PlanRAG-Audio: Planning and Retrieval Augmented Generation for Long-form Audio Understanding

Anonymous ACL submission

Abstract

Long-form audio understanding poses significant challenges for large audio language models (LALMs) due to the extreme length of audio sequences and the need to reason over heterogeneous acoustic cues distributed over time, such as speech content, speaker identity, emotion, and sound events. To address these challenges, we propose **PlanRAG-Audio**, a planning-based retrieval-augmented generation framework for scalable long-form audio understanding. Rather than having audio LALMs process entire recordings directly, PlanRAG-Audio explicitly plans which modalities and temporal spans are required for a given query, and retrieves only query-relevant information from a structured text and audio database. This retrieval planning enables effective reasoning over complex, cross-domain audio queries while substantially reducing the input length passed to the large language models. Experiments across a wide range of speech/audio retrieval demonstrate that PlanRAG-Audio improves reasoning accuracy and stabilizes performance as audio duration increases by decoupling inference cost from raw audio length.

1 Introduction

Spoken interaction has become a key modality for human–machine communication, driving the development of large audio language models (LALMs) that jointly reason over linguistic and non-verbal acoustic cues (Kong et al., 2024; Défossez et al., 2024; Yu et al., 2025; Tian et al., 2025), yet long-form speech remains extremely challenging due to the resulting growth in both token length and multimodal complexity. For example, a one-hour lecture corresponds to roughly 12 k text tokens but over 100 k speech tokens based on Gemini (Google, 2023), leading to severe computational and memory bottlenecks when conversations span minutes or hours. Enabling LALMs to efficiently understand and reason over such extended long-form speech data remains an open challenge.

Retrieval-augmented generation (RAG) (Lewis et al., 2020) has proven effective in text domains for mitigating hallucination and improving reasoning by retrieving external evidence. Given the finite context window of large language models (LLM), it is infeasible to include all potentially relevant documents or web data directly within the model input. RAG addresses this limitation by *selectively retrieving* only the information necessary to answer a user query, allowing the model to focus on a compact and relevant subset of knowledge rather than processing the entire corpus. As such, RAG serves as a mechanism for efficient information extraction under constrained context length. Extending this paradigm to audio provides a natural route toward scalable speech, text, and acoustic understanding: instead of examining every audio token in long speech recordings, a system can retrieve only the most relevant semantic, paralinguistic, or acoustic segments needed for reasoning. Recent work on RAG has shown that explicitly planning the retrieval process before execution can improve efficiency and attribution for complex queries (Lee et al., 2024; Verma et al., 2025).

However, most prior work on long-form audio understanding still relies on automatic speech recognition (ASR) or audio captioning pipelines, which convert speech to text before applying conventional NLP models (Shankar et al., 2024; Prasad et al., 2023; Ahia et al., 2025; Arora et al., 2025). This transcript/text-centric design ignores prosody, speaker variation, and non-verbal acoustic events that are essential for human communication. Other approaches use audio-text contrastive embeddings (Chen et al., 2025; Johnson et al., 2024; Zhu et al., 2024) or direct audio retrieval (Abdelnour et al., 2023; Sudarsanam and Virtanen, 2023), but these methods remain limited to short clips and fail to account for long-range dependencies across modalities.

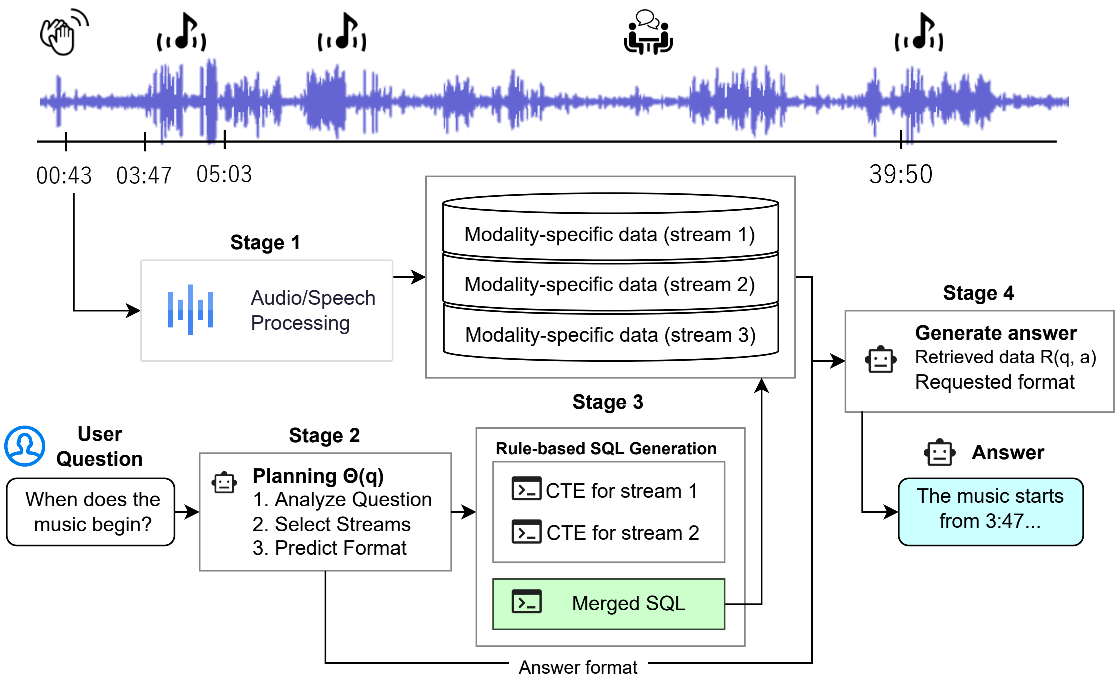

Figure 1: Overview of question-driven multimodal retrieval over long-form audio. Given a user question grounded in audio (derived from audio and speech processing; see Figure 2 and Section 3.2), the system plans the required reasoning steps by analyzing the question, selecting relevant streams, and predicting the requested output format. For each selected stream, the planned retrieval is compiled into stream-specific Common Table Expressions (CTEs), which are then composed into a unified SQL query via a hybrid LLM–rule-based SQL generator. The retrieved segments are finally aggregated and passed to the generation model to produce the answer.

Importantly, the difficulty of long-form audio understanding does not arise solely from the increased input length. As recordings extend over longer time spans, they inherently induce queries that require *compositional reasoning across heterogeneous acoustic cues*. Such queries often depend on the interaction between spoken content, speaker, and non-verbal sound events distributed over time. For example, a user may ask to summarize segments of a radio broadcast by jointly considering spoken reports and background acoustic events, while associating each summary with its corresponding time span. These problems cannot be adequately addressed by text-only representations, such as those obtained via ASR, as their semantics emerge from cross-domain dependencies and long-range temporal structure. Consequently, effective long-form audio understanding requires a framework that can selectively reason over multiple modalities while preserving their temporal relationships.

To overcome these limitations, we propose **PlanRAG-Audio**, a *planning-based retrieval-augmented generation framework* for long-form audio understanding. Instead of retrieving all available information, PlanRAG-Audio first analyzes a user query to plan which modalities (e.g., spoken content, speaker information, emotional cues, and non-verbal acoustic events), which temporal spans, and which retrieval constraints are required to answer the query. It then retrieves only the corresponding information from a database. This explicit planning stage prevents redundant retrieval, allowing the model to reason efficiently over hours of audio without requiring an extremely large input window. As illustrated in Figure 1, PlanRAG-Audio uses an LLM to analyze a user query and produce a retrieval plan, which is then compiled into structured database queries to retrieve relevant audio metadata, before a generation LLM aggregates the retrieved evidence to produce the final answer.

Our contributions are as follows.

- We propose **PlanRAG-Audio**, a planning-based retrieval-augmented framework for long-form audio understanding that performs *planning before retrieval*.

- We show that PlanRAG-Audio enables effective reasoning over long-form audio by selectively retrieving information from multiple modalities while preserving temporal structure.

- We demonstrate that PlanRAG-Audio handles a wide range of long-form audio understanding tasks, from base tasks such as QA and diarization to advanced and compositional reasoning tasks, in a zero-shot manner without task-specific prompt engineering or manually crafted SQL queries.

2 Related Work

We review related work on audio information retrieval, retrieval-augmented generation, and long-form audio evaluation.

2.1 Audio Information Retrieval

Prior work on audio information retrieval can be broadly categorized into several directions. One line of research focuses on learning transferable audio representations for tagging and retrieval (Kong et al., 2020; Elizalde et al., 2023; Zhu et al., 2024). Another line addresses spoken question answering and spoken document retrieval by combining speech recognition, as well as ASR-free formulations for spoken QA (Lin et al., 2024, 2022; Abdelnour et al., 2023; Sudarsanam and Virtanen, 2023). More recently, retrieval-augmented generation has been extended to speech and audio, integrating audio representations or transcripts with RAG-style pipelines (Min et al., 2025; Chen et al., 2025; Maben et al., 2025; Semnani et al., 2023). Despite these advances, existing approaches are typically task-specific or limited to short audio segments, and rely on transcript-centric or embedding-based retrieval. As a result, they lack support for structured, modality-aware retrieval over long-form audio.

2.2 Retrieval Augmented Generation

Recent work on retrieval-augmented generation has moved beyond the simple *retrieve–then–generate* paradigm toward more structured approaches that incorporate iteration and planning to handle complex queries (Asai et al., 2024; Liu et al., 2024; Lee et al., 2024; Verma et al., 2025). Related ideas have also been explored in long-context domains such as video understanding, where hierarchical decomposition and step-by-step retrieval are used to scale RAG to longer inputs (Liu et al., 2025b). Extending these ideas to long-form audio remains an open challenge, as effective retrieval must account for speaker changes, emotional dynamics, and non-speech acoustic events over extended time spans.

2.3 Long-Form Audio Evaluation

Research on spoken and long-form audio understanding has evolved from synthetic datasets to realistic, open-domain benchmarks. Early efforts on acoustic QA (AQA) such as CLEAR (Abdelnour et al., 2019), inspired by CLEVR (Johnson et al., 2017), and its extension NAAQA (Abdelnour et al., 2023), introduced controlled synthetic acoustic scenes to study compositional reasoning over sound attributes. Later datasets such as Clotho-AQA (Lipping et al., 2022) and Audiopedia (Penamakuri et al., 2025) extended AQA to crowdsourced and knowledge-intensive settings. However, most of these benchmarks focus on short clips and do not support multi-hour reasoning or multimodal alignment.

More recently, BLAB (Ahia et al., 2025) introduced a benchmark for long-form audio understanding, but remains limited in reproducibility and evaluation scope. In contrast, our work evaluates long-form audio understanding using fully open datasets and aligns the evaluation domains with tasks commonly studied in recent Interspeech conferences, as detailed in Appendix C.

3 Proposed Framework

In this section, we introduce PlanRAG-Audio by formulating long-form audio understanding as a retrieval planning problem, and describe both the planning mechanism and the underlying audio database design that enable efficient and scalable retrieval over extended audio.

3.1 PlanRAG-Audio

3.1.1 Stage 1: Audio and Speech Processing

As illustrated in Figure 1, PlanRAG-Audio first converts raw audio into a set of modality-specific representations through audio and speech processing. Each representation corresponds to an independent retrieval stream, such as speech transcripts, speaker segments, sound events, or emotional cues. These streams are stored in a structured audio database and serve as the basic units for downstream retrieval. Details of the audio ingestion and stream construction process are described in Section 3.2.

3.1.2 Stage 2: Retrieval Planning

Given a user question q, PlanRAG-Audio formulates retrieval as a planning problem, where a planning LLM analyzes the query and produces a structured retrieval plan $\Theta(q)$. Because $\Theta(q)$ is generated via constrained decoding under a fixed schema, the planning stage is effectively deterministic and does not produce invalid retrieval plans. This planning step explicitly determines what information should be retrieved before executing any retrieval operations. Concretely, the retrieval plan $\Theta(q)$ specifies: 1) which modality-specific representations are required; 2) what filters are applied to each stream; 3) how multiple streams are joined; 4) which fields are returned from the merged SQL results; 5) what schema the final generation LLM must follow. Example 1 shows a simplified example of a retrieval plan $\Theta(q)$. For clarity, the example omits implementation-specific details and retains only the fields necessary to illustrate the core planning decisions, including stream selection, filtering, fusion, retrieval outputs, and the generation schema.

Example 1: Retrieval Planning Contract

```
{
 "streams": [ // (1)
  "transcription", "speaker"
 ],
 "filters": { // (2)
  "text": "employment",
  "speaker": "SPEAKER_02"
 },
 "fusion": { // (3)
  "anchor": "transcript"
 },
 "output": { // (4)
  "return_fields": [
   "start","end","speaker","text"
  ]
 },
 "answer_schema": { // (5)
  "properties": {
   "answer": {
    "type": "string",
    "enum": ["A", "B", "C", "D"]
   }
  },
  "required": ["answer"],
 }
}
```

3.1.3 Stage 3: Structured Retrieval

Given the retrieval plan $\Theta(q)$ from Stage 2, a rule-based SQL generator deterministically compiles it into an executable merged SQL query $Q(\Theta(q))$, constructed using stream selection, filtering, fusion, and the output contract (items 1–4 in Stage 2). The query is executed against the audio database $D(a)$ for the target recording a, yielding the retrieved segments $R(q, a)$:

$$R(q, a) = \mathrm{Exec}\big(Q(\Theta(q)),\ D(a)\big), \qquad (1)$$

where $\mathrm{Exec}(\cdot, \cdot)$ denotes a deterministic database execution operator.

Example 2 illustrates how the retrieval plan in Example 1 is compiled into a structured SQL query. The `WITH` clause defines stream-specific Common Table Expressions (CTEs) ((A), (C)) with their corresponding filters ((B), (D)), while the final `SELECT` projects the requested fields (E) and merges the streams via a temporal join (F), realizing the fusion strategy specified in the plan. We use a simple nearest-neighbor temporal fusion strategy based on timestamp alignment; implementation details are provided in Appendix D. Because each stream is translated into an independent CTE with its own filters, the resulting query is modular and naturally scalable to additional modalities. We adopt a simple keyword-based retrieval mechanism in this work, as our goal is to isolate the effect of explicit retrieval planning for complex long-form audio reasoning, including both cross-modality and single-modality inference.

Example 2: Simplified Merged SQL

```
WITH
tx AS (  -- (A) transcript stream
  SELECT start, end, text
  FROM transcription
  WHERE text ILIKE '%employment%'
    -- (B) text filter
),
sp AS (  -- (C) speaker stream
  SELECT start, end, label
  FROM speaker
  WHERE label = 'SPEAKER_02'
    -- (D) speaker filter
)
SELECT  -- (E) output projection
  tx.start, tx.end, sp.label, tx.text
FROM tx
JOIN sp ON temporal_overlap(tx, sp);
  -- (F) stream fusion
```

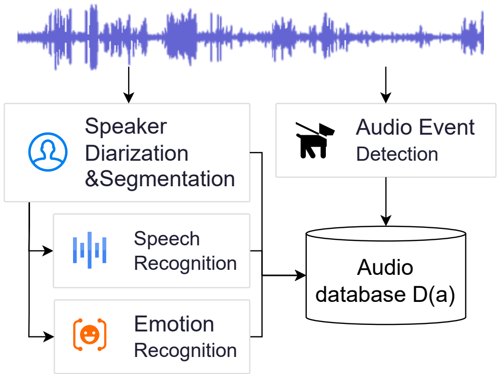

Figure 2: Audio database construction. Raw audio is processed by task-specific modules to construct a structured, time-aligned audio database $D(a)$ consisting of modality-specific metadata streams.

3.1.4 Stage 4: Answer Generation

The execution of the merged SQL query yields a set of relevant segments $R(q, a)$ from the audio database for the target recording a. These segments are provided to a generation LLM together with an explicit output schema, and the model produces the final answer while adhering to the schema constraints specified during planning.

3.2 Audio Database Construction

Figure 2 illustrates the construction of the structured audio database $D(a)$ from raw audio. The pipeline begins with speaker diarization, which produces a sequence of speaker-homogeneous temporal segments. These diarization timestamps define the fundamental alignment units used throughout the database construction process.

Given these shared segment boundaries, speech transcription and emotion recognition are applied to exactly the same temporal spans. As a result, transcript, speaker, and emotion streams share identical start and end times, as illustrated by the record examples in Table 1. This design enables cross-stream alignment, allowing text content to be directly matched with speaker identity and emotional cues using simple timestamp-based joins.

In contrast, sound event streams are generated at a different temporal resolution using a sliding-window audio tagging approach. Unlike speech-centric streams, acoustic events do not necessarily depend on conversational structure or speaker boundaries. Therefore, event predictions are produced independently of diarization segments and stored with their own start and end times. As with emotion, event labels are stored as label–score pairs in `JSONB` fields, following a unified schema.

Table 1: Examples of records stored in the audio database $D(a)$, where each stream consists of time-aligned records; label–score pairs for `emotion` and `sound_event` are shown in simplified form.

Stream	Start (s)	End (s)	Example
`transcript`	20.50	22.10	He talks about it
`speaker`	20.50	22.10	SPEAKER_07
`emotion`	20.50	22.10	Neutral (0.58), $\cdots$
`sound_event`	22.00	27.00	Speech (0.87), $\cdots$

4 Experimental Setup

Our experiment is designed to evaluate whether explicit retrieval planning enables reliable and scalable understanding of long-form audio in a zero-shot setting. We decompose evaluation into two levels. *Base tasks* assess fundamental audio understanding capabilities, including semantic understanding, speaker diarization, emotion recognition, and sound event detection, while controlling for input duration. *Advanced tasks* require additional reasoning beyond direct retrieval, such as counting, temporal ordering, and compositional constraints across modalities. Motivated by the limitations of existing long-form audio benchmarks discussed in Section 2.3, we design our evaluation to emphasize reproducibility, domain relevance, and scalable reasoning over extended audio.

4.1 Datasets

Base tasks We use only publicly available datasets to construct an evaluation set for each capability for the reproduction purpose. For semantic understanding, we construct long-form recordings from LibriSpeech (Panayotov et al., 2015) and evaluate question answering using LibriSQA (Zhao et al., 2024), which is built on the `train-clean-360` subset and includes both abstractive QA (QA-1) and multiple-choice QA (MCQA). For speaker diarization (SD), we generate long-form recordings by cropping or concatenating meeting audio from AMI (McCowan et al., 2005). For summarization, we use AMI recordings segmented into non-overlapping 10-minute windows and generate reference summaries by prompting ChatGPT with the corresponding reference transcriptions to produce 5–7 sentence abstractive summaries. Emotion recognition (ER) is evaluated

Table 2: Representative query examples for each task. These queries illustrate the user-level intent of each task, while the full prompts used for LLM inference are provided in Appendix A

Task	Example Query
	Base Task
QA-1	Who is Bela and why was no single arm is able to knock him down?
MCQA	What did the woman do to try to get the man's attention? A) ...
Summarization	Please provide an abstractive summary of the meeting segment between 0 and 600 seconds.
Diarization	Perform speaker diarization between 300 and 600 seconds.
Emotion	Analyze the audio between 325.41 and 332.23 seconds and respond with the emotion
SED	Detect occurrences of the following sound event label(s): Flamenco
	Advanced Task
Event Ordering	Determine the order of first occurrence for: (1) Music (2) Bird flight, flapping wings (3) Change ringing
Speaker count	You should count the number of speakers starting from 300 sec to 600 sec.
Speaker-Constrained QA	You should work on the utterance from speaker 439. What does the woman say to the executioner? A)...

on the test-1 split of MSP-Podcast (Busso et al., 2025), which contains natural podcast recordings annotated with categorical emotion labels. For sound event detection (SED), we construct extended recordings from VoxPopuli (Wang et al., 2021) and insert short AudioSet (Gemmeke et al., 2017) clips as target events, following the needle-in-a-haystack paradigm used in Gemini 1.5 evaluation.

To control evaluation scale, we generate one question per 5 minutes of audio, resulting in 1000 questions per task, except for summarization where we use 100 questions. For SD and summarization, queries are generated by partitioning each recording into non-overlapping temporal windows sampled sequentially from start to end, using 5-minute windows for SD and 10-minute windows for summarization. For long-form evaluation, we construct audio inputs with durations of 10, 30, 60, 300, and 540 minutes. All questions are instantiated from task-specific templates, with representative query examples provided in Appendix A.

Advanced Tasks For advanced tasks, we construct evaluation datasets by transforming base-task datasets to require additional inference over the same audio. Specifically, for SD, we replace base-task queries with questions that require counting the number of distinct speakers appearing in a 10-minute recording. Similarly, for SED, we reuse the original SED evaluation data (six events per 30-minute recording), randomly selecting three events to form event-ordering questions based on annotated onset times. Due to this formulation, each recording yields a single counting question; we therefore construct this evaluation using 100 recordings, resulting in 100 ordering samples.

Table 3: Model configurations used in our experiments

Model	**Version/Variation**	**Params**
Qwen	Qwen3-4B-Instruct-2507	4B
Gemini	Gemini 2.5 Flash	undisclosed
Voxtral	Voxtral-Mini-3B-2507	5B
ASR	OWSM-CTC v4 medium	1.01B
SED	BEATs iter3+, AS2M finetuned	90M
SD	Pyannote, community-1	8.1M
ER	Odyssey 2024 SER baseline	316M

We design compositional tasks that combine multiple base capabilities within a single query. In speaker-constrained question answering, a target speaker is specified and the system must answer using only utterances produced by that speaker. We generate both answerable and unanswerable cases by leveraging speaker annotations associated with existing QA pairs: for answerable cases, the original speaker is specified, while for unanswerable cases, a different speaker appearing in the same recording is specified, requiring the system to abstain. We construct this task using 60-minute recordings, which contain 3-4 speakers and enable meaningful speaker-constrained reasoning.[1]

4.2 Models

Table 3 summarizes the models and configurations used in our experiments. We adopt OWSM-CTC v4 as the ASR backbone due to its connectionist temporal classification (CTC)-only design, which avoids language-model memorization and enables fast, fully open multilingual recognition. For audio understanding tasks, we employ standard pretrained models for SD, SED, and ER, while using `Qwen3-4B-Instruct` as the primary genera-

[1]Data and code will be released upon acceptance.

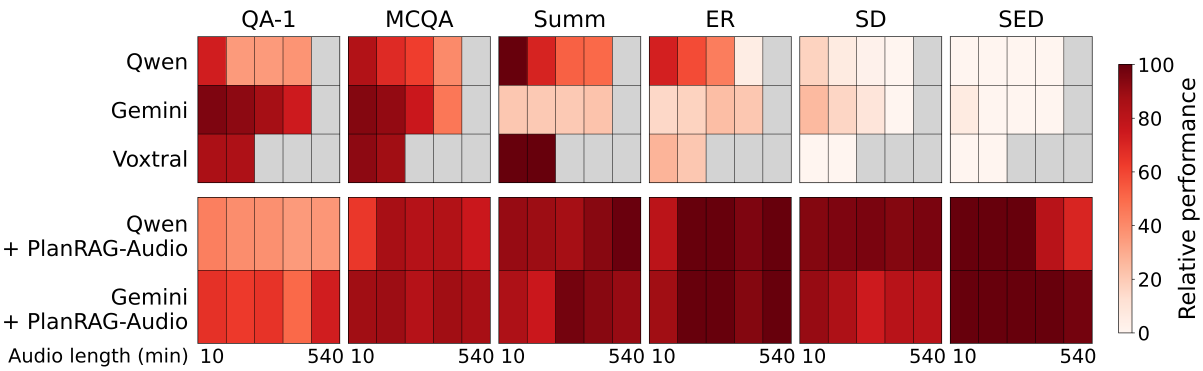

Figure 3: Relative performance of base tasks under long-form audio inputs. Results are for QA-1, MCQA, Summ, ER, SD, and SED across audio context lengths (10, 30, 60, 300, and 540 minutes). All scores are normalized to a 0–100 scale per task, with error-based metrics inverted (e.g., $100 - \text{DER}$) so that higher values indicate better performance. Gray cells indicate unsupported or failed settings. Exact values are provided in Appendix B.

tion model. As long-context baselines, we evaluate Gemini 2.5 Flash and Voxtral (Liu et al., 2025a) using their default inference settings, where the full audio input is provided without retrieval or segmentation.

4.3 Metrics

We report task-specific metrics following standard evaluation protocols. For QA-1 (LibriSQA Part I), we evaluate generated answers using Rouge-L, while accuracy is reported for MCQA (Part II). For summarization, we evaluate system outputs using Rouge-L against reference summaries generated by ChatGPT from the ground-truth transcript of the target time span specified in each query. For SD, we report Diarization Error Rate (DER). For ER, we report macro F1-score. For SED, we report event-level F1-score (Mesaros et al., 2016) using a fixed 5-second onset tolerance without enforcing end-time localization. To enable unified visualization across tasks, we normalize all metrics to a common 0–100 scale. For metrics where higher is better, scores are linearly scaled by the task-specific topline; for error-based metrics such as DER, we apply $(100 - \text{DER})$ before normalization.

The topline for each task is defined as an oracle upper bound obtained by applying a pretrained model directly to the ground-truth–aligned segment without long-form context or retrieval. Specifically, QA and summarization toplines are computed by applying Qwen3-4B model to the ground-truth answer text or transcript segment; SD and ER toplines use the corresponding pretrained models. For SED, BEATs performs clip-level classification without explicit timestamps, so the topline is computed on the evaluation window using macro F1-score.

Table 4: Average number of input tokens passed to the LLM for MCQA with 60-min recordings.

Model	Avg. Tokens
Gemini	115.2k
Gemini + PlanRAG-Audio	0.9k
Qwen + PlanRAG-Audio	1.2k

We report relative performance in the main figures, with absolute values provided in the Appendix B.

For advanced tasks, we report exact-match accuracy for speaker counting, Spearman's rank correlation for sound event ordering, and accuracy for speaker-constrained MCQA, with abstention accuracy for unanswerable cases. For all tasks, outputs that do not conform to the required response format or cannot be parsed are treated as incorrect. For QA, since it is sensitive to retrieval quality, we report accuracy over parseable outputs to isolate reasoning given retrieved evidence; end-to-end results are reported in the Appendix B.

5 Results

5.1 Base Task Results

The evaluation results for base tasks are shown in Figure 3. Without retrieval planning, performance degrades as audio duration increases for both Qwen and Gemini, with audio-only tasks such as SD, ER, and SED being particularly challenging. Voxtral performs well on text-centric tasks but fails on non-text-based ones. In contrast, PlanRAG-Audio stabilizes performance across input lengths by retrieving only query-relevant segments, effectively decoupling inference cost from raw audio duration

Table 5: Performance on single-modality reasoning tasks. Speaker counting and event ordering are evaluated using exact match accuracy and Spearman's rank correlation, respectively.

Model	Speaker Count	Event Order
Voxtral	9.17	-0.10
Gemini	14.20	0.30
+ PlanRAG-Audio	69.40	0.68
Qwen + PlanRAG-Audio	36.66	0.34

Table 6: Speaker-constrained MCQA. We report QA accuracy for answerable cases and abstention accuracy for non-answerable cases. **SC** indicates whether speaker constraints are applied.

Model	SC	QA Acc.	Abst. Acc.
Without PlanRAG-Audio			
Gemini		58.83	–
	✓	68.13	0.54
With PlanRAG-Audio			
Gemini		65.00	–
	✓	70.96	94.90
Qwen		65.09	–
	✓	67.59	82.20

and enabling reasoning over audio-derived signals.

Despite Gemini's long-context support, we observe notable degradation on long recordings, especially for speaker diarization. Although the maximum output length is uniformly set to 4096 tokens, Gemini often produces incomplete or malformed outputs, causing parsing failures. Across audio durations from 10 to 540 minutes, 17.92% of diarization outputs cannot be successfully parsed and are therefore counted as incorrect. As shown in Table 4, Gemini processes the full speech input (e.g., 115k+ tokens for 60 minutes), whereas PlanRAG-Audio reduces the effective input to 1k tokens via retrieval, keeping the LLM input size nearly constant and avoiding long-form accuracy degradation.

5.2 Advanced Task Results

Single-modality tasks Table 5 reports results on reasoning-intensive single-modality tasks, speaker counting and sound event ordering. Without PlanRAG-Audio, both Gemini and Voxtral perform poorly on speaker counting, while Gemini exhibits limited but non-trivial capability on event ordering, reflecting its long-context modeling capacity, and Voxtral fails to capture temporal order. Applying PlanRAG-Audio substantially improves both tasks, most notably for Gemini, where speaker counting accuracy increases from 14.20% to 69.40% and Spearman's rank correlation for event ordering increases from 0.30 to 0.68. These gains stem from how retrieval planning restructures reasoning: for speaker counting, the retrieved results explicitly include speaker identifiers for each segment, reducing the task to counting the number of unique speaker labels, while for event ordering, the LLM is provided with event labels and their associated start and end timestamps and only needs to sort them by temporal order. Compared to long-context models that must implicitly infer speaker identities or event boundaries directly from raw audio, PlanRAG-Audio externalizes temporal structure and symbolic attributes through retrieval, thereby transforming the original reasoning task into a simpler and reliable post-retrieval problem for the LLM.

Compositional tasks Table 6 reports results on speaker-constrained multiple-choice QA. Without PlanRAG-Audio, Gemini shows limited abstention ability under speaker constraints. With PlanRAG-Audio, QA accuracy is preserved while abstention accuracy is substantially improved, reaching 94.90% for Gemini. Qwen with PlanRAG-Audio achieves comparable constrained QA accuracy (67.59%) and 82.20% abstention accuracy. These results demonstrate that PlanRAG-Audio correctly identifies the speaker specified in the question and retrieves evidence exclusively from the corresponding speaker stream, enabling reliable speaker-conditioned reasoning and abstention without task-specific SQL or handcrafted rules.

6 Conclusion

We presented **PlanRAG-Audio**, a planning-based retrieval-augmented generation framework for scalable long-form audio understanding. By planning which modalities, temporal spans, and output requirements are needed and retrieving only the relevant data. PlanRAG-Audio expresses complex cross-domain audio reasoning within a unified framework, without relying on task-specific prompts or bespoke query logic, and naturally extends to new modalities and longer audio streams.

Limitations

Our evaluation of Gemini is subject to practical limitations imposed by the publicly available API. Although Gemini advertises support for long-context inputs of up to several hours, we were unable to reliably evaluate 9-hour audio recordings. Despite our efforts in prompt tuning and output format specification, Gemini frequently produced incomplete responses or outputs that did not conform to the expected JSON schema, preventing parsing and evaluation.

We adopt a simple keyword-based retrieval mechanism to isolate the effect of retrieval planning. More sophisticated retrievers may improve retrieval recall and particularly benefit text-centric tasks such as QA.

From an ethical and risk perspective, PlanRAG-Audio does not introduce new modeling assumptions beyond those of the pretrained components it relies on. As a result, potential risks present in pretrained models also apply to our framework.

The Use of AI Assistant

Parts of this manuscript were edited for clarity and language using an AI-based writing assistant. The authors take full responsibility for the content.

References

Jerome Abdelnour, Giampiero Salvi, and Jean Rouat. 2019. Clear: A dataset for compositional language and elementary acoustic reasoning.

Jérôme Abdelnour, Jean Rouat, and Giampiero Salvi. 2023. Naaqa: A neural architecture for acoustic question answering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Orevaoghene Ahia, Martijn Bartelds, Kabir Ahuja, Hila Gonen, Valentin Hofmann, Siddhant Arora, Shuyue Stella Li, Vishal Puttagunta, Mofetoluwa Adeyemi, Charishma Buchireddy, Ben Walls, Noah Bennett, Shinji Watanabe, Noah A. Smith, Yulia Tsvetkov, and Sachin Kumar. 2025. Blab: Brutally long audio bench. *Preprint*, arXiv:2505.03054.

Siddhant Arora, Haidar Khan, Kai Sun, Xin Luna Dong, Sajal Choudhary, Seungwhan Moon, Xinyuan Zhang, Adithya Sagar, Surya Teja Appini, Kaushik Patnaik, Sanat Sharma, Shinji Watanabe, Anuj Kumar, Ahmed Aly, Yue Liu, Florian Metze, and Zhaojiang Lin. 2025. Stream rag: Instant and accurate spoken dialogue systems with streaming tool usage. *Preprint*, arXiv:2510.02044.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*.

Carlos Busso, Reza Lotfian, Kusha Sridhar, Ali N. Salman, Wei-Cheng Lin, Lucas Goncalves, Srinivas Parthasarathy, Abinay Reddy Naini, Seong-Gyun Leem, Luz Martinez-Lucas, Huang-Cheng Chou, and Pravin Mote. 2025. The msp-podcast corpus. *Preprint*, arXiv:2509.09791.

Yifu Chen, Shengpeng Ji, Haoxiao Wang, Ziqing Wang, Siyu Chen, Jinzheng He, Jin Xu, and Zhou Zhao. 2025. WavRAG: Audio-integrated retrieval augmented generation for spoken dialogue models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics.

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

Benjamin Elizalde, Soham Deshmukh, Mahmoud Al Ismail, and Huaming Wang. 2023. Clap learning audio concepts from natural language supervision. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE.

Jort F. Gemmeke, Daniel P. W. Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R. Channing Moore, Manoj Plakal, and Marvin Ritter. 2017. Audio set: An ontology and human-labeled dataset for audio events. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Gemini Team Google. 2023. et al. gemini: a family of highly capable multimodal models.

Alexander Johnson, Peter Plantinga, Pheobe Sun, Swaroop Gadiyaram, Abenezer Girma, and Ahmad Emami. 2024. Efficient SQA from Long Audio Contexts: A Policy-driven Approach. In *Interspeech 2024*.

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross Girshick. 2017. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Qiuqiang Kong, Yin Cao, Turab Iqbal, Yuxuan Wang, Wenwu Wang, and Mark D. Plumbley. 2020. Panns: Large-scale pretrained audio neural networks for audio pattern recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.

Zhifeng Kong, Arushi Goel, Rohan Badlani, Wei Ping, Rafael Valle, and Bryan Catanzaro. 2024. Audio flamingo: A novel audio language model with few-shot learning and dialogue abilities. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*.

Myeonghwa Lee, Seonho An, and {Min Soo} Kim. 2024. Planrag: A plan-then-retrieval augmented generation for generative large language models as decision makers. NAACL 2024. Association for Computational Linguistics (ACL).

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20.

Chyi-Jiunn Lin, Guan-Ting Lin, Yung-Sung Chuang, Wei-Lun Wu, Shang-Wen Li, Abdelrahman Mohamed, Hung-Yi Lee, and Lin-Shan Lee. 2024. Speechdpr: End-to-end spoken passage retrieval for open-domain spoken question answering. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Guan-Ting Lin, Yung-Sung Chuang, Ho-Lam Chung, Shu wen Yang, Hsuan-Jui Chen, Shuyan Annie Dong, Shang-Wen Li, Abdelrahman Mohamed, Hung yi Lee, and Lin shan Lee. 2022. Dual: Discrete spoken unit adaptive learning for textless spoken question answering. In *Interspeech 2022*.

Samuel Lipping, Parthasaarathy Sudarsanam, Konstantinos Drossos, and Tuomas Virtanen. 2022. Clotho-aqa: A crowdsourced dataset for audio question answering. In *2022 30th European Signal Processing Conference (EUSIPCO)*.

Alexander H. Liu, Andy Ehrenberg, Andy Lo, Clément Denoix, Corentin Barreau, Guillaume Lample, Jean-Malo Delignon, Khyathi Raghavi Chandu, Patrick von Platen, Pavankumar Reddy Muddireddy, Sanchit Gandhi, Soham Ghosh, Srijan Mishra, Thomas Foubert, Abhinav Rastogi, Adam Yang, Albert Q. Jiang, Alexandre Sablayrolles, Amélie Héliou, and 87 others. 2025a. Voxtral. *Preprint*, arXiv:2507.13264.

Heng Liu, Siru Jiang, Fangyun Duan, Yongzhe Lyu, Xiusong Wang, Hanlin Ge, and Chao Liang. 2025b. Cadencerag: Context-aware and dependency-enhanced retrieval augmented generation for holistic video understanding. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*.

Yanming Liu, Xinyue Peng, Xuhong Zhang, Weihao Liu, Jianwei Yin, Jiannan Cao, and Tianyu Du. 2024. RA-ISF: Learning to answer and understand from retrieval augmentation via iterative self-feedback. In *Findings of the Association for Computational Linguistics: ACL 2024*. Association for Computational Linguistics.

Leander Melroy Maben, Gayathri Ganesh Lakshmy, Srijith Radhakrishnan, Siddhant Arora, and Shinji Watanabe. 2025. Aura: Agent for understanding, reasoning, and automated tool use in voice-driven tasks. *Preprint*, arXiv:2506.23049.

I. McCowan, J. Carletta, W. Kraaij, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, M. Kronenthal, G. Lathoud, M. Lincoln, A. Lisowska, W. Post, Dennis Reidsma, and P. Wellner. 2005. The ami meeting corpus. In *Proceedings of Measuring Behavior 2005, 5th International Conference on Methods and Techniques in Behavioral Research*. Noldus Information Technology.

Annamaria Mesaros, Toni Heittola, and Tuomas Virtanen. 2016. Metrics for polyphonic sound event detection. *Applied Sciences*, 6(6):162.

Do June Min, Karel Mundnich, Andy Lapastora, Erfan Soltanmohammadi, Srikanth Ronanki, and Kyu Han. 2025. Speech retrieval-augmented generation without automatic speech recognition. In *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: An asr corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Abhirama Subramanyam Penamakuri, Kiran Chhatre, and Akshat Jain. 2025. Audiopedia: Audio qa with knowledge. In *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Archiki Prasad, Trung Bui, Seunghyun Yoon, Hanieh Deilamsalehy, Franck Dernoncourt, and Mohit Bansal. 2023. MeetingQA: Extractive question-answering on meeting transcripts. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics.

Sina Semnani, Violet Yao, Heidi Zhang, and Monica Lam. 2023. WikiChat: Stopping the hallucination of large language model chatbots by few-shot grounding on Wikipedia. In *Findings of the Association for Computational Linguistics: EMNLP 2023*. Association for Computational Linguistics.

Natarajan Balaji Shankar, Alexander Johnson, Christina Chance, Hariram Veeramani, and Abeer Alwan. 2024. Coraal qa: A dataset and framework for open domain spontaneous speech question answering from long audio files. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Parthasaarathy Sudarsanam and Tuomas Virtanen. 2023. Attention-based methods for audio question answering. In *31st European Signal Processing Conference (EUSIPCO)*. EUSIPCO.

Jinchuan Tian, William Chen, Yifan Peng, Jiatong Shi, Siddhant Arora, Shikhar Bharadwaj, Takashi Maekaku, Yusuke Shinohara, Keita Goto, Xiang Yue, Huck Yang, and Shinji Watanabe. 2025. OpusLM: A Family of Open Unified Speech Language Models. In *Interspeech 2025*.

Prakhar Verma, Sukruta Prakash Midigeshi, Gaurav Sinha, Arno Solin, Nagarajan Natarajan, and Amit Sharma. 2025. Plan*RAG: Efficient test-time planning for retrieval augmented generation. In *Workshop on Reasoning and Planning for Large Language Models*.

Changhan Wang, Morgane Riviere, Ann Lee, Anne Wu, Chaitanya Talnikar, Daniel Haziza, Mary Williamson, Juan Pino, and Emmanuel Dupoux. 2021. VoxPopuli: A large-scale multilingual speech corpus for representation learning, semi-supervised learning and interpretation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics.

Wenyi Yu, Siyin Wang, Xiaoyu Yang, Xianzhao Chen, Xiaohai Tian, Jun Zhang, Guangzhi Sun, Lu Lu, Yuxuan Wang, and Chao Zhang. 2025. Salmonn-omni: A standalone speech llm without codec injection for full-duplex conversation. *Preprint*, arXiv:2505.17060.

Zihan Zhao, Yiyang Jiang, Heyang Liu, Yu Wang, and Yanfeng Wang. 2024. Librisqa: A novel dataset and framework for spoken question answering with large language models. *IEEE Transactions on Artificial Intelligence*.

Ge Zhu, Jordan Darefsky, and Zhiyao Duan. 2024. Cacophony: An improved contrastive audio-text model. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 32.

A User Queries

Example: Query template for QA

```
Given the context, answer the following question in a short sentence:
<question here>
```

Example: Query template for Summary

```
## Task
Please provide an abstractive summary of this meeting.

You should work on summarization starting from <start> sec to <end> sec.
Produce a concise, factual summary covering goals, key decisions, concerns,
and next steps.

Stay within 5-7 sentences.
## Answer
```

Example: Query template for Speaker Diarization

~~~~
## Task
Perform speaker diarization for the provided audio segment spanning
<start> to <end> seconds in the original recording.

## Requirements
- Generate the following json
```
[
  {"start": "start_time", "end": "end_time", "speaker": "speaker1"},
  {"start": "start_time", "end": "end_time", "speaker": "speaker2"},
    ...
]
```

## Answer
~~~~

Example: Query template for Emotion Recognition

```
You are an emotion recognition model. Analyze the audio between <start> and <end>
seconds and respond with the emotion in the format
{"labels": ["Happy", "Angry"]}.
Return the most likely label(s).
```

Example: Query template for Sound Event Detection

```
You are a sound event localization (SED) model. Detect occurrences of the
following sound event label(s): <event label> in the audio clip.
Return JSON in the format
{"events": [{"label": "<event label>", "start": 0.0, "end": 0.0}]}.
Fewer events are preferred.
```

Example: Query template for Speaker Count task

````
## Task
You should count the number of speakers starting from <start> sec to <end>
sec.

## Requirements
- Generate the following json including the number of speakers in integer.

```
{"answer": <integer>}
```

## Answer
````

Example: Query template for Event Ordering task

```
You are given the following sound event labels:
(1) <label 1>
(2) <label 2>
(3) <label 3>
Determine the correct chronological order of these events based on their first
occurrence in the audio clip (<start> to <end> seconds). Return JSON in the
format {"order": [1, 2, 3]}.
```

Example: Query template for Speaker Constrained MCQA task

```
## Task
You should work on the utterance from speaker <speaker>.
If you cannot answer the question from the given speaker, just reply
"This question is not answerable."

## Question
Given the context, answer the following question in a short sentence:
<question here>

## Answer
```

B Detailed experimental results for base tasks

B.1 Top line results

Table 7: Topline results for abstractive QA (QA-1) and multiple-choice QA (MCQA)

	QA-1				MCQA
Duration(min)	BLEU	Rouge1	Rouge2	RougeL	acc
10	29.03	56.07	38.67	51.39	79.4
30	30.39	56.80	39.50	52.02	77.94
60	31.27	56.78	40.35	52.19	78.9
300	31.15	58.16	40.6	53.31	77.06
540	30.54	56.38	39.4	51.78	75.56

Table 8: Topline results for summarization

	Summ			
Duration(min)	BLEU	Rouge1	Rouge2	RougeL
10	9.21	44.82	14.74	24.56
30	8.17	42.66	12.85	23.08
60	7.82	42.26	12.61	22.85
300	6.24	36.99	9.63	20.04
540	4.61	34.82	7.56	18.58

Table 9: Topline results for speaker diarization, emotion recognition, and sound event detection

	SD		ER		SED
Duration(min)	DER	JER	Macro-f1	Micro-f1	acc
10	10.55	23.31	26.48	27.54	50.89
30	12.64	24.09	19.32	22.50	48.48
60	12.42	22.76	14.56	17.27	49.03
300	12.42	22.51	18.73	19.26	50.38
540	12.58	22.59	20.13	25.41	51.46

B.2 Owsm + Qwen model

Table 10: Results for abstractive QA (QA-1) and multiple-choice QA (MCQA)

		QA-1				MCQA	
PlanRAG	Duration(min)	BLEU	Rouge1	Rouge2	RougeL	acc (parseable)	acc (end-to-end)
	10	18.09	42.41	26.47	37.66	66.24	61.80
	30	6.48	20.18	8.26	18.15	53.13	31.34
	60	6.57	20.10	8.55	18.28	48.60	20.73
	300	7.44	22.15	9.77	19.90	30.69	8.33
	540	6.28	20.23	8.14	18.15	-	-
✓	10	18.35	40.02	25.40	36.51	65.67	50.05
✓	30	17.05	38.34	23.64	34.63	67.23	51.63
✓	60	17.12	38.29	24.10	34.64	65.09	52.29
✓	300	17.09	38.30	23.76	34.52	63.87	50.95
✓	540	12.73	30.24	16.93	26.91	56.70	41.04

Table 11: Results for summarization (Summ), speaker diarization (SD), emotion recognition (ER), and sound event detection (SED) with OWSM + Qwen model.

		Summ				SD		ER		SED	
PlanRAG	Duration(min)	BLEU	Rouge1	Rouge2	RougeL	DER	JER	Macro-f1	Micro-f1	F1 (1 sec)	F1 (5sec)
	10	9.10	43.52	13.79	24.48	85.91	82.08	15.40	19.13	0	0
	30	5.22	27.11	6.17	16.37	95.09	94.01	10.81	12.55	0	0
	60	2.33	18.55	2.96	11.83	98.61	97.68	7.29	8.25	0	0
	300	1.37	15.09	2.32	10.07	99.70	99.58	0.91	0.95	0	0
	540	-	-	-	-	-	-	-	-	-	-
✓	10	8.43	40.02	11.55	22.13	16.16	25.24	18.96	21.83	14.08	73.94
✓	30	6.33	37.27	9.55	20.46	16.93	23.66	22.48	24.85	16.01	55.27
✓	60	5.97	35.46	8.81	19.82	22.35	27.72	20.56	23.51	16.01	49.76
✓	300	5.18	33.29	7.43	18.72	24.44	28.49	18.63	22.65	11.44	41.04
✓	540	4.25	33.49	6.96	18.42	26.34	30.03	20.94	25.55	10.08	35.94

B.3 Gemini results

Table 12: Gemini results for abstractive QA (QA-1) and multiple-choice QA (MCQA)

		QA-1				MCQA	
PlanRAG	Duration(min)	BLEU	Rouge1	Rouge2	RougeL	acc (parseable)	acc (end-to-end)
	10	25.92	52.71	36.47	48.59	74.45	74.30
	30	25.74	52.11	35.97	48.09	70.84	70.56
	60	23.76	48.70	33.66	45.32	59.00	58.83
	300	18.90	43.10	27.16	39.62	35.29	35.29
	540	-	-	-	-	-	-
✓	10	16.47	36.63	23.39	34.12	70.00	45.24
✓	30	15.65	35.40	22.14	32.69	69.53	47.21
✓	60	15.18	36.07	21.38	33.70	65.00	41.60
✓	300	11.48	30.12	16.09	26.89	67.62	42.40
✓	540	20.42	41.24	27.99	37.61	65.13	43.90

Table 13: Gemini results for summarization (Summ), speaker diarization (SD), emotion recognition (ER), and sound event detection (SED)

		Summ				SD		ER		SED	
PlanRAG	Duration(min)	BLEU	Rouge1	Rouge2	RougeL	DER	JER	Macro-f1	Micro-f1	F1 (1 sec)	F1 (5sec)
	10	0.00	8.14	2.55	5.19	79.47	86.18	3.27	13.09	3.00	3.28
	30	0.00	7.72	1.79	4.73	86.43	90.15	3.25	12.97	0.23	0.24
	60	0.00	7.52	1.37	4.61	92.68	92.76	4.01	11.56	0.14	0.16
	300	0.00	6.98	1.24	4.31	100.00	91.77	3.85	12.34	0.02	0.03
	540	-	-	-	-	-	-	-	-	-	-
✓	10	6.27	34.42	9.37	20.59	19.69	29.69	20.89	22.74	21.77	74.58
✓	30	4.10	30.16	6.87	17.41	26.23	33.81	27.82	28.69	19.07	72.17
✓	60	7.84	38.38	11.06	22.18	35.35	38.63	19.62	20.90	13.53	73.96
✓	300	4.53	30.98	6.56	18.71	29.23	34.63	19.83	25.52	7.86	60.01
✓	540	3.57	28.39	5.65	16.78	29.54	34.77	25.88	28.44	13.40	49.82

B.4 Voxtral results

Table 14: Voxtral results for abstractive QA (QA-1) and multiple-choice QA (MCQA)

	QA-1				MCQA	
Duration(min)	BLEU	Rouge1	Rouge2	RougeL	acc (parseable)	acc (end-to-end)
10	22.34	47.85	31.59	43.90	73.17	73.10
30	22.32	47.08	30.79	43.44	68.84	68.36

Table 15: Voxtral results for summarization (Summ), speaker diarization (SD), emotion recognition (ER), and sound event detection (SED)

	Summ				SD		ER		SED	
Duration(min)	BLEU	Rouge1	Rouge2	RougeL	DER	JER	Macro-f1	Micro-f1	F1 (1 sec)	F1 (5 sec)
10	12.14	46.86	16.53	28.12	100.00	100.00	5.70	14.80	0.00	0.00
30	9.52	41.69	13.42	24.97	100.00	100.00	3.97	12.44	0.00	0.00

C Survey on Interspeech Paper

We analyze Interspeech 2020–2025 main conference topics to justify that our evaluation tasks (speaker, emotion, event, and long-form speech understanding) reflect dominant research directions in the speech community. Table 16 summarizes the resulting topic distribution, highlighting the prevalence of speech recognition, speaker-related tasks, and paralinguistic analysis. This analysis informed our choice of evaluation domains in the main experiments.

Table 16: Overview of research topics and their frequency in Interspeech 2020–2025 sessions.

Topics	**count**
Speech Recognition	3556
Speaker and Language Identification	333
Speech Recognition: Architecture, Search & Linguistic Components	288
Speech Perception, Production and Acquisition	273
Spoken Language Processing: Translation, Retrieval and Resources	264
Phonetics, Phonology and Prosody	244
Spoken Dialog Systems and Conversational Analysis	225
Paralinguistics and Affective Computing	148
Speech, Voice and Hearing Disorders	116
Speech Coding and Enhancement	349
Speech Synthesis and Spoken Language Generation	674
Analysis of Speech and Audio Signals	363

D Temporal Fusion Details

For a given base stream segment, we consider candidate segments from a target stream that temporally overlap with the base segment within a fixed tolerance window $\pm\tau$ seconds. Among these candidates, we select at most one segment whose temporal midpoint is closest to that of the base segment. Formally, for each base segment b and target segment t, the midpoint distance is defined as:

$$\left|\frac{\text{b.start} + \text{b.end}}{2} - \frac{\text{t.start} + \text{t.end}}{2}\right|.$$

The target segment with the minimum midpoint distance is selected, and the two segments are grouped into the same retrieval record. If no target segment satisfies the temporal overlap constraint, the base segment is retained without a matched target segment.

Unless otherwise specified, we use $\tau = 2.5$ seconds in all experiments.