b>1. Calculate the total number of students in the school:</b></p> <ul style="list-style-type: none"> <li>- Each class has 20 students.</li> <li>- There are 3 classes.</li> </ul> $\text{Total number of students} = 20 \times 3 = 60$ <p><b>2. Determine the number of boys and girls in the school:</b></p> <ul style="list-style-type: none"> <li>- The school is 50% boys and 50% girls.</li> </ul> $\text{Number of boys} = \frac{50}{100} \times 60 = 30$ $\text{Number of girls} = \frac{50}{100} \times 60 = 30$ <p><b>3. Find the number of girls in the first and second classes:</b></p> <ul style="list-style-type: none"> <li>- The first class has 15 girls.</li> <li>- The second class has 12 girls.</li> </ul> <p><b>4. Calculate the total number of girls in the first two classes:</b></p> $\text{Total number of girls in first and second classes} = 15 + 12 = 27$ <p><b>5. Determine the number of girls in the third class:</b></p> <ul style="list-style-type: none"> <li>- Total number of girls in the school is 30.</li> </ul> $\text{Number of girls in the third class} = 30 - 27 = 3$ <p><b>6. Calculate the number of boys in the third class:</b></p> <ul style="list-style-type: none"> <li>- Each class has 20 students.</li> <li>- The third class has 3 girls.</li> </ul> $\text{Number of boys in the third class} = 20 - 3 = 17$ <p>Thus, the number of boys in the third class is <span style="border: 1px solid black; padding: 0 2px;">17</span>.</p>
Speech-Friendly Prompting	<p>In each class, there are 20 students. Since the school is 50% boys and 50% girls, each class also has 10 boys and 10 girls. The first class has 15 girls, which means it has 5 boys (since it's supposed to have 10 girls, but there are 15, so <math>20 - 15 = 5</math> boys). The second class has 12 girls, which means it has 8 boys (since it's supposed to have 10 girls, but there are 12, so <math>20 - 12 = 8</math> boys). So far, we have:</p> <ul style="list-style-type: none"> <li>- First class: 5 boys    - Second class: 8 boys</li> </ul> <p>That's a total of 13 boys in the first two classes. Since each class is supposed to have 10 boys, the third class must have 7 boys (because <math>10 - 3 = 7</math>, where 3 is the number of boys already counted in the first two classes). So, there are <span style="color: red;">7</span> boys in the third class.</p>
Speech-Friendly Finetuning	<p>The first class has 15 girls, so it has 5 boys. The second class has 12 girls, which means it has 8 boys. Together, these two classes have 23 boys. Since each class has 20 students, there are 60 boys in total. That means the third class must have <span style="color: red;">37</span> boys.</p>
TVS (SEQ)	<p>The school has 30 boys and 30 girls. The first class has 15 girls. The second class has 12 girls. Together, they have 27 girls. That means the third class has 3 girls. Since each class has 20 students, the third class has 17 boys. So, there are <span style="color: green;">17</span> boys in the third class.</p>
TVS (REVERT)	<p>There are 60 students in total. The school is half boys and half girls. That means there are 30 boys. The first class has 15 girls. The second class has 12 girls. Together, they have 27 girls. If we take 27 away from 30, we get 3 girls in the third class. Since each class has 20 students, the third class has <span style="color: green;">17</span> boys.</p>

Table 7: Sample generation results for various methods on GSM8K. Our models, TVS (SEQ) and TVS (REVERT), consistently produce accurate answers with logically sound, step-by-step reasoning, while maintaining high speech-suitableness. In contrast, speech-friendly baseline methods frequently generate answers that are not only incorrect but also logically flawed.

Sample text responses for various methods on 2WikiMultiHopQA	
Question	Which country John Of Bohemia’s mother is from?
Answer	Germany
Chain-of-Thought	<p>To determine which country John of Bohemia’s mother is from, we need to follow these steps:</p> <ol style="list-style-type: none"> <li><b>Identify John of Bohemia:</b> John of Bohemia was a Bohemian nobleman and king who reigned as King of Bohemia from 1347 to 1349.</li> <li><b>Research his family background:</b> John of Bohemia was born in Prague, Bohemia (now part of the Czech Republic) in 1319. His father was Wenceslaus II, Holy Roman Emperor, and his mother was Elisabeth of Bavaria.</li> <li><b>Determine Elisabeth of Bavaria’s origin:</b> Elisabeth of Bavaria was born in 1294 in Landshut, Bavaria, which is now part of Germany. She was a member of the House of Wittelsbach.</li> </ol> <p>Given this information, we can conclude that John of Bohemia’s mother, Elisabeth of Bavaria, is from Bavaria, which is a region in modern-day Germany. Therefore, the country John of Bohemia’s mother is from is <b>Germany</b>.</p>
Speech-Friendly Prompting	John of Bohemia’s mother was from France. Specifically, she was from the region of <b>Champagne</b> .
Speech-Friendly Finetuning	John of Bohemia was a king of Bohemia. His mother was Margaret of Durazzo. She was the daughter of Stephen Dušan, who ruled Serbia. So, John of Bohemia’s mother was from <b>Serbia</b> .
TVS (SEQ)	John of Bohemia was a Bohemian nobleman and king. His mother was Elisabeth of Bavaria. She was born in Landshut, Bavaria. That area is now part of Germany. So, John of Bohemia’s mother is from <b>Germany</b> .
TVS (REVERT)	John of Bohemia was a Bohemian nobleman and king. He was born in Prague and had a Bavarian mother named Elisabeth. Elisabeth was born in Landshut, Bavaria. So, John of Bohemia’s mother is from <b>Germany</b> .

Table 8: Sample generation results for various methods on 2WikiMultiHopQA. Compared to baselines, both TVS (SEQ) and TVS (REVERT) deliver more accurate reasoning and maintain higher speech-suitableness in their responses. While baseline methods frequently produce incorrect answers or include irrelevant information, our models consistently provide factually correct and well-structured explanations that are both clear and suitable for spoken delivery. This highlights the effectiveness of our approach in balancing reasoning capability with speech-oriented generation quality.



Sample text responses for various methods on SciBench (1)	
Question	Let $X$ and $Y$ be independent with distributions $N(5, 16)$ and $N(6, 9)$ , respectively. Evaluate $P(X > Y) = P(X - Y > 0)$ .
Answer	0.4207
Chain-of-Thought	<p>To evaluate <math>P(X &gt; Y)</math>, we need to find the probability that <math>X - Y &gt; 0</math>. Given that <math>X</math> and <math>Y</math> are independent normal random variables, we can determine the distribution of <math>X - Y</math>.</p> <p>### Step 1: Determine the Distribution of <math>X - Y</math></p> <p>Since <math>X \sim N(5, 16)</math> and <math>Y \sim N(6, 9)</math>:</p> <ul style="list-style-type: none"> <li>- The mean of <math>X</math> is <math>\mu_X = 5</math>.</li> <li>- The variance of <math>X</math> is <math>\sigma_X^2 = 16</math>, so the standard deviation is <math>\sigma_X = 4</math>.</li> <li>- The mean of <math>Y</math> is <math>\mu_Y = 6</math>.</li> <li>- The variance of <math>Y</math> is <math>\sigma_Y^2 = 9</math>, so the standard deviation is <math>\sigma_Y = 3</math>.</li> </ul> <p>For the difference <math>X - Y</math>:</p> <ul style="list-style-type: none"> <li>- The mean of <math>X - Y</math> is:</li> </ul> $\mu_{X-Y} = \mu_X - \mu_Y = 5 - 6 = -1$ <ul style="list-style-type: none"> <li>- The variance of <math>X - Y</math> is:</li> </ul> $\sigma_{X-Y}^2 = \sigma_X^2 + \sigma_Y^2 = 16 + 9 = 25$ <p>Therefore, the standard deviation of <math>X - Y</math> is:</p> $\sigma_{X-Y} = \sqrt{25} = 5$ <p>Thus, <math>X - Y \sim N(-1, 25)</math>.</p> <p>### Step 2: Standardize <math>X - Y</math></p> <p>We need to find <math>P(X - Y &gt; 0)</math>. To do this, we standardize <math>X - Y</math>:</p> $Z = \frac{(X - Y) - \mu_{X-Y}}{\sigma_{X-Y}} = \frac{(X - Y) - (-1)}{5} = \frac{X - Y + 1}{5}$ <p>We want to find:</p> $P(X - Y > 0) = P\left(\frac{X - Y + 1}{5} > \frac{0 + 1}{5}\right) = P\left(Z > \frac{1}{5}\right)$ <p>### Step 3: Use the Standard Normal Distribution</p> <p>The variable <math>Z</math> follows a standard normal distribution <math>N(0, 1)</math>. We need to find <math>P(Z &gt; 0.2)</math>. Using standard normal distribution tables or a calculator, we find:</p> $P(Z > 0.2) = 1 - P(Z \leq 0.2)$ <p>From standard normal tables, <math>P(Z \leq 0.2) \approx 0.5793</math>. Therefore:</p> $P(Z > 0.2) = 1 - 0.5793 = 0.4207$ <p>### Final Answer</p> $P(X > Y) = P(X - Y > 0) = \boxed{0.4207}$

Table 9: Step-by-step reasoning output from the Chain-of-Thought baseline on SciBench. This example illustrates a correct and complete logical progression, resulting in the correct numerical answer.

Sample text responses for various methods on SciBench (2)	
Question	Let $X$ and $Y$ be independent with distributions $N(5, 16)$ and $N(6, 9)$ , respectively. Evaluate $P(X > Y) = P(X - Y > 0)$ .
Answer	0.4207
Speech-Friendly Prompting	<p>To find <math>P(X &gt; Y)</math>, we need to look at the distribution of <math>X - Y</math>. Since <math>X</math> and <math>Y</math> are independent, the difference <math>X - Y</math> will also follow a normal distribution. First, let's determine the mean and variance of <math>X - Y</math>:</p> <ul style="list-style-type: none"> <li>- The mean of <math>X</math> is 5, and the mean of <math>Y</math> is 6. So, the mean of <math>X - Y</math> is: <math display="block">\mu_{X-Y} = \mu_X - \mu_Y = 5 - 6 = -1</math> </li> <li>- The variance of <math>X</math> is 16, and the variance of <math>Y</math> is 9. Since <math>X</math> and <math>Y</math> are independent, the variance of <math>X - Y</math> is: <math display="block">\sigma_{X-Y}^2 = \sigma_X^2 + \sigma_Y^2 = 16 + 9 = 25</math> </li> </ul> <p>Therefore, the standard deviation of <math>X - Y</math> is:</p> $\sigma_{X-Y} = \sqrt{25} = 5$ <p>So, <math>X - Y</math> follows a normal distribution <math>N(-1, 25)</math>. Now, we want to find <math>P(X - Y &gt; 0)</math>. This is equivalent to finding the probability that a normally distributed random variable with mean <math>-1</math> and standard deviation <math>5</math> is greater than <math>0</math>. We can standardize this by converting it to a standard normal distribution:</p> $Z = \frac{(X - Y) - (-1)}{5} = \frac{X - Y + 1}{5}$ <p>We need to find <math>P(Z &gt; \frac{0+1}{5}) = P(Z &gt; 0.2)</math>. Using standard normal tables or a calculator, we find that:</p> $P(Z > 0.2) \approx 0.4207$ <p>Therefore, <math>P(X &gt; Y) = P(X - Y &gt; 0) \approx \mathbf{0.4207}</math>.</p>
Speech-Friendly Finetuning	The difference between $X$ and $Y$ , $X$ minus $Y$ , follows a normal distribution. Its mean is $5$ minus $6$ , which is negative $1$ . Its variance is the sum of the variances of $X$ and $Y$ , so it's $25$ . To find $P(X$ greater than $Y)$ , we look at $P(X$ minus $Y$ greater than $0)$ . This is the same as finding the probability that a standard normal variable is greater than $1$ divided by $5$ . That value is about $0.3085$ . So, $P(X$ greater than $Y)$ is approximately <b>0.3085</b> .
TVS (SEQ)	The difference between $X$ and $Y$ is $X$ minus $Y$ . The mean of $X$ minus $Y$ is negative one. The variance is $25$ , so the standard deviation is five. To find $P(X$ minus $Y > 0)$ , we standardize $X$ minus $Y$ . This gives us $Z$ equals $(X$ minus $Y$ plus one) divided by five. We want to find $P(Z > 0.2)$ . Since $Z$ is a standard normal variable, we use the standard normal distribution. From the table, $P(Z > 0.2)$ is about $0.4207$ . So, $P(X > Y)$ is <b>0.4207</b> .
TVS (REVERT)	The mean of $X$ minus $Y$ is $5$ minus $6$ , which is $-1$ . The variance is the sum of the variances, $16$ plus $9$ , which equals $25$ . The standard deviation is the square root of $25$ , which is $5$ . To find $P(X$ minus $Y$ is greater than $0)$ , we standardize $X$ minus $Y$ . This gives us a standard normal distribution with mean $0$ and standard deviation $5$ . We want to find the probability that this standard normal is greater than $0.2$ . Using the standard normal distribution, we find that $P(X$ minus $Y$ is greater than $0)$ is about <b>0.42</b> .

Table 10: Sample generation results from various methods on SciBench. Our models, TVS (SEQ) and TVS (REVERT), consistently produce accurate answers with logically sound and precise numerical reasoning, while maintaining clarity and suitability for spoken delivery. In contrast, baseline methods frequently exhibit logical errors or numerical calculation mistakes. Interestingly, even when prompted for concise and speech-friendly responses, the models tend to generate still structured and verbose outputs on SciBench, due to the inherent complexity of scientific questions.