SAR-LM: SYMBOLIC AUDIO REASONING WITH LARGE LANGUAGE MODELS

Anonymous Authors
Anonymous Affiliations
anonymous@ismir.net

ABSTRACT

While large language models (LLMs) have made huge strides in text and vision, their ability to reason about sound remains limited. Most recent approaches rely on dense audio embeddings that are hard to interpret and often fail on tasks requiring fine-grained or structured understanding.

This project introduces SAR-LM, a symbolic audio reasoning pipeline that extracts structured, text-based features from audio across three aspects: speech, general sound, and music. For speech, we use Whisper-large and Wav2Vec2-based emotion recognition. For sound events, we rely on PANNs. For music, we combine low-level transcription from MT3, mid-level chord progressions from Chordino, and high-level tags from MusicNN. These symbolic features are used in two ways: either directly as flat prompts, or summarized into natural-language captions using Gemini 2.5 Pro. To evaluate performance, we compare both approaches against captions generated end-to-end from raw audio, and a mixed version using both symbolic and audio inputs.

We test all methods on the MMAU benchmark, which pairs audio clips with multiple-choice questions for audio understanding and reasoning across speech, music, and environmental sounds. We find that symbolic prompts can match or outperform dense baselines in several reasoning tasks. These findings suggest that symbolic audio inputs, combined with structured prompting, offer a promising path toward more accurate and explainable audio question answering with LLMs.

1. INTRODUCTION

Sound plays an important role in how people understand the world. We do not just hear sounds, we make sense of them. For example, we can guess who is speaking, how they feel, or what caused a noise in the background. This kind of reasoning comes naturally to humans, but it's still very difficult for AI systems.

Large language models (LLMs) have made major progress in understanding text, images, and code. But when it comes to audio, they still struggle. Most existing methods rely on dense audio embeddings, which are hard to interpret and not well suited for reasoning. These features are often noisy, unstructured, and hard to align with language-based models.

In this project, we take a different approach. Instead of giving the model raw audio or dense features, we convert the audio into symbolic, time-aligned text, a format that is more familiar to LLMs. Our pipeline extracts structured features from different parts of the audio, depending on its type. For example, we use Whisper for speech transcripts [1], Wav2Vec2 for speech emotion [2], PANNs for general sound events [3], MT3 for musical notes [4], Chordino for chord progressions [5, 6], and MusicNN for music tags [1] .

We build text-based prompts using these symbolic features, and optionally summarize them into natural-language captions using Gemini 2.5 Pro [7]. We then test how well an LLM (Qwen3-32B) [8] can answer multiple-choice questions from the MMAU benchmark [9], which pairs audio clips with questions to evaluate audio understanding and reasoning across speech, music, and environmental sounds, using different types of input: raw symbolic features, symbolic-based captions, and end-to-end captions generated from audio.

Our results show that symbolic prompts are competitive with end-to-end approaches, while offering much greater interpretability. This suggests that symbolic audio reasoning is a promising direction for building more transparent and controllable audio-language systems.

2. METHODOLOGY

2.1 Pipeline Overview

Our goal is to help large language models (LLMs) understand and reason about audio. To achieve this, we design a modular pipeline that converts raw audio into structured, interpretable prompts for a language model. The pipeline consists of four main stages: symbolic feature extraction, prompt construction, LLM-based reasoning, and answer prediction, as shown in Figure 1.

Given an input audio clip x, we extract symbolic features using pretrained models:

$$\mathcal{F}(x) = \{f_1, f_2, \ldots, f_n\},$$

where each f_i is a discrete, time-aligned feature such as

[1] https://github.com/jordipons/musicnn

a transcript, tag, or chord sequence. These features are filtered and composed into a textual prompt $p = \mathcal{T}(\mathcal{F}(x), s)$, where s denotes the selected prompt style (e.g., flat, conditional, caption-based). The prompt is paired with a question q and passed to a large language model $\mathcal{M}$, which produces a predicted answer $\hat{y} = \mathcal{M}(p, q)$.

We support multiple prompt styles, including a simple flat format and variants that incorporate audio captions generated by Gemini 2.5 Pro [7], either from the raw audio or from symbolic features. In some styles, we apply prompt-level restrictions (e.g., "Do not overthink") to reduce hallucinations. The predicted answer is then evaluated against the ground-truth label from the MMAU benchmark.

This pipeline is fully modular and text-based. Each component, feature extractor, prompt generator, or language model, can be modified independently without retraining the whole system. This makes the setup highly extensible for future experiments. Figure 1 illustrates the full architecture of the SAR-LM pipeline.

2.2 Symbolic Feature Extraction

Rather than relying on dense audio embeddings, which can be difficult for language models to interpret, we convert each audio clip into a set of symbolic, time-aligned features that are easier to read and reason about. These features are extracted using a suite of pretrained models, each targeting a different semantic layer of the audio signal, such as sound events, speech, emotion, or music. All symbolic features are represented in plain text and aligned to the timeline of the audio clip.

As a first step, we run PANNs [3] to generate a multi-label set of audio event tags. These tags are used as a coarse guide to determine the content of the clip. If PANNs predicts the presence of speech, we extract transcriptions and emotion cues. If it detects music, we extract symbolic music features such as notes, chords, and stylistic tags. This adaptive filtering helps reduce noise and ensures that only relevant information is included in the prompt.

Sound event tags. PANNs provides a list of timestamped labels that describe the audio scene, including categories like music, laughter, footsteps, and speech. These tags form the backbone of our filtering logic and are also included directly in the prompt to provide a high-level summary of the audio.

Speech transcription. If speech is present, we extract a full transcript using Whisper-large [1]. Whisper performs well even on multilingual or noisy clips and produces stable outputs that support reasoning about content, speaker identity, or dialogue structure.

Emotion recognition. When speech is detected, we extract emotional cues using the DAWN Transformer [2], which predicts continuous values for valence, arousal, and dominance (VAD). These values provide a fine-grained affective profile of the speaker but are not directly usable by language models. To convert them into interpretable symbolic tags, we discretize each dimension into `low`, `mid`, or `high` bins using dataset-specific thresholds derived from empirical value distributions.

Music transcription. For clips containing music, we use MT3 [4], a multitask transformer model that outputs symbolic MIDI sequences, including pitch, instrument, and note timing information. These MIDI files are post-processed using `pretty_midi` [10] to extract a structured list of symbolic note events, each annotated with note name, pitch value, instrument type, and onset/offset times. This allows the model to reason about musical structure, such as which instruments are playing, when notes occur, or how melodies evolve over time.

Chord progression. To capture harmonic structure, we use Chordino [5, 6] to identify chord sequences with their temporal boundaries. Chords offer a mid-level abstraction of the audio and support tasks involving musical progression or genre understanding.

Music tagging. To complement low- and mid-level musical features, we apply Musicnn [2] to produce high-level tags that reflect genre and timbral qualities (e.g., classical, electronic, solo, bright). These tags offer semantic grounding for questions related to mood or style.

Each audio clip is processed selectively: only features relevant to the content type are included.

2.3 Prompt Construction

Once we extract symbolic features, we convert them into a natural language prompt suitable for input to a language model. The goal is to describe the audio content in a way that supports downstream reasoning tasks. Since each audio clip varies in modality, we dynamically construct the prompt based on available features.

We begin by analyzing each clip using PANNs to identify its high-level content. If PANNs detects speech or speech-like events, we include a transcript generated by Whisper, as well as a predicted speech emotion label derived from valence-arousal-dominance (VAD) scores. If PANNs detects music, we incorporate symbolic note sequences from MT3, chord progressions from Chordino, and music tags from musicnn. In all cases, we include both clipwise and timestamped sound events from PANNs to provide a general overview of the acoustic scene.

All extracted features are formatted as plain text using a consistent, readable structure. Irrelevant features are filtered out for each clip to reduce noise and keep the prompt focused. An overview of the prompt construction process is shown in Figure 1.

After the symbolic features, we append the question and multiple-choice options, followed by a fixed instruction block that guides the model's decoding. These instructions tell the model to select one answer verbatim from the provided options without guessing or adding extra words.

2.4 Caption Generation

In addition to constructing symbolic prompts, we generate natural language captions using Gemini 2.5 Pro [3] via the

[2] `https://github.com/jordipons/musicnn`
[3] `https://deepmind.google/technologies/gemini/#gemini-25`

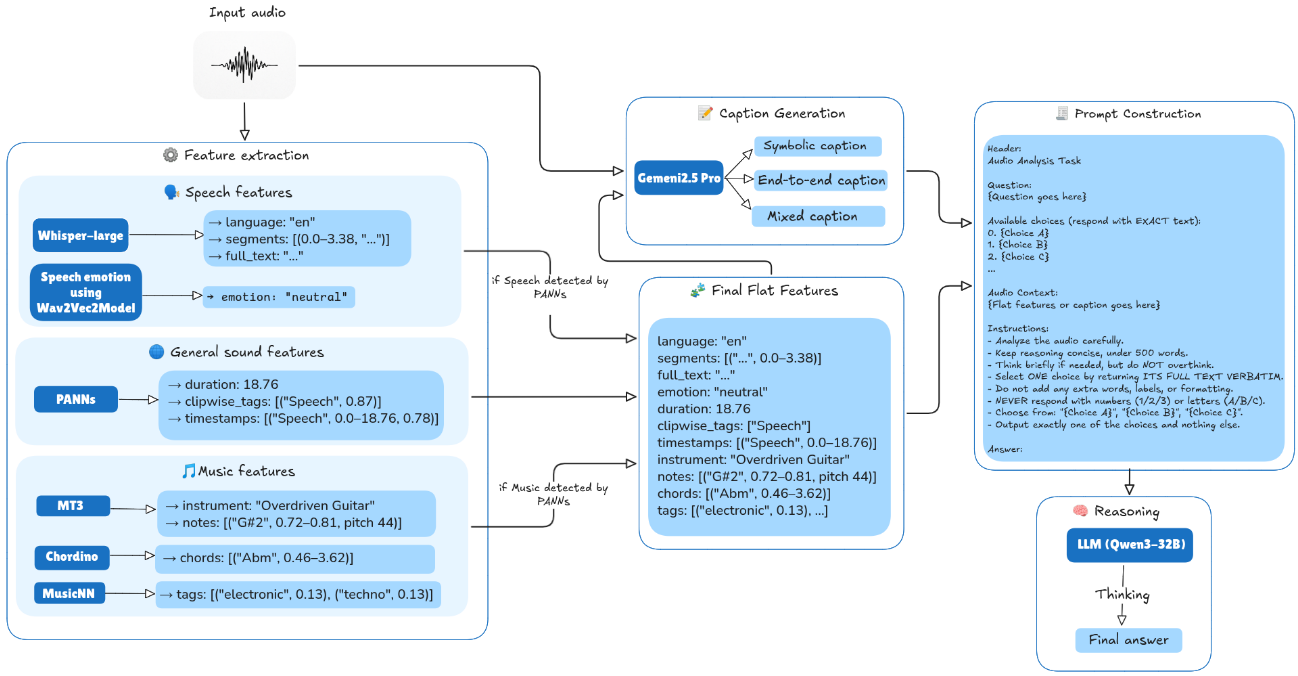

Figure 1. Overview of the SAR-LM pipeline. Given an input audio clip, PANNs is first used to predict sound event tags, which serve as a reference to determine the presence of speech, music, or general environmental sounds. Based on this, relevant symbolic features are extracted using specialized models: Whisper-large and Wav2Vec2.0 for speech transcription and emotion, MT3 for low-level musical notes, Chordino for chord progression, and Musicnn for high-level music tags. The full set of symbolic features is then used to construct three types of prompts: (1) a flat symbolic prompt containing all time-aligned features, (2) a structured caption generated from symbolic features using Gemini2.5-pro, and (3) an end-to-end caption directly generated from raw audio. All prompts are paired with an MMAU question and passed to Qwen3 (32B), which produces an answer. Predictions are compared to ground-truth answers to evaluate performance. The pipeline is fully modular, enabling flexible substitution of feature extractors, prompt styles, and reasoning models.

Google GenerativeAI API. These captions serve two purposes: (1) they provide a more human-readable representation of the audio scene, and (2) they allow us to compare different levels of abstraction for prompting. We generate three types of captions in total: symbolic, end-to-end, and mixed.

Symbolic caption. To generate symbolic captions, we first build a structured text prompt from the extracted symbolic features, reformatting them into readable bullet points and time-aligned descriptions. Each prompt includes detected sound events (PANNs), musical content (MT3, MusicNN, Chordino), and speech-related information (Whisper transcript, segments, emotion), depending on what is present in the clip. All prompts use the same fixed instruction. This prompt is passed to Gemini 2.5 Pro via the Google GenerativeAI API. The model returns a paragraph-style caption summarizing the audio scene, which is stored for downstream reasoning.

End-to-end caption. To establish a baseline, we generate captions directly from raw audio using the same instruction. Instead of symbolic input, we provide the waveform as audio bytes. This allows us to evaluate the impact of symbolic conditioning on content quality and hallucination reduction.

Mixed caption. To explore whether combining raw audio with symbolic features leads to richer descriptions, we generate mixed captions by providing both as input to Gemini 2.5 Pro. Each prompt includes the audio clip along with the structured symbolic text used in the symbolic caption setting.

We use the same fixed instruction and zero-shot setup as before. The model processes both modalities simultaneously and returns a fluent paragraph.

2.5 Reasoning with Language Models

We evaluate symbolic reasoning by testing how well Qwen3-32B [4], an open-source LLM, answers audio-based multiple-choice questions using non-audio inputs. We run the model locally using HuggingFace Transformers with deterministic decoding.

We test three input types:

1. **Flat symbolic features:** Raw features serialized into plain English (e.g., Whisper, PANNs, MT3, Chordino, MusicNN)
2. **Symbolic captions:** Natural captions generated by Gemini 2.5 Pro using symbolic inputs
3. **End-to-end captions:** Captions generated directly from raw audio

[4] https://huggingface.co/Qwen/Qwen3-32B

Each input is wrapped in a structured prompt with the corresponding question and answer choices from the MMAU benchmark. To address cases where Qwen3 overthinks simple questions and produces long internal reasoning (sometimes exceeding the token limit), we include explicit instructions discouraging overthinking and enforcing strict output formatting. This approach ensures stable decoding and prevents truncation.

3. EXPERIMENTS AND RESULTS

3.1 Setup

Our initial setup used Qwen3-32B with prompt restrictions to reduce overthinking. While this approach worked in most cases, we still observed occasional hallucinations and unnecessarily long reasoning chains, which sometimes reduced accuracy. To address these limitations, we tested the updated Qwen3-30B-A3B-Instruct-2507 [8], which avoids overthinking, follows instructions more reliably, and delivers faster inference. This model became the focus of our final analysis, and all subsequent statistical tests are performed on its results.

We evaluate all predictions using the MMAU benchmark [9], a large-scale testbed for audio understanding and reasoning. Each sample consists of an audio clip paired with a natural language question and four multiple-choice answers, requiring models to recognise acoustic events and integrate contextual cues to select the correct option. The benchmark spans 27 task types across speech, music, and environmental domains, covering challenges such as speaker identification, instrument recognition, temporal event ordering, and emotion detection. While the full benchmark contains over 91,000 samples, we use the `test-mini` split of 1,000 samples, which is the only split with publicly available ground-truth answers.

Following the official MMAU evaluation script[5], we use a string-matching function where a prediction is considered correct if it contains all key tokens from the reference answer and none from the incorrect options. This fuzzy matching accounts for minor wording variations while ensuring answer precision.

3.2 Dynamic feature selection with a GPT-style agent

We also test a dynamic variant that lets a GPT-style agent (Gemini 2.5 Pro[6]) choose which symbolic tools to use for each sample. We give the agent a short description of the available tools (Whisper, PANNs, MT3, Chordino, Musicnn, speech emotion) and ask it to return a JSON object with the selected tools. We then build the prompt accordingly and run Qwen3-30B-A3B-Instruct-2507 [8] for answer prediction. The evaluation setup is the same as before.

[5] `https://github.com/Sakshi113/mmau`

[6] `https://deepmind.google/technologies/gemini/#gemini-25`

3.3 Comparison with Baseline Methods

To contextualize our results, we compare our symbolic approaches with prior benchmark methods reported in the MMAU paper and related work. Table 1 presents a task-wise breakdown of accuracy for three baselines: MMAU (Best), Audio-CoT, and Audio-Reasoner, alongside our main variants and agent-controlled versions.

Table 1. Comparison with baseline methods (task-wise accuracy)

Method	Sound (%)	Music (%)	Speech (%)
MMAU (Best)	57.35	50.98	64.86
Audio-CoT	62.16	57.78	56.16
Audio-Reasoner	60.06	**64.30**	60.70
Flat Symbolic Features (ours)	69.37	56.59	73.87
Symbolic Gemini Captions (ours)	69.67	58.38	71.77
E2E Gemini Captions (ours)	68.17	62.28	69.97
Mixed Gemini Captions (ours)	71.77	61.08	72.97
Flat Symbolic (agent)	72.67	57.78	73.87
Symbolic Captions (agent)	70.57	58.08	70.27
Mixed Captions (agent)	**74.77**	63.17	**73.87**

Our symbolic methods substantially outperform existing approaches in sound and speech reasoning tasks. Compared to Audio-Reasoner, flat symbolic features improve accuracy by +9.31% on sound and +13.17% on speech tasks. Symbolic Gemini captions also show strong speech performance (71.77%), and the mixed caption approach reaches 72.97%.

The agent-controlled mixed variant achieves the highest sound accuracy (74.77%) and improves music accuracy to 63.17%, outperforming all baselines and non-agent variants in these categories. However, for speech tasks, the non-agent flat method still slightly leads.

While Audio-Reasoner remains strong in music (64.30%) among the baselines, both our non-agent E2E (62.28%) and agent mixed (63.17%) approaches are competitive, with the added benefit of richer interpretability.

We also evaluated the open-source Qwen1.5-1.8B model using flat symbolic prompts and observed a drop in overall performance (55.8% accuracy), particularly in speech tasks. This confirms that model scale and alignment play a critical role in symbolic reasoning performance.

Together, these comparisons highlight the strength of symbolic representations, especially when paired with high-capacity, instruction-following language models, and the added benefits of per-sample tool selection when symbolic features are combined with raw audio in a mixed-caption setting.

4. CONCLUSION

We presented SAR-LM, a symbolic audio reasoning pipeline that converts audio into interpretable text features for LLMs. Symbolic inputs perform competitively with end-to-end captions and provide greater transparency. Mixed captions that combine symbolic and raw audio achieve the highest scores, and agent-controlled selection further improves results. These findings show that symbolic reasoning is a promising path toward more accurate and explainable audio question answering.

5. REFERENCES

[1] A. Radford *et al.*, "Robust speech recognition via large-scale weak supervision," *arXiv preprint arXiv:2212.04356*, 2023.

[2] J. Huang *et al.*, "Dawn: A transformer-based framework for emotion recognition from speech," in *Proc. Interspeech*, 2023.

[3] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, M. D. Plumbley *et al.*, "Panns: Large-scale pretrained audio neural networks for audio pattern recognition," in *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, 2020, pp. 2880–2894.

[4] J. Gardner *et al.*, "Mt3: Multi-task multitrack music transcription," in *Proc. International Conference on Learning Representations (ICLR)*, 2021.

[5] M. Mauch and S. Dixon, "Chordino: A vamp plugin for extracting chord sequences from audio," in *Proc. ISMIR*, 2009, pp. 215–218.

[6] L. Holloway *et al.*, "A deep learning approach to chord extraction from audio," in *Proc. ICASSP*, 2021, pp. 556–560.

[7] G. Team, "Gemini 2.5 technical report," arXiv preprint arXiv:2502.04600, 2025.

[8] J. Bai *et al.*, "Qwen3 technical report," arXiv preprint arXiv:2503.06710, 2025.

[9] A. Sakshi *et al.*, "Mmau: A massive multi-task audio understanding and reasoning benchmark," *arXiv preprint arXiv:2402.10954*, 2024.

[10] C. Raffel, B. McFee, E. J. Humphrey, J. Salamon, O. Nieto, D. Liang, and D. P. Ellis, "mir_eval: A transparent implementation of common mir metrics," in *Proc. ISMIR*, 2014.