PolyAudio: Advancing Multi-Audio Reasoning in Large Audio Language Models with Interleaved Multi-Audio Contexts

Sonal Kumar[1], **Sreyan Ghosh**[1], **Yueqian Lin**[2], **S Sakshi**[1], **Ashish Seth**[1], **Yiran Chen**[2], **Ramani Duraiswami**[1], **Dinesh Manocha**[1]
[1]University of Maryland, College Park, USA [2]Duke University, USA
Correspondence: sonalkum@umd.edu

Abstract

Large Audio Language Models have demonstrated impressive performance on single-clip audio understanding tasks, including automatic speech recognition, captioning, sound event recognition, etc. However, their ability to reason over *interleaved audio-language contexts*—where answering a query requires relating information across *multiple audio clips*—remains limited. We present **PolyAudio**, an LALM built on Audio Flamingo 3 that targets multi-audio understanding via post-training *instruction tuning*. To train PolyAudio, we also propose **PolyAudio-Instruct**, a high-quality instruction-tuning dataset consisting of 1.3M+ QA pairs, spanning over **14 diverse tasks** to learn multi-audio understanding and reasoning. PolyAudio uses an explicit interleaved representation with clip indexing to encourage faithful grounding and reduce ambiguity in multi-clip references. We evaluate PolyAudio on a diverse suite of multi-audio benchmarks alongside standard single-audio tasks. PolyAudio achieves strong performance on multi-audio reasoning, outperforming competitive baselines that are also often limited to reasoning over up to 2 audio clips, while preserving robust single-clip performance. Overall, our results suggest that precise, academic-scale multi-audio instruction tuning can unlock advanced multi- and cross-clip audio reasoning capabilities, enabling more capable audio-centric assistants. Project Page: https://github.com/sonalkum/PolyAudio.

1 Introduction

Large Audio Language Models (LALMs) have achieved remarkable performance in *single-clip* understanding. Recent foundation models such as GAMA (Ghosh et al., 2024), Audio Flamingo 3 (Goel et al., 2025), Audio Flamingo 2 (Ghosh et al., 2025), Qwen2-Audio (Chu et al., 2024), LTU (Gong et al., 2024), SALMONN (Tang et al., 2023), MiMo-Audio (Xiaomi, 2025) and

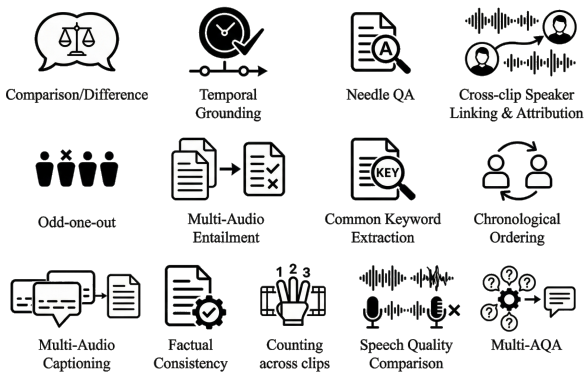

Figure 1: Overview of tasks proposed in PolyAudio.

have demonstrated strong capabilities in automatic speech recognition (ASR), captioning, and sound event detection. However, real-world auditory environments are rarely isolated; they are inherently *multi-contextual*. Users frequently need to compare distinct recordings, track a speaker across separate files, identify acoustic anomalies between two timeframes, or infer a narrative sequence from disjoint clips. In these scenarios, a model must do more than process audio; it must perform *interleaved multi-audio reasoning*—integrating evidence across multiple inputs and resolving complex cross-clip references (e.g., "Is the background noise in the *second* clip louder than the *first*?"). Despite this practical necessity, multi-audio reasoning remains an under-explored frontier. While recent benchmarks like MMAU (Sakshi et al., 2025) and Dynamic-SUPERB (Huang et al., 2024) have begun to test broader audio capabilities, most existing work still treats audio samples in isolation. As observed in the visual domain by recent works like Mantis (Jiang et al., 2024), LLaVA-Interleave (Li et al., 2024), and mPLUG-Owl3 (Ye et al., 2024), pre-training is computationally inefficient and often fails to capture the fine-grained logical dependencies required for high-level reasoning. Furthermore, multi-audio contexts introduce unique challenges

Findings of the Association for Computational Linguistics: ACL 2026, pages 42335–42353
July 2-7, 2026

absent in single-clip tasks: acoustic events may recur with subtle variations, speakers may reappear under different channel conditions, and critical information is often sparse "needle-in-a-haystack" problem spanning long, disjoint contexts (Chen et al., 2024; Ahia et al., 2025).

Main Contributions. In this work, we introduce **PolyAudio**, an LALM trained to reason over interleaved audio-text sequences. PolyAudio is built using our proposed post-training technique of *interleaved multi-audio-text instruction tuning* with **Audio Flamingo 3** (Goel et al., 2025) as the base LALM. To train PolyAudio, we curate **PolyAudio-Instruct**, a high-quality instruction-tuning dataset designed to induce core multi-audio competencies at academic scale. PolyAudio-Instruct comprises **14 distinct task subsets** (Figure 1) that collectively cover the spectrum of multi-auditory reasoning, from low-level acoustic comparison to high-level semantic logic, addressing gaps highlighted in recent logic reasoning studies (Xie et al., 2025; Diao et al., 2025). Overall, our work also demonstrates that multi-audio intelligence can be achieved using only post-training with high-quality data, without the need for large-scale pre-training. Our main contributions are summarized as follows:

- **PolyAudio:** A multi-audio instruction-tuned LALM built on Audio Flamingo 3 (Goel et al., 2025), capable of processing and reasoning over interleaved audio-text sequences without requiring massive multi-audio pre-training.
- **PolyAudio-Instruct:** A curated instruction-tuning mixture of **14 multi-audio tasks** designed to induce robust skills in co-reference, comparison, temporal reasoning, and retrieval (including novel subsets for **Speech Quality Comparison** and **Contrast-Captioning**).
- **PolyAudio Bench:** A comprehensive evaluation benchmark covering all 14 task categories, allowing for rigorous assessment of multi-audio capabilities of Large Audio Language Models.
- **Empirical Results:** We demonstrate that PolyAudio achieves state-of-the-art performance on multi-audio benchmarks, significantly outperforming baselines while maintaining robust performance on standard single-audio tasks.

2 Related Work

2.1 LALMs

Large Audio Language Models (LALMs) are a specific class of LMMs focused on audio understanding and generation. They move beyond the traditional paradigms of speech recognition by using natural language supervision, where models learn from descriptive text rather than predefined labels. This approach allows LALMs to understand and reason about complex audio events, including spoken language, natural sounds, and music, enabling applications such as audio captioning, music composition, and virtual assistants. While some LALMs like Mellow (Deshmukh et al., 2025a) and MALLM (Chen et al., 2024) try to perform multi-audio analysis on up to two audio clips, the majority of LALMs are still restricted to single audio analysis, and research efforts have focused on improving performance on single, contiguous audio streams. Our work addresses this limitation by introducing a model purpose-built for multi-clip reasoning.

2.2 Single and Multi-Audio Benchmarking

Recent work has begun to move beyond single-clip audio understanding, but most efforts remain fundamentally *pairwise*. Mellow (Deshmukh et al., 2025a), ADIFF (Deshmukh et al., 2025b) introduced a framework for explaining differences between two recordings via cross-projection and target comparative reasoning with exactly two audio inputs, making them hard to generalize to $N > 2$ clips. The MAE benchmark and MALLM model (Chen et al., 2024) are framed as stepping into a "multi-audio era," yet their construction and training strategy are centered on synthesized *audio pairs*, and the resulting evaluation primarily reflects pairwise inter-clip context rather than compositional reasoning over multiple discrete clips. In parallel, audio evaluation benchmarks have evolved from representation-centric suites such as SUPERB (Yang et al., 2021) and HEAR (Turian et al., 2022) to instruction-following and reasoning-focused evaluations, including AIR-Bench (which increases difficulty through audio mixing but still evaluates a single mixed stream, $N = 1$) and reasoning benchmarks such as MMAU (Sakshi et al., 2025), and MMSU (Wang et al., 2025a), which probe deeper cognition but largely remain single-clip. MMAU-Pro (Kumar et al., 2025) is among the first to explicitly include multi-audio instances, alongside other challenging tasks such as long-

form and spatial audio; however, multi-audio performance remains low with no model exceeding 30% accuracy, underscoring that generalized multi-clip reasoning is still an open problem. Together, these lines of work indicate a clear gap in existing datasets and models either focus on $N = 1$ reasoning depth or $N = 2$ comparisons, but do not provide a comprehensive approach to evaluate and enable compositional reasoning across multiple audios. MMAR (Ma et al., 2025) also contains a small subset of multi-audio QA, but resorts to concatenating the audios for the benchmark and is not explicitly a mentioned category.

3 Interleaved Multi-Audio-Text

Decomposing Multi-Audio Intelligence. Following the works of recent multi-image studies (Jiang et al., 2024; Zhao et al., 2024), we argue that multi-audio understanding is not merely processing "many audios," but rather mastering specific *relational skills*. We categorize these into four core competencies for our dataset:

1. **Comparison & Difference:** The ability to capture nuances between clips, such as identifying the *Odd-one-out* or performing *Speech Quality Comparison* across multiple samples.

2. **Co-reference & Linking:** The ability to ground natural language references (e.g., "the second speaker") to specific audio segments, essential for tasks like *Cross-clip Speaker Linking* and *Common Keyword Extraction*.

3. **Reasoning & Logic:** Synthesizing information to verify claims, grounded in *Audio Entailment*, *Factual Consistency*, and retrieving specific details via *NeedleQA*, extending concepts from single-audio reasoning (Ghosh et al., 2023; Xie et al., 2025).

4. **Temporal Understanding:** Observing sequences to understand progression, enabling *Chronological Ordering*, *Temporal Grounding*, and *Counting Across Clips*.

PolyAudio-Instruct & Interleaved Training. To systematically teach these behaviors, PolyAudio-Instruct balances these skills with generative tasks like *Multi-Audio Captioning* and **Contrast-Captioning**. We also introduce **Multi-AQA**, a question-answering subset where the answer cannot be derived from a single clip alone, forcing the model to aggregate information. Technically, PolyAudio represents multi-audio inputs using an explicit interleaving format (e.g., `User: <sound-1> vs <sound-2>...`), utilizing the Audio Flamingo 3 (Goel et al., 2025) backbone to process distinct audio tokens. This design mirrors natural user interaction and prevents the model from collapsing multi-clip problems into single-clip heuristics.

3.1 Method

Model Architecture: Most existing LALMs are optimized for single-clip audio understanding and do not natively support *interleaved multi-audio* inputs, either due to architectural constraints or missing training/inference codepaths for handling multiple clips as a single context. Following the design philosophy used to extend single-image LMMs to interleaved multi-image settings, we build **PolyAudio** on top of **Audio Flamingo 3** and add explicit support for multi-clip training and inference. Concretely, PolyAudio consists of (i) an audio encoder that converts each waveform into frame-level representations, (ii) a lightweight resampling/projection interface that compresses each clip into a fixed number of *audio tokens* compatible with the LLM embedding space, and (iii) a decoder-only LLM that performs instruction following and generation conditioned on the interleaved sequence of audio-token blocks and text. This design keeps the model *clip-separable* at the representation level (each clip becomes its own token block) while allowing *joint reasoning* via the LLM's attention over the full interleaved context.

Context Lengths: How many audio clips can PolyAudio accept? As in multi-modal LMMs, this is constrained by (a) the LLM backbone's maximum context length and (b) the number of audio tokens produced per clip. Let L denote the LLM's maximum token budget, T the number of text tokens in the prompt (including instructions and conversation history), and K the effective number of audio tokens contributed by each clip after resampling/projection. Then the maximum number of clips that can be accommodated is approximately:

$$N_{\max} \approx \left\lfloor \frac{L - T}{K} \right\rfloor. \tag{1}$$

For instance, with Qwen-2.5-7B (Yang et al., 2025) as the LLM (context length $L = 131,072$) and the Whisper Large v3 encoder, which outputs 1500 feature frames (1280-dim) per 30-second clip, the

Task Name	Description	Example Prompt (Interleaved Text–Audio)
Comparison/Difference	Compares and contrasts two or more clips.	Compare (audio 1: [BOA]<sound-1>[EOA]) with (audio 3: [BOA]<sound-3>[EOA]). What is the most salient difference between them?
Temporal Grounding	Localizes a described event to the correct clip(s) and time.	Here are three clips: (audio 1: [BOA]<sound-1>[EOA]) (audio 2: [BOA]<sound-2>[EOA]) (audio 3: [BOA]<sound-3>[EOA]). In which clip does a car driving by occur, and what is the approximate timestamp range?
Needle QA	Finds sparse evidence across clips and answers a specific query.	You are given four clips: (audio 1: [BOA]<sound-1>[EOA]) (audio 2: [BOA]<sound-2>[EOA]) (audio 3: [BOA]<sound-3>[EOA]) (audio 4: [BOA]<sound-4>[EOA]). One clip contains a mention of the number of birds. Which clip mentions it, and what number is stated?
Cross-clip Speaker Linking & Attribution	Links a speaker across non-contiguous clips and attributes speech correctly.	Consider (audio 1: [BOA]<sound-1>[EOA]) and (audio 3: [BOA]<sound-3>[EOA]). Do they contain speech from the same person? If yes, state so and briefly justify using vocal/linguistic cues.
Odd-one-out	Identifies the semantically/thematically distinct clip.	Which clip does not belong? (audio 1: [BOA]<sound-1>[EOA]) (audio 2: [BOA]<sound-2>[EOA]) (audio 3: [BOA]<sound-3>[EOA]). Answer with {audio i} and a short reason.
Multi-Audio Entailment	Determines support/contradiction/insufficient evidence across clips.	Given (audio 1: [BOA]<sound-1>[EOA]) and (audio 2: [BOA]<sound-2>[EOA]), evaluate the claim: "The weather is sunny today." Does (audio 3: [BOA]<sound-3>[EOA]) support, contradict, or not mention this claim?
Common Keyword Extraction	Extracts a shared keyword/phrase across clips.	Find one word/short phrase that appears in all clips: (audio 1: [BOA]<sound-1>[EOA]) (audio 2: [BOA]<sound-2>[EOA]) (audio 3: [BOA]<sound-3>[EOA]).
Chronological Ordering	Arranges clips into the correct temporal sequence.	The following clips are shuffled: (audio 1: [BOA]<sound-1>[EOA]) (audio 2: [BOA]<sound-2>[EOA]) (audio 3: [BOA]<sound-3>[EOA]) (audio 4: [BOA]<sound-4>[EOA]). Output the chronological order as indices (e.g., 2→4→1→3).
Multi-audio Captioning (single and aggregated)	Generates per-clip captions and/or an aggregated summary.	Describe each clip in one sentence: (audio 1: [BOA]<sound-1>[EOA]) (audio 2: [BOA]<sound-2>[EOA]) (audio 3: [BOA]<sound-3>[EOA]). Then give a two-sentence overall summary across all clips.
Factual Consistency	Checks factual agreement across clips and identifies inconsistencies.	Across (audio 1: [BOA]<sound-1>[EOA]) (audio 2: [BOA]<sound-2>[EOA]) (audio 3: [BOA]<sound-3>[EOA]), the speaker states a person's age but the numbers disagree. What is the correct age (if determinable), and which clip(s) support it?
Counting across clips	Counts occurrences of an event/object across all clips.	Count total dog barks across the set: (audio 1: [BOA]<sound-1>[EOA]) (audio 2: [BOA]<sound-2>[EOA]) (audio 3: [BOA]<sound-3>[EOA]) (audio 4: [BOA]<sound-4>[EOA]).
Speech Quality Comparison	Selects best/worst speech quality (clarity/noise/echo/distortion/bandwidth).	Which clip has the clearest speech for transcription: (audio 1: [BOA]<sound-1>[EOA]), (audio 2: [BOA]<sound-2>[EOA]), or (audio 3: [BOA]<sound-3>[EOA])? Answer with {audio i} and briefly name the dominant impairment in the others.
Contrast-Captioning	Produces a discriminative caption to identify a target clip among distractors.	Target: (audio 2: [BOA]<sound-2>[EOA]). Distractors: (audio 1: [BOA]<sound-1>[EOA]) and (audio 3: [BOA]<sound-3>[EOA]). Write a caption that uniquely identifies the target clip by its distinctive sounds/speaker traits/events.
Multi-AQA	Requires synthesis across multiple clips (not solvable from any single clip).	Across the clips (audio 1: [BOA]<sound-1>[EOA]) (audio 2: [BOA]<sound-2>[EOA]) (audio 3: [BOA]<sound-3>[EOA]) (audio 4: [BOA]<sound-4>[EOA]), do they maintain the same tempo?

Table 1: Multi-audio analysis and reasoning tasks in **PolyAudio-Instruct**. Example prompts use an interleaved text–audio format with explicit clip indices and boundary tokens [BOA] / [EOA].

theoretical capacity is substantial. Even with a conservative projection where $K \approx 1500$, the model could accommodate over 80 clips. However, in our setting, we restrict focus to *compositional reasoning across up to five clips* ($N \leq 5$) not due to context limits, but to ensure robust cross-clip integration and avoid the performance degradation often observed in extremely long interleaved sequences.

Interleaved Text-Audio: A proper text–audio interleaving format is critical for learning multi-clip grounding and reasoning. We adopt the same two desiderata highlighted in interleaved multi-image work—*(i) clearly marking clip boundaries* and *(ii) explicitly denoting clip indices* and instantiate them for audio as:

(audio {i}: <BOA><audio_tokens><EOA>)

where audio {i} provides the serialized clip index, <BOA> and <EOA> delimit the beginning/end of the i-th clip's audio-token block, and <audio_tokens> is the compressed representation produced by the Audio Flamingo 3 interface. In practice, we implement the delimiters as dedicated special tokens (e.g., [BOA] and [EOA]) and also expose clip identity through explicit clip markers (e.g., <sound-1>, <sound-2>, ...) so that natural language references such as "in the third clip" can be grounded unambiguously to the intended audio segment.

3.2 PolyAudio-Instruct: A large scale multi-audio QA dataset

We construct **PolyAudio-Instruct**, a large scale multi-audio text interleaved instruction tuning dataset with 1.3M single-turn QA pairs from publicly available open source datasets. Detailed statistics about the dataset are shown in Section 2.

Task Category	# Samples
Comparison & Difference	393,990
Reasoning & Logic	328,323
Co-reference & Linking	328,323
Temporal Understanding	262,659
Total	**1,313,295**

Table 2: Distribution of instruction-tuning samples across the four core categories in PolyAudio-Instruct.

PolyAudio-Instruct is collected based on the 4 multi-audio skills described below. We further divide these into 14 sub-tasks shown in Table 1.

Comparison & Difference. Many multi-audio questions are inherently comparative. A user may ask what changed between two recordings, which clip sounds most similar to a reference, or which sample is the cleanest for transcription. This skill requires the model to listen to each clip, form a stable representation of its salient attributes, and then contrast those attributes across clips. Deshmukh et al. (2025b) focuses specifically on this task, where the model is asked to explain the difference between a pair of audios. PolyAudio-Instruct captures this through tasks such as *Comparison/Difference*, *Odd-one-out*, and *Speech Quality Comparison*. These tasks encourage fine-grained judgments that go beyond keyword matching, including differences in acoustic events, speaking style, background conditions, and degradations such as noise, clipping, or bandwidth limitation. Training on these tasks teaches the model to explicitly identify what is shared and what is unique across clips, and to articulate the basis of its decision when the answer is not a single label.

Co-reference & Linking. Multi-audio instructions often contain references that only make sense in a multi-clip context, such as "in the third clip", "the second speaker", or "the clip that mentions ...". Solving these problems requires grounding language to the correct clip index and, in many cases, linking entities that reappear across clips. In PolyAudio-Instruct, this skill is enabled by *Cross-clip Speaker Linking & Attribution* and by content-level linking tasks such as *Common Keyword Extraction*. These questions push the model to recognize whether the same speaker appears in non-contiguous segments despite changes in channel, topic, or speaking rate and to align repeated lexical items or short phrases across clips, which implicitly tests multi-clip ASR robustness and reference resolution. Across these subsets, the model learns to keep a consistent notion of "who" and "what" each clip refers to, which is essential for downstream multi-audio dialogue and long-context interactions.

Reasoning & Logic. Beyond aligning references, many questions require the model to combine evidence from multiple clips and apply basic logical structure. This includes verifying whether a claim is supported, contradicted, or not mentioned, resolving conflicting statements, and retrieving a sparse detail that is only present in one clip. PolyAudio-Instruct instantiates this skill via *Multi-Audio Entailment*, *Factual Consistency*, and *NeedleQA*. These subsets force the model to reason with information that is distributed across clips, rather than relying on a single dominant segment. They also promote careful evidence tracking, since the model must cite which clip provides support, which clip introduces a contradiction, or where the key fact occurs. In this way, PolyAudio-Instruct extends single-audio reasoning settings to multi-audio contexts, where compositionality and conflict resolution are important (Ghosh et al., 2023).

Temporal Understanding. Multi-audio queries frequently involve time and ordering. Examples include reconstructing the correct sequence of events from shuffled clips, finding where a particular event occurs, or counting repeated events across clips. This skill is captured in PolyAudio-Instruct through *Chronological Ordering*, *Temporal Grounding*, and *Counting Across Clips*. *Chronological Ordering* requires the model to infer progression using cues such as discourse structure, event causality, and narrative flow. *Temporal Grounding* requires linking a natural language event description to the relevant clip and, when timestamps are available, to an approximate interval within that clip. *Counting Across Clips* requires tracking occurrences across the entire context and aggregating them reliably. Together, these tasks encourage the model to build a coherent temporal view over multiple discrete recordings, which is necessary for meeting summarization, surveillance-style audio review, and multi-part instructional audio understanding.

3.3 PolyAudio-Bench: A comprehensive multi-audio benchmark

We also construct an evaluation benchmark to test the multi-audio understanding skills induced by PolyAudio-Instruct. We meticulously curate a test set of 2000 QA pairs to test the interleaved multi-audio text understanding of LALMs consisting of

the skills from PolyAudio-Instruct.

3.4 Evaluation

We evaluate our model outputs with an automated judge, adopting the LLM-as-a-judge paradigm popularized in recent text and multimodal evaluation. Specifically, we use the open-source **Qwen3-7B-Instruct** as our judge for reproducibility and cost efficiency. For each item, the judge receives (i) the question, (ii) the ground truth answer and (iii) the model's answer. A rubric-guided prompt instructs the judge to produce a *scalar score* in $[1, 5]$ based on *factual correctness with respect to the provided clips*. While evaluating LLM-as-a-judge, we average over two paraphrased judge prompts. Inspired by prior LLM-as-a-judge protocols, we report only the judge's scalar score as our main metric across tasks. To validate this setting, we obtain human scores on 100 items from our PolyAudio-Bench scored under the same rubric and report the correlation between human and judge scores (Spearman's ρ and Kendall's τ) in Table 3.

LLM	Spearman's ρ	Kendall's τ
Qwen-3 7B Instruct	**0.748**	**0.692**
Llama 3.1 8B	0.675	0.612

Table 3: Correlation between human and LLM-as-a-Judge for evaluating model responses for answer correctness on PolyAudio-Bench.

3.5 Data Curation

Figure 2 illustrates the three-stage data curation pipeline used to construct **PolyAudio-Instruct** and **PolyAudio-Bench**.

Stage 1: Raw Data Aggregation. We first aggregate diverse single-audio samples from two primary streams: (i) **Open-Source Corpora**: We source high-quality recordings from established datasets including AudioSet (Gemmeke et al., 2017), Clotho (Drossos et al., 2020), and AudioCaps (Kim et al., 2019) for sound events; LibriSpeech (Panayotov et al., 2015), CommonVoice (Ardila et al., 2020), and DiPCo (Van Segbroeck et al., 2019) for speech; and MusicCaps (Agostinelli et al., 2023), Music4All (Santana et al., 2020), QualiSpeech (Wang et al., 2025b) and PicoAudio (Xie et al., 2024) for musical and synthetic audio. (ii) **Synthetic Generation**: To enhance diversity, we generate synthetic speech

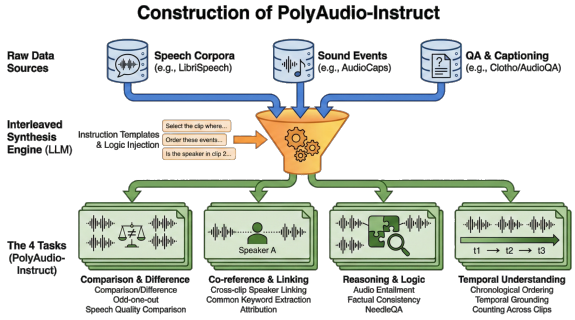

Figure 2: We synthesize interleaved multi-audio instances from raw single-audio sources using varied prompts. The resulting dataset is organized into four tasks representing distinct cognitive skills: Comparison & Difference, Co-reference & Linking, Reasoning & Logic, and Temporal Understanding.

samples by prompting GPT-5 to produce rich transcripts, which are then rendered into high-fidelity audio using Higgs Audio v2 (Boson AI, 2025).

Stage 2: Interleaved Synthesis Engine. These raw inputs (metadata) are then fed into our **Interleaved Synthesis Engine**. We utilize GPT-OSS-120B (OpenAI, 2025)) to act as logic injectors and generate QA pairs. The engine synthesizes complex, multi-clip contexts from the isolated metadata. This process transforms single-audio labels into intricate, interleaved question-answer pairs that require cross-clip reasoning. The specific prompts used to induce behaviors for each task bucket are detailed in Appendix C. Finally, the synthesized instances are classified into four core competency pillars: *Comparison & Difference*, *Co-reference & Linking*, *Reasoning & Logic*, and *Temporal Understanding*.

4 Experiments

4.1 Training Details

For training PolyAudio, we use 8 A6000 GPUs, with a learning rate of 1e-5 and an effective batch size of 128. We set the warmup ratio to be 0.05 and use a cosine learning rate scheduler. We adopt LoRA-based training (Hu et al., 2022)–similar to LTU and GAMA–by freezing the model's original weights and training LoRA adapters for the LLM. This approach allows end-users to flexibly enhance the model's reasoning and multi-audio understanding capabilities on demand.

4.2 Evaluation Benchmarks

Apart from evaluating on PolyAudio-Bench, we also evaluate on the following benchmarks for the multi-audio analysis and reasoning tasks:

Model	PolyAudio-Bench Tasks														Aggregates	
	Comp	Odd	Qual	Link	Key	Ent	Fact	NQA	Ord	Temp	Cnt	Cap	CCap	AQA	Avg	MMAU-Pro
Qwen2-Audio	48.2	44.5	35.0	42.1	50.4	46.5	48.0	40.2	42.6	38.1	40.5	58.2	52.1	48.0	45.1	49.5
ADIFF	58.4	45.2	50.1	25.3	20.1	22.5	20.4	15.6	18.2	15.0	15.1	40.2	45.1	35.0	30.4	27.4
Mellow	45.1	35.4	38.2	15.6	15.2	18.5	18.1	12.4	12.1	12.5	12.0	25.3	28.1	25.0	22.3	20.8
AF-3	32.5	30.1	28.4	32.0	38.5	34.2	34.1	28.5	30.2	30.0	30.5	42.1	40.2	38.5	32.9	26.0
Gemini 2.5 Pro	38.5	35.2	25.1	35.4	55.2	52.1	50.5	35.2	35.1	25.4	30.2	60.5	45.2	45.1	40.4	52.2
PolyAudio	**75.2**	**72.4**	**78.1**	**74.5**	**76.2**	**70.5**	**71.4**	**68.2**	**72.1**	**70.5**	**72.4**	**78.5**	**76.1**	**75.2**	**73.5**	**60.2**

Table 4: **Detailed Performance Breakdown on PolyAudio-Bench and MMAU-Pro.** We report accuracy (%) across all 14 task subsets of PolyAudio-Bench, along with the macro-average and the performance on the multi-audio subset of MMAU-Pro. Abbreviations: **Comp**: Comparison/Difference, **Odd**: Odd-one-out, **Qual**: Speech Quality, **Link**: Cross-clip Speaker Linking, **Key**: Keyword Extraction, **Ent**: Entailment, **Fact**: Factual Consistency, **NQA**: NeedleQA, **Ord**: Chronological Ordering, **Temp**: Temporal Grounding, **Cnt**: Counting, **Cap**: Multi-Audio Captioning, **CCap**: Contrast Captioning, **AQA**: Multi-AQA.

Models	CLD-1		CLD-2		CLD-3		ACD-1		ACD-2		ACD-3	
	$BLEU_4$	SPICE	$BLEU_4$	SPICE	$BLEU_4$	SPICE	$BLEU_4$	SPICE	$BLEU_4$	SPICE	$BLEU_4$	SPICE
Baseline	8.8	6.4	26.5	26.4	13.7	17.2	13.1	10.1	21.7	21.5	15.1	13.8
QwenAC (L)	9.3	8.2	26.4	21.1	13.5	14.5	14.5	9.5	22.0	24.0	16.1	16.0
QwenAC (F)	7.6	9.5	28.5	**27.3**	12.2	14.9	12.3	9.4	21.7	21.9	14.5	14.3
ADIFF	15.3	11.9	24.5	23.2	17.1	16.7	14.8	12.7	23.4	22.2	16.9	17.1
Mellow	17.3	13.9	26.1	25.0	**17.9**	17.3	15.6	**13.9**	24.3	23.7	17.9	**18.7**
PolyAudio	**17.9**	**15.1**	**28.7**	26.8	17.7	**17.9**	**18.2**	13.2	**26.1**	**24.5**	**18.3**	18.2

Table 5: Comparison of different models on Audio Difference Benchmark across CLD and ACD datasets.

MMAU-Pro: To assess performance on complex multi-audio question answering, we evaluate PolyAudio on the multi-audio subset of MMAU-Pro. This subset contains 456 instances with two or three audio clips, testing a variety of complex reasoning skills. We report the official accuracy metric for this benchmark.

Audio Difference Dataset: Introduced by Deshmukh et al. (2025b), aims to evaluate LALMs on comparative reasoning task over two audio inputs and text. Here each example consists of two audio recordings and a text question, with the goal of identifying the differences between the two audios. The text question specifies the level of detail required in the answer and is categorized into three tiers: Tier 1 (concise), Tier 2 (brief), and Tier 3 (detailed). Beyond assessing comparative reasoning, this task requires the model to integrate audio information with world knowledge to effectively distinguish between the two audios.

4.3 Baselines

We compare PolyAudio against a range of state-of-the-art LALMs and specialized models: **Gemini 2.5 Pro & Audio Flamingo 3:** Leading proprietary and open-weight LALMs, respectively. For these models, which do not natively support multiple discrete audio inputs, we follow the evaluation protocol established by MMAU-Pro and concatenate the audio clips with a 2-second silent separator.

Qwen2-Audio-7B-Instruct: A 7B-parameter audio-language model that takes audio + text inputs and generates text outputs, supporting both voice chat and audio analysis modes

ADIFF: This model supports up to 2 audio inputs natively and is trained mainly for audio difference task. We evaluate ADIFF on our benchmarks. For instances where the number of audios exceeds 2, we join audios with a silence of 2 seconds in consecutive audios and pass it as last audio.

Mellow: Aimed at being an intelligent, small reasoning audio model supporting input of up to 2 audios. We evaluate Mellow in the same fashion as ADIFF.

5 Results

5.1 Result on Multi-Audio Benchmarks

Table 4 summarizes the performance of PolyAudio compared to state-of-the-art baselines across all subsets of tasks and aggregated result on PolyAudio-Bench and multi-audio subset of MMAU-Pro. PolyAudio achieves superior performance across both benchmarks, significantly outperforming the strongest open-source baseline, Qwen2-Audio-7B-Instruct, by a margin of 28.4%

on PolyAudio-Bench and 10.7% on the multi-audio subset of MMAU-Pro. Notably, PolyAudio also surpasses the proprietary Gemini 2.5 Pro on both evaluations, demonstrating that specialized instruction tuning can yield capabilities exceeding those of much larger generalist models. Furthermore, the dramatic improvement over the Audio Flamingo 3 backbone (from 32.9% to 73.5% on PolyAudio-Bench) validates the effectiveness of PolyAudio-Instruct in unlocking multi-audio reasoning, while the lower scores of pairwise-specialized models like ADIFF and Mellow underscore the necessity of handling generalized multi-clip contexts beyond simple $N = 2$ comparisons. We further validate PolyAudio's fine-grained comparative capabilities on the Audio Difference Benchmark (Table 5), a task specifically designed for explaining differences between audio pairs. Despite being a generalist multi-audio model, PolyAudio achieves state-of-the-art results across most metrics, securing the top rank in BLEU_4 on 5 out of 6 subsets.

5.2 Result on Single-Audio Tasks

Our key objective is to enhance multi-audio reasoning capabilities in LALMs, however, we did not want to achieve it at the cost of performance degradation on already established tasks. Table 6 presents a comparison between our model, PolyAudio, and the baseline Audio Flamingo 3 on the MMAU-*test* and MMAR. The results show that our approach is effective; PolyAudio not only preserves but slightly improves upon the baseline's performance on these benchmarks.

Model	Sound	Music	Speech	Average
Audio Flamingo 3	75.83	74.47	66.97	72.42
PolyAudio	76.72	74.93	69.58	73.74

Table 6: Performance comparison of Audio Flamingo 3 and PolyAudio on MMAU-*test*

6 How effective is scaling the number of audios during training?

To understand the impact of context complexity on model performance, we conduct an ablation study by varying the maximum number of interleaved audio clips (N) in the training mixture. As illustrated in Figure 3, we observe a consistent upward trend in reasoning capabilities as we scale from $N = 2$ (64.3%) to $N = 4$ (73.1%), demonstrating that exposure to diverse multi-clip scenarios is essential for learning robust cross-clip relationships. Notably, the performance gains plateau as we approach $N = 5$, with the model achieving a final average score of 73.5%. This saturation suggests that while increasing audio context is beneficial, training with four to five clips is sufficient to unlock generalized multi-audio reasoning skills, with $N = 4$ and $N = 5$ yielding comparable results.

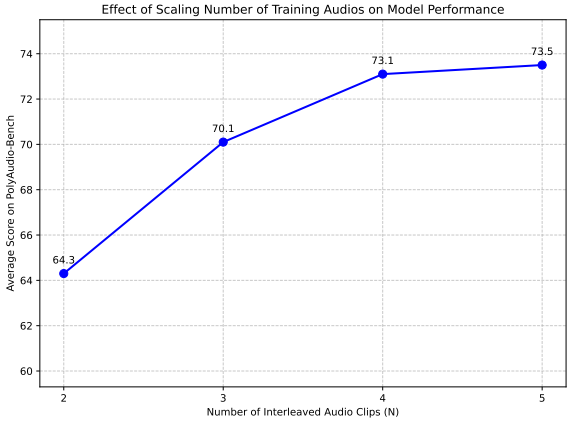

Figure 3: Effect of scaling the number of interleaved training clips (N) on PolyAudio-Bench performance. We observe a steep performance increase up to $N = 4$, after which gains saturate.

7 Conclusion

In this work, we addressed the gap in multi-audio reasoning by introducing **PolyAudio**, a LALM capable of processing and reasoning over interleaved audio-text sequences without relying on massive-scale pre-training. We introduce **PolyAudio-Instruct**, a high-quality dataset comprising 14 distinct task subsets that systematically induce skills in comparison & difference, co-reference & linking, reasoning & logic, and temporal understanding. We observe that PolyAudio sets a new state-of-the-art on complex multi-audio benchmarks, significantly outperforming competitive baselines and proprietary models like Gemini 2.5 Pro, while preserving robust performance on single-clip tasks. Our results demonstrate that targeted, academic-scale instruction tuning is a highly efficient way to solve fine-grained cross-clip reasoning.

8 Limitations and Future Work

While PolyAudio demonstrates strong capabilities in multi-audio reasoning, it has a few limitations. First, PolyAudio-Instruct relies heavily on a synthetic generation pipeline to create interleaved contexts from single-audio metadata (Section 3).

While this ensures diverse and logically complex instructions, it may not perfectly model naturally occurring multi-source environments where acoustic characteristics overlap or shift dynamically between segments in ways that synthetic mixing might miss. Secondly, as PolyAudio is built upon Audio Flamingo 3, it inherits the limitations of the base model, including any domain-specific limitations in its pre-trained audio encoder or single-clip perception capabilities.

In future work, we plan to address these constraints by incorporating efficient attention mechanisms to scale the number of supported clips and integrating naturally recorded multi-source data to complement our synthetic pipeline. Furthermore, we intend to validate the generalizability of our approach by extending the PolyAudio-Instruct training recipe to a broader range of open-source Large Audio Language Models (LALMs), such as Qwen2-Audio and other LALMs, to establish a unified framework for multi-audio intelligence.

9 Acknowledgment

This research is supported in part by Adobe, Amazon, NVIDIA and Sesame.

References

Andrea Agostinelli, Timo I Denk, Zalán Borsos, Jesse Engel, Mauro Verzetti, Antoine Caillon, Qingqing Huang, Aren Jansen, Adam Roberts, Marco Tagliasacchi, and 1 others. 2023. Musiclm: Generating music from text. *arXiv preprint arXiv:2301.11325*.

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.

Rosana Ardila, Megan Branson, Kelly Davis, Michael Kohler, Josh Meyer, Michael Henretty, Reuben Morais, Lindsay Saunders, Francis Tyers, and Gregor Weber. 2020. Common voice: A massively-multilingual speech corpus. In *Proceedings of the twelfth language resources and evaluation conference*, pages 4218–4222.

Boson AI. 2025. Higgs Audio V2: Redefining Expressiveness in Audio Generation. https://github.com/boson-ai/higgs-audio. GitHub repository. Release blog available at https://www.boson.ai/blog/higgs-audio-v2.

Yiming Chen, Xianghu Yue, Xiaoxue Gao, Chen Zhang, Luis Fernando D'Haro, Robby T Tan, and Haizhou Li. 2024. Beyond single-audio: Advancing multi-audio processing in audio large language models. *arXiv preprint arXiv:2409.18680*.

Yunfei Chu, Jin Xu, Qian Yang, Haojie Wei, Xipin Wei, Zhifang Guo, Yichong Leng, Yuanjun Lv, Jinzheng He, Junyang Lin, Chang Zhou, and Jingren Zhou. 2024. Qwen2-audio technical report. *Preprint*, arXiv:2407.10759.

Soham Deshmukh, Satvik Dixit, Rita Singh, and Bhiksha Raj. 2025a. Mellow: a small audio language model for reasoning. *Preprint*, arXiv:2503.08540.

Soham Deshmukh, Shuo Han, Rita Singh, and Bhiksha Raj. 2025b. ADIFF: Explaining audio difference using natural language. In *The Thirteenth International Conference on Learning Representations*.

Xingjian Diao, Chunhui Zhang, Keyi Kong, Weiyi Wu, Chiyu Ma, Zhongyu Ouyang, Peijun Qing, Soroush Vosoughi, and Jiang Gui. 2025. Soundmind: Rl-incentivized logic reasoning for audio-language models. *Preprint*, arXiv:2506.12935.

Konstantinos Drossos, Samuel Lipping, and Tuomas Virtanen. 2020. Clotho: An audio captioning dataset. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 736–740. IEEE.

Jort F Gemmeke, Daniel PW 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)*, pages 776–780. IEEE.

Sreyan Ghosh, Zhifeng Kong, Sonal Kumar, S Sakshi, Jaehyeon Kim, Wei Ping, Rafael Valle, Dinesh Manocha, and Bryan Catanzaro. 2025. Audio flamingo 2: An audio-language model with long-audio understanding and expert reasoning abilities. *Preprint*, arXiv:2503.03983.

Sreyan Ghosh, Sonal Kumar, Ashish Seth, Chandra Kiran Reddy Evuru, Utkarsh Tyagi, S Sakshi, Oriol Nieto, Ramani Duraiswami, and Dinesh Manocha. 2024. GAMA: A large audio-language model with advanced audio understanding and complex reasoning abilities. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 6288–6313, Miami, Florida, USA. Association for Computational Linguistics.

Sreyan Ghosh, Ashish Seth, Sonal Kumar, Utkarsh Tyagi, Chandra Kiran Evuru, S Ramaneswaran, S Sakshi, Oriol Nieto, Ramani Duraiswami, and Dinesh Manocha. 2023. Compa: Addressing the gap in compositional reasoning in audio-language models. *arXiv preprint arXiv:2310.08753*.

Arushi Goel, Sreyan Ghosh, Jaehyeon Kim, Sonal Kumar, Zhifeng Kong, Sang gil Lee, Chao-Han Huck Yang, Ramani Duraiswami, Dinesh Manocha, Rafael Valle, and Bryan Catanzaro. 2025. Audio flamingo 3: Advancing audio intelligence with fully open large audio language models. *Preprint*, arXiv:2507.08128.

Yuan Gong, Hongyin Luo, Alexander H. Liu, Leonid Karlinsky, and James R. Glass. 2024. Listen, think, and understand. In *The Twelfth International Conference on Learning Representations*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

Chien-yu Huang, Wei-Chih Chen, Shu-wen Yang, Andy T Liu, Chen-An Li, Yu-Xiang Lin, Wei-Cheng Tseng, Anuj Diwan, Yi-Jen Shih, Jiatong Shi, and 1 others. 2024. Dynamic-superb phase-2: A collaboratively expanding benchmark for measuring the capabilities of spoken language models with 180 tasks. *arXiv preprint arXiv:2411.05361*.

Dongfu Jiang, Xuan He, Huaye Zeng, Cong Wei, Max Ku, Qian Liu, and Wenhu Chen. 2024. Mantis: Interleaved multi-image instruction tuning. *Transactions on Machine Learning Research*.

Chris Dongjoo Kim, Byeongchang Kim, Hyunmin Lee, and Gunhee Kim. 2019. Audiocaps: Generating captions for audios in the wild. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 119–132.

Sonal Kumar, Šimon Sedláček, Vaibhavi Lokegaonkar, Fernando López, Wenyi Yu, Nishit Anand, Hyeonggon Ryu, Lichang Chen, Maxim Plička, Miroslav Hlaváček, and 1 others. 2025. Mmau-pro: A challenging and comprehensive benchmark for holistic evaluation of audio general intelligence. *arXiv preprint arXiv:2508.13992*.

Feng Li, Renrui Zhang, Hao Zhang, Yuanhan Zhang, Bo Li, Wei Li, Zejun Ma, and Chunyuan Li. 2024. Llava-next-interleave: Tackling multi-image, video, and 3d in large multimodal models. *arXiv preprint arXiv:2407.07895*.

Ziyang Ma, Yinghao Ma, Yanqiao Zhu, Chen Yang, Yi-Wen Chao, Ruiyang Xu, Wenxi Chen, Yuanzhe Chen, Zhuo Chen, Jian Cong, Kai Li, Keliang Li, Siyou Li, Xinfeng Li, Xiquan Li, Zheng Lian, Yuzhe Liang, Minghao Liu, Zhikang Niu, and 15 others. 2025. Mmar: A challenging benchmark for deep reasoning in speech, audio, music, and their mix. *Preprint*, arXiv:2505.13032.

OpenAI. 2025. gpt-oss-120b & gpt-oss-20b model card. *Preprint*, arXiv:2508.10925.

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)*, pages 5206–5210. IEEE.

S Sakshi, Utkarsh Tyagi, Sonal Kumar, Ashish Seth, Ramaneswaran Selvakumar, Oriol Nieto, Ramani Duraiswami, Sreyan Ghosh, and Dinesh Manocha. 2025. MMAU: A massive multi-task audio understanding and reasoning benchmark. In *The Thirteenth International Conference on Learning Representations*.

Igor André Pegoraro Santana, Fabio Pinhelli, Juliano Donini, Leonardo Catharin, Rafael Biazus Mangolin, Valéria Delisandra Feltrim, Marcos Aurélio Domingues, and 1 others. 2020. Music4all: A new music database and its applications. In *2020 International Conference on Systems, Signals and Image Processing (IWSSIP)*, pages 399–404. IEEE.

Changli Tang, Wenyi Yu, Guangzhi Sun, Xianzhao Chen, Tian Tan, Wei Li, Lu Lu, Zejun Ma, and Chao Zhang. 2023. Salmonn: Towards generic hearing abilities for large language models. *arXiv preprint arXiv:2310.13289*.

Joseph Turian, Jordie Shier, Humair Raj Khan, Bhiksha Raj, Björn W Schuller, Christian J Steinmetz, Colin Malloy, George Tzanetakis, Gissel Velarde, Kirk McNally, and 1 others. 2022. Hear: Holistic evaluation of audio representations. In *NeurIPS 2021 Competitions and Demonstrations Track*, pages 125–145. PMLR.

Maarten Van Segbroeck, Ahmed Zaid, Ksenia Kutsenko, Cirenia Huerta, Tinh Nguyen, Xuewen Luo, Björn Hoffmeister, Jan Trmal, Maurizio Omologo, and Roland Maas. 2019. Dipco–dinner party corpus. *arXiv preprint arXiv:1909.13447*.

Dingdong Wang, Jincenzi Wu, Junan Li, Dongchao Yang, Xueyuan Chen, Tianhua Zhang, and Helen Meng. 2025a. Mmsu: A massive multi-task spoken language understanding and reasoning benchmark. *arXiv preprint arXiv:2506.04779*.

Siyin Wang, Wenyi Yu, Xianzhao Chen, Xiaohai Tian, Jun Zhang, Lu Lu, Yu Tsao, Junichi Yamagishi, Yuxuan Wang, and Chao Zhang. 2025b. Qualispeech: A speech quality assessment dataset with natural language reasoning and descriptions. *arXiv preprint arXiv:2503.20290*.

LLM-Core-Team Xiaomi. 2025. Mimo-audio: Audio language models are few-shot learners.

Zeyu Xie, Xuenan Xu, Zhizheng Wu, and Mengyue Wu. 2024. Picoaudio: Enabling precise timestamp and frequency controllability of audio events in text-to-audio generation. *arXiv preprint arXiv:2407.02869*.

Zhifei Xie, Mingbao Lin, Zihang Liu, Pengcheng Wu, Shuicheng Yan, and Chunyan Miao. 2025. Audio-reasoner: Improving reasoning capability

in large audio language models. *arXiv preprint arXiv:2503.02318*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Shu-wen Yang, Po-Han Chi, Yung-Sung Chuang, Cheng-I Jeff Lai, Kushal Lakhotia, Yist Y Lin, Andy T Liu, Jiatong Shi, Xuankai Chang, Guan-Ting Lin, and 1 others. 2021. Superb: Speech processing universal performance benchmark. *arXiv preprint arXiv:2105.01051*.

Jiabo Ye, Haiyang Xu, Haowei Liu, Anwen Hu, Ming Yan, Qi Qian, Ji Zhang, Fei Huang, and Jingren Zhou. 2024. mplug-owl3: Towards long image-sequence understanding in multi-modal large language models. *arXiv preprint arXiv:2408.04840*.

Bingchen Zhao, Yongshuo Zong, Letian Zhang, and Timothy Hospedales. 2024. Benchmarking multi-image understanding in vision and language models: Perception, knowledge, reasoning, and multi-hop reasoning. *arXiv preprint arXiv:2406.12742*.

A Appendix

In the Appendix, we provide:

1. Section B: Ablations
2. Section C: Prompts
3. Section D: Use of AI
4. Section E: Dataset Licenses
5. Section F: Broader Impact & Risks

B Ablations

B.1 Robustness to Clip Ordering

To ensure that PolyAudio relies on genuine semantic reasoning rather than positional heuristics (e.g., biasing towards the most recently processed clip), we evaluate its permutation invariance. We construct a dedicated robustness split comprising 500 randomly selected samples from PolyAudio-Bench and perform inference in two settings: (1) utilizing the original clip order defined in the dataset, and (2) randomly shuffling the input sequence of the audio clips while keeping the text query constant. We observe a negligible performance drop of just **0.4%** in the shuffled setting (from 73.8% to 73.4%). This high consistency confirms that PolyAudio effectively grounds reasoning in the acoustic content and clip identifiers, rather than exploiting spurious positional cues.

B.2 Concatenation vs. Interleaved

A primary intuition of multi-audio evaluation is the potential architectural mismatch between models that natively support discrete inputs and those forced to process concatenated audio streams. To isolate the benefit of our interleaved architecture, we compare PolyAudio's native performance against a variant where input clips are concatenated into a single audio stream separated by silence. As shown in Table 7, the concatenated approach faces significant limitations. While standard LALMs are often restricted to short context windows (e.g., 30 seconds), our backbone, Audio Flamingo 3, supports contexts up to 10 minutes, making concatenation technically feasible. However, despite this capacity, the *PolyAudio (Concatenated)* variant lags behind our native interleaved model by **12.3%**. We hypothesize that while long-context models can process the aggregate duration, they struggle to differentiate distinct "events" within a

single continuous stream without explicit boundary tokens. In contrast, PolyAudio's interleaved format preserves clip identity, enabling the sharp comparative reasoning required for tasks like *Comparison*, *NeedleQA* and others.

Model Configuration	Input Format	Avg. Score
Audio Flamingo 3	Concatenated	32.9
PolyAudio (Ablation)	Concatenated	61.2
PolyAudio (Ours)	**Interleaved**	**73.5**

Table 7: Ablation study comparing explicit interleaved inputs vs. concatenated audio streams. Explicit interleaving significantly outperforms concatenation, likely due to better preservation of clip boundaries and identities.

B.3 End-to-End Reasoning vs. Textual Late Fusion

To determine whether multi-audio reasoning requires direct access to acoustic features or if it can be solved via intermediate textual summaries, we implement a strong "Late Fusion" baseline. In this setup, we use the base Audio Flamingo 3 model to generate detailed captions for each input clip, which are then fed into Qwen3-7B-Instruct alongside the original user query to synthesize an answer.

This cascaded approach achieves a score of only **59.8%**, lagging behind PolyAudio by **13.7%**. We attribute this performance gap to the *information bottleneck* inherent in captioning: text descriptions inevitably discard fine-grained acoustic details—such as subtle background textures, speaker prosody, or precise timbral differences that are essential for comparative tasks like *Speech Quality* and *Odd-one-out*. PolyAudio's "early fusion" approach allows the LLM to reason directly over the raw audio tokens, preserving these critical nuances and enabling more accurate cross-clip deductions.

C Prompts

Below are the prompts used to generate the dataset:

D Use of AI assistants

We leveraged LLMs for three key aspects of our work: grammar and word choice refinement during the writing process, comprehensive literature searches to ensure proper citation of related work, and text data curation, consistent with common practices in LLM-related research.

E Dataset Licenses

The development of PolyAudio relies on several open-source audio corpora and synthetic generation tools. We strictly adhere to the licensing terms of each dataset as follows:

- **AudioSet**: Released under the **Creative Commons Attribution 4.0 International (CC BY 4.0)** license.
- **Clotho**: Audio files are sourced from Freesound under their respective licenses (predominantly **CC BY** and **CC0**), while the captions are distributed under a **Tampere University non-commercial license**.
- **AudioCaps**: Distributed under the **Creative Commons Attribution 4.0 International (CC BY 4.0)** license.
- **LibriSpeech**: Available under the **Creative Commons Attribution 4.0 International (CC BY 4.0)** license.
- **CommonVoice**: Released into the public domain under the **CC0 1.0 Universal** license.
- **DiPCo**: Licensed under the **Community Data License Agreement – Permissive (CDLA-Permissive)**.
- **MusicCaps**: Provided under the **Creative Commons Attribution-ShareAlike 4.0 International (CC BY-SA 4.0)** license.
- **Music4All**: Distributed under the **Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0)** license.
- **PicoAudio**: Released under the **Apache License 2.0**.
- **QualiSpeech**: Distributed under the **Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0)** license.
- **Higgs Audio v2**: The model weights and associated code are licensed under the **Apache License 2.0**.

```
Generate a natural, conversational response to identify which audio clip answers the question.

Question: {question}
Correct answer: The {ordinals[correct_idx]} audio
Emotion/characteristic: {question_data.get('target_emotion', 'described characteristic')}

Generate a response that:
1. Clearly identifies the {ordinals[correct_idx]} audio as the answer
2. Sounds natural and conversational
3. Varies the phrasing (don't always say "demonstrates" or "exhibits")
4. Is 1-2 sentences long

Response:"""
```

Figure 4: Prompt for Diverse Response Generation

```
"You are a careful data author for a factual consistency task with multiple audio clips. "
    "Generate ONE concise factual claim (<= 20 words) and K short scripts (6–15 words each). "
    "Each script should sound natural when read aloud. The claim must be grounded only in the
scripts you produce. "
    "If the target label is 'Yes', all scripts must support the claim. If 'No', make at least one
script contradict or fail to support the claim.\n\n"
    "Output STRICT JSON with this schema:\n"
    "{\n  \"claim\": string,\n  \"label\": \"Yes\" | \"No\",\n  \"clips\": [ { \"support\":
boolean, \"text\": string, \"style\": string } ... K items ],\n  \"rationale\": string (<= 25
words)\n}\n"
)
```

Figure 5: Factual Consistency Across Multiple Audios

F Broader Impact

PolyAudio advances audio intelligence from single-clip processing to complex, multi-context reasoning. This capability enables significant improvements in intelligent media editing, content search, and accessibility tools that require tracking speakers or events across disjoint recordings. By releasing our model, **PolyAudio-Instruct** dataset, and benchmark, we aim to accelerate community research into more capable and reliable audio-centric assistants.

Potential Risk. The ability to perform cross-clip speaker linking raises potential privacy concerns regarding unauthorized tracking or surveillance. Additionally, like many foundation models, PolyAudio may inherit biases from open-source training data (e.g., LibriSpeech, AudioSet), which could result in uneven performance across different accents or demographics. We recommend that downstream applications incorporate appropriate safeguards to mitigate these risks.

```
You craft short, simple, natural English sentences for TTS."

user_prompt = (
    "Task: Write " + str(nclips) + " standalone sentences for synthetic speech.\n"
    + "Constraints:\n"
    + "- Topic: " + topic + "\n"
    + "- Shared keywords to include in EVERY sentence (exact words): " + ", ".join(common) + "\n"
    + "- Optionally include one extra topical word from this set: " + ", ".join(topic_keywords) +
"\n"
    + "- 6–14 words per sentence. No numbering, no quotes, one sentence per line.\n"
    + ("- Style hint: " + style_prefix + "\n" if style_prefix else "")
    + "Output: exactly " + str(nclips) + " lines, each a sentence."
)


### Question Templates (randomly selected):

variants = [
    "List the common keywords present in all audio clips (ignore stop words).",
    "What keywords appear across all clips? Exclude stop words.",
    "Identify the shared keywords across the audio clips (no stop words).",
    "Extract the common non-stopword keywords across the clips.",
    "Provide the common keywords across the clips (comma-separated).",
]
```

Figure 6: Common Keyword Extraction Prompt

```
Analyze this sequence of audio clips from a LibriSpeech audiobook and create a temporal ordering question.
Audio sequence:
{context}

Your task:
1. Identify 3 key story developments/events that occur in this sequence
2. List the events in RANDOM order (not chronological order)
3. Create a multiple-choice question asking about the correct chronological order
4. Generate 4 answer options (A, B, C, D) with different permutations
5. Provide a rationale explaining why the correct order makes narrative sense

Return your response as valid JSON with this structure:
{{
    "events": [
        {{"order": 1, "description": "Event description"}},
        {{"order": 2, "description": "Event description"}},
        {{"order": 3, "description": "Event description"}}
    ],
    "question": "Your temporal ordering question here",
    "answer_options": {{
        "A": "1, 2, 3",
        "B": "2, 1, 3",
        "C": "3, 1, 2",
        "D": "1, 3, 2"
    }},
    "correct_answer": "A",
    "rationale": "Your explanation here"
}}"""

### Rationale Generation Prompt:

prompt = f"""Analyze this narrative sequence from a LibriSpeech audiobook and explain why the story developments occur in
a specific chronological order.
Audiobook sequence:
{context}

Story developments to order:
{elements_text}

The correct chronological order is: {correct_order}

Provide a clear, concise explanation (2-3 sentences) of why this is the correct narrative sequence based on the story flow
and logical progression. Focus on how these story elements naturally follow each other in the narrative."""
```

Figure 7: Temporal & Sequential Event Ordering

```
### Instruction Style Variations (randomly selected):
- "Analyze this dialogue and answer the question:"
- "Based on this conversation, please respond to the question:"
- "Examine the following dialogue exchange and address the question:"
- "Consider this conversational sequence and answer:"
- "Review this dialogue interaction and respond to:"

### Causal Question Prompts (5 variations):

f"Explain why the speaker in the {ordinal} turn made this specific response based on the conversational context. Provide a
concise explanation (1-2 sentences)."
f"What motivated the {ordinal} speaker's response? Give a brief analysis of the conversational factors."
f"Analyze the reasoning behind the {ordinal} turn. What conversational elements led to this response?"
f"Why did the {ordinal} speaker choose this particular response? Explain the conversational logic."
f"What drove the {ordinal} speaker to respond this way? Describe the causal factors briefly."

### Coherence Question Prompts (5 variations):

"Describe how these dialogue turns work together to create coherence. Focus on structural patterns and logical connections.
Provide a concise answer (1-2 sentences)."
"How do these conversational elements connect to form a coherent exchange? Explain the structural relationship briefly."
"What makes this dialogue sequence coherent? Identify the key connecting elements in 1-2 sentences."
"Analyze the coherence pattern in this conversation. How do the parts fit together logically?"
"Explain the conversational flow and how each turn contributes to overall coherence."

### Grounding Question Prompts (5 variations):

"Identify what shared understanding or implicit knowledge allows these speakers to communicate effectively. Provide a concise
answer (1-2 sentences)."
"What common ground enables this conversation? Describe the shared context or assumptions briefly."
"What do these speakers mutually understand that isn't explicitly stated? Explain the implicit knowledge."
"Identify the underlying shared context that makes this dialogue possible. What do speakers assume?"
"What implicit understanding connects these speakers? Describe the conversational grounding briefly."

### Pragmatic Question Prompts (5 variations):

"Explain the social purpose or communicative goals of this conversation. Provide a concise answer (1-2 sentences)."
"What social function does this dialogue serve? Describe the interpersonal goals briefly."
"Analyze the communicative intentions in this exchange. What are the speakers trying to accomplish socially?"
"What pragmatic purpose drives this conversation? Explain the social dynamics concisely."
"Identify the social goals and communicative strategies in this dialogue exchange."

causal_questions = [
    "Considering all audio clips, what is the most likely reason the {} says '{}'?",
    "Based on the conversational context, why does the {} respond with '{}'?",
    "Given the dialogue flow, what prompted the {} to say '{}'?",
    "Analyzing the turn sequence, what best explains why the {} says '{}'?",
    "What is the underlying reason for the {}'s response of '{}'?",
    "Considering the complete conversation, why does the {} make the statement '{}'?"
]

coherence_questions = [
    "What is the logical relationship between the first and last audio clips?",
    "How do the middle audio clips connect the first and final statements?",
    "What conversational pattern emerges across all the audio clips?",
    "Which best describes the overall dialogue flow represented in these clips?",
    "What is the primary conversational dynamic at play across these audio segments?",
    "How do these audio clips work together to create a coherent dialogue?"
]

grounding_questions = [
    "What shared context links all the speakers in these audio clips?",
    "What underlying topic or situation connects these dialogue turns?",
    "Based on all clips, what is the speakers' shared understanding about?",
    "What implicit information do the speakers reference across these turns?",
    "What situational context best explains the progression of these audio clips?",
    "What common knowledge or situation do these speakers assume?"
]

pragmatic_questions = [
    "What social dynamic is demonstrated across these conversational turns?",
    "What communicative strategy is being employed in this dialogue sequence?",
    "What interpersonal relationship is reflected in these audio exchanges?",
    "What conversational goal is being pursued across these dialogue turns?",
    "What social function does this dialogue sequence serve?",
    "What pragmatic meaning emerges from this conversational interaction?"
]
```

Figure 8: Multi-Turn Dialogue Coherence & Grounding

```
You generate compact JSON datapoints for Multi-Audio Entailment. Each datapoint has 3-5 short
single-speaker clips, each with a short TTS instructions block, and 5-7 QAs that sometimes require
integrating multiple clips. Use only fictional content. Output strictly valid JSON.

Generate one dataset example.

Rules
        •       Clips: choose K ∈ {3,4,5}. Each clip has exactly one speaker, a short transcript
(~8-20s worth of text), and an instructions string for TTS.
        •       Speaker whitelist (lowercase, exact): alloy, ash, ballad, coral, echo, fable, nova,
onyx, sage, shimmer.
        •       Use one of these per clip (you may reuse names across clips). No other names.
        •       TTS instructions (per clip): provide a single string with exactly these labeled
lines (capitalize labels):
        •       Voice: ...
        •       Tone: ...
        •       Punctuation: ...
        •       Delivery: ...
Keep each line concise. No quotes, emojis, or stage directions. Use \n\n between lines.
        •       Content: across clips, include at least one fact that is supported in one clip and
contradicted in another (day/time, quantity, inclusion/exclusion). Add 1-2 neutral/distractor
details.
        •       QAs: produce 5-7 items. For each QA:
        •       uses: the minimal set of clip_ids needed (≥2 for at least half the QAs).
        •       evidence_quotes: 1-2 short verbatim substrings from the referenced clips.
        •       answer: a natural-language sentence that begins with "Yes, …", "No, …", or "Not
enough information, …", consistent with the union of the uses clips.
        •       Language: English, TTS-friendly (no URLs/emojis; no sound stage directions).
        •       Keep everything concise and consistent. No contradictory QAs.

Return exactly this JSON shape (no extra keys):

{
  "datapoint_id": "string",
  "scenario": "string",
  "clips": [
    {
      "clip_id": "A1",
      "speaker": "alloy|ash|ballad|coral|echo|fable|nova|onyx|sage|shimmer",
      "instructions": "Voice: ...\\n\\nTone: ...\\n\\nPunctuation: ...\\n\\nDelivery: ...",
      "transcript": "Plain text monologue. No timestamps. TTS-friendly."
    }
  ],
  "qa": [
    {
      "qid": "Q1",
      "question": "string",
      "uses": ["A1","A3"],
      "evidence_quotes": [
        {"clip_id": "A1", "quote": "verbatim phrase"},
        {"clip_id": "A3", "quote": "verbatim phrase"}
      ],
      "answer": "Yes, ... / No, ... / Not enough information, ..."
    }
  ]
}

Checks
        •       Every speaker is exactly from the whitelist.
        •       Every quote is a verbatim substring of its referenced transcript.
        •       At least half of QAs require ≥2 clips (as listed in uses).
        •       answer starts with Yes, / No, / Not enough information, and matches the evidence.
```

Figure 9: Audio Entailment Script and QA Generation

```
I am creating a training dataset for an Audio Language Model that must handle questions relating to both the frequency and timing
of events in multiple audio clips.

### Audio Data
You will receive data for up to five audio clips. Each clip provides:
1. **Audio Path**: A string location of the audio file.
2. **Captions**: Short descriptive statements about the sounds or events (e.g., "door slamming three times," "explosion heard
between 3 and 6 seconds," etc.).
3. **Time-Stamped Caption** and **Frequency Caption**:
   - **Time-stamped** details: Where each event starts and ends (e.g., door slamming from 0.45s-1.991s).
   - **Frequency** details: How many times a certain event occurs (e.g., "door slamming three times").

### Task Requirements
1. **Generate Frequency-Based Questions**
   - Example: "How many times does door slamming occur in the first audio?"
   - Another Example: "Which audio has the greatest number of cow moos?"

2. **Generate Time-Stamp-Based Questions**
   - Example: "When does the explosion start and end in audio 3?"
   - Another Example: "At which intervals do we hear door knocking in the second clip?"

3. **Combine or Compare**
   - Feel free to combine frequency and timestamp aspects: "Which audio has the most door slams, and at what times do they occur?"

4. **Provide a Correct Answer**
   - The answer should be a short statement or, if it's a multiple-choice format, indicate the correct choice.
   - It must reference the **actual** times, events, or frequency counts provided in the data (e.g., "It slams three times at
roughly 0.45-1.991s, 3.019-5.8s, and 6.623-8.102s.").

5. **Multiple Question-Answer Pair per Audio Set**
   - Each set of up to five audio clips can yield **multiple** questions and answers.
   - The questions should not repeat.
   - The questions can be **MCQ** or **short-answer** type.

---

## JSON Output Format

Output your question and answer in the following **JSON** structure:

{
  "audio_paths": ["<path1>", "<path2>", ...],
  "frequencyCaption": ["<frequencyCaption1>", "<frequencyCaption2>", ...],
  "question": [ "<Frequency- or Time-Stamp-Based Question 1 >", "<Frequency- or Time-Stamp-Based Question 2>", …],
  "answer": ["<Correct Answer 1 Grounded in the Provided Data>", "<Correct Answer 2 Grounded in the Provided Data>", …]
}

1.      audio_paths: Up to five audio file paths.
2.      frequencyCaption: The descriptive or frequency-based captions for each audio.
3.      questions: A list of the generated direct question.
4.      answer: A list of short or MCQ correct answer that references the time-stamped or frequency data corresponding to each
question.

Example:
{
  "audio_paths": [
    "multi_event_train/syn_21.wav",
    "multi_event_train/syn_91.wav",
    "multi_event_train/syn_153.wav"
  ],
  'time_stamped_caption' = ['door slamming at 0.45-1.991, 3.019-5.8, 6.623-8.102','door knocking at 1.155-5.305','door knocking at
1.155-5.305','cow mooing at 1.592-4.602, 6.719-9.729 and explosion at 3.329-6.882'],
'frequencyCaption': ['door slamming three times','door knocking one times','cow mooing two times and explosion one times'],
  "question": [
    "Which audio clip has the highest frequency of door slamming, and at what timestamps do they occur?", "When does the cow moo in
the third audio?"
  ],
  "answer": [
    "Audio 1 contains door slamming three times: from about 0.45-1.991s, 3.019-5.8s, and 6.623-8.102s, which is more than any other
clip.", "Cow mooing can be heard 2 times at 1.592-4.602, 6.719-9.729"
  ]
}

Here is the audio data:
```

Figure 10: Temporal & Frequency based QA generation.

```
I am creating a training dataset for an Audio Language Model that can perform question answering on multiple audios. You will help me in creating the training dataset.
You are given up to five audio clips. For each audio clip, you have the following information:
        1.      Audio Path: A string representing the file's location.
        2.      Caption: A descriptive caption of the content (e.g., "A dog barking in the backyard," or "Rain pouring outside"). Note that this caption does not necessarily contain a transcript of speech because these audios primarily have non-speech events.
        3.      Time-Stamped Events: Labels of key acoustic events or sounds, along with their start and end times. For instance: ['(Background noise-0.000-1.144)','(Video game sound-0.000-10.000)','(Music-1.179-7.371)','(Sound effect-6.375-8.045)','(Music-9.034-10.000)'] which means the caption has "background noise from 0.000 seconds to 1.144 seconds, vide game sound from 0.000 to 10.000 seconds, music sound from 1.179 to 7.371 seconds, sound effects from 6.375 to 8.045 seconds and music sound from 9.034 to 10.000 seconds."

Your Task:
        1.      Generate questions focusing on differences among the provided audio clips. These differences can be about any audible elements such as the types of sounds, durations of events, overlaps of events, or notable acoustic aspects that might be inferred (e.g., loudness or pitch changes, even if you don't have exact numerical values).
        2.      Ensure that the question:
                -       Strictly compares or contrasts some aspect of at least two (or more) of the audio files.
                -       Is not about any visual component or transcript-based speech recognition, but rather about the sounds present and their time-stamped events.
                -       May optionally be a multiple-choice question (MCQ) or a short textual question.
        3.      Generate a corresponding, correct answer:
                -       This answer can be a short sentence or, in the case of MCQs, the correct option.
                -       The answer should be grounded in the captions and time-stamped events.
                -       Since these audios contain no explicit speech transcripts, your answer should reflect differences in non-speech audio events (e.g., "The first audio clip has a dog barking at the beginning, whereas the second has a car honking.").
        4.      No Follow-Up Questions: You are to provide only one Q-A pair per set of up to five audio clips.

Format Requirements:
        -       When you output the question and answer, maintain the following json structure:

{
  "audio_paths": [ "<path1>", "<path2>", ... ],  // up to 5
  "captions": [ "<caption1>", "<caption2>", ... ],
  "question": ["<Your first generated comparison question here>","<Your second  generated comparison question here>", …] // upto 5 questions
  "answer": ["<Correct answer for first question here>", "<Correct answer for second question here>", ….] // answers corresponding to each question
}

        -       audio_paths is a list of up to five paths.
        -       captions is a list of corresponding descriptive captions (one per audio).
        -       time_stamped_events is a list of lists, with each sub-list containing the labeled events for a specific audio.
        -       question is a list of questions that compares at least two of these clips.
        -       answer is a list of short sentence or the correct choice to the question.

Key Points:
        -       Focus your question on the differences in sounds, events, or other notable audio characteristics among the clips.
        -       The answer must be correct and should clearly reflect the content in the time-stamped events or captions.
        -       The goal is to train an Audio Language Model that can handle multiple audios at once and answer a single comparison question about them.

For example:
{
  "audio_paths": ["YJoQj-tobYOw.wav", "YSUxfKJP4bJ4.wav", "YsbY-zp5Lfew.wav", "YqExVrE3FyjM.wav" ],
  "captions": [
    "A man speaks with electronic music playing in the background at a martial arts gym.",
    "A sewing machine hums in the background as various tools are used intermittently in a workshop.",
    "A heavy engine starts and runs in the background noise.",
    "People are chewing and using machinery, with occasional breathing and tapping sounds."
  ],
  "question": ["Which audio has the heavy engine beginning after 3 seconds, and which one has vacuum noise lasting from 0 to 10 seconds?",
    "Between the first and fourth audios, which has continuous music from start to end, and which features chewing from around 2s to 5s?",
    "Comparing male speech events, which audio features multiple speech segments throughout, and which has minimal or no speech segments?"
  ],
  "answer": [
    "The third audio starts with a heavy engine sound after about 3 seconds, while the second audio has vacuum noise for the entire 0-10s duration.",
    "The first audio clip has continuous electronic music from 0-10s, whereas the fourth audio has chewing sounds between roughly 2s and 5s.",
    "Audio 1 has multiple male speech events spread across the 0-10s window, while Audio 2 is dominated by mechanical sounds (vacuum noise) and does not have male speech segments."
  ]
}

Here are the audio details:
```

Figure 11: Audio Difference QA generation.

You are assisting in creating a training dataset for a Large Audio Language Model (LALM) that can answer questions about multiple music audio clips at once.

You are given up to five audio clips. For each audio clip, the following metadata may be available (not all fields must be used in every question, and the LALM itself does not see these fields directly, only the raw audio):

1. name: The audio file path (e.g., /path/to/audio.mp3).
2. artist: The performer's name (e.g., "Cheryl").
3. song: The song title (e.g., "Rain on Me").
4. album_name: The album title.
5. tags/genres: Genres or descriptive tags (e.g., "pop," "rock," "instrumental").
6. release: The year of release (e.g., "2009").
7. danceability, energy, key, mode, valence, tempo, duration_ms: Numerical audio features that might indicate rhythmic feel, intensity, musical key/mode, emotional positivity, speed, or duration.
8. lang: The primary language of any vocals (e.g., "en" for English).

Possible Reasoning Types:
When generating your comparison question, pick one of the following reasoning types (or a similar audio-based angle) so that the LALM must rely on listening-based analysis. Each type is illustrated with a question example:

1. Instrument & Vocal Presence
   - Example: "Which audio track has a more dominant presence of electric guitar throughout most of its duration?"
2. Tempo & Rhythmic Analysis
   - Example: "Which of these two tracks maintains a higher average tempo from start to finish?"
3. Tonality & Key Detection
   - Example: "Which track sounds like it is in a minor key based on its chord progression and overall tonal color?"
4. Dynamic Range & Volume Variation
   - Example: "Which track shows a wider dynamic range from its beginning to its ending section?"
5. Language & Vocal Style Identification
   - Example: "Which track features lyrics in Spanish and a rap-style vocal delivery?"
6. Genre & Stylistic Interpretation
   - Example: "Which track best fits a symphonic metal style, considering its instrumentation and overall sound?"
7. Structural Form Analysis
   - Example: "Which track follows a verse-chorus-verse structure based on repeated melodic sections?"

Your task is to generate a Q-A pair for each set of two to five audio clips. This Q-A pair should:

1. Compare or contrast at least two of the audio clips on an aspect that requires "listening" rather than just reading text (e.g., comparing energy, tempo, presence of vocals, musical complexity, or any audio-derived feature).
2. Avoid purely textual or trivial metadata lookups—questions should be framed as if the model must analyze the raw audio.
3. Demand multi-level reasoning, not a simple yes/no or single-step answer. For instance, the question might ask which track has a higher tempo and also has a more upbeat feel, or which track shifts from minor to major key, etc.
4. Only one question in every question. There should not be nested questions.
5. Provide a correct answer grounded in what the audio would contain (inferred from the metadata).
6. Format each Q-A pair using the JSON structure below.
7. Make sure to refer the audios only by their position (like 1st audio, second audio) and not by their name or any other metadata.

JSON Output Format

For each set of 2-5 audios, output a single JSON object with the following fields:

```
{
  "audio_paths": ["<audio_path_1>", "<audio_path_2>", ...],
  "metadata": [
    {
      "artist": "...",
      "song": "...",
      "tags": "...",
      "release": "...",
      "danceability": "...",
      "energy": "...",
      ...
    },
    ...
  ],
"resoning_type": <reasoning_type from the above list>
  "question": ["<Your 1st reasoning question>","<Your 2nd reasoning question>…],
  "answer": ["<Correct answer for 1st question>", "<Correct answer for 2nd question>"…]
}
```

1. audio_paths: A list of the file paths (one for each audio).
2. metadata: A list of objects, each containing relevant fields (artist, song, tags, etc.) for the corresponding audio. You can include or omit fields as necessary.
3. question: A list containing the questions string.
4. answer: A list containing the correct answers corresponding to each question in the list.

Important Guidelines

1. Comparison Focus:
   - Your question must compare or contrast at least two audios in a way that necessitates listening.
   - Examples: Differences in tempo, key, vocal presence, energy levels, or changes over time in each track.
2. No Multi-Part Questions:
   - Only one direct question. (E.g., "Which track has higher energy?" rather than "Which track has higher energy, and which one has a key change?")
3. Audio-Centric:
   - The LALM is trained to listen, so the question should be answerable by analyzing the sound, not just reading textual metadata.
   - Avoid purely text-based or metadata-based questions like "Who is the artist?" or "Which track is from 2009?"
4. Answer:
   - The answer should be concise, directly addressing the comparison.
   - Reference the audio differences logically. For instance: "Track A has a faster tempo (around 110 BPM) compared to Track B's slower beat (around 90 BPM)."
5. No Visual or Speech Transcripts:
   - Stick to audio attributes like instrumentation, vocals, tempo, or mood.
   - If the track has vocals, focus on style or language, not detailed speech transcripts.

Example Output Snippet

Below is a fictional example demonstrating how to format one Q-A pair (assume we have two audios):

```
{
  "audio_paths": [
    "/path/music_track1.mp3",
    "/path/music_track2.mp3"
  ],
  "metadata": [
    {
      "artist": "Cheryl",
      "song": "Rain on Me",
      "tags": "pop,british,female vocalists",
      "release": "2009",
      "danceability": "0.635",
      "energy": "0.746"
    },
    {
      "artist": "Oddisee",
      "song": "After Thoughts",
      "tags": "instrumental hip-hop,underground hip hop",
      "release": "2013",
      "danceability": "0.591",
      "energy": "0.513"
    }
  ],
  "question": [
    "Which track has a higher overall energy level and is more likely to feature prominent vocals?"
  ],
  "answer": [
    "The first track has a higher energy score (0.746) and includes a distinct female vocal lead, whereas the second track's energy is lower (0.513) and is primarily instrumental."
  ]
}
```

Here is the input metadata:

Figure 12: Multi-Audio QA Generation for Music4All.