QWEN2.5-OMNI WITH ALL-AUDIO AUDIO-TAH FOR AUDIO-DEPENDENT QUESTION ANSWERING

Technical Report

Chenglin Wu[1], *Daiqing Wu*[2], *Yuxi Huang*[1]

[1] Xiamen University, Xiamen, China
[2] Tsinghua University, Beijing, China
chenglinwu@stu.xmu.edu.cn, wudaiqing04@gmail.com, huangyuxi@stu.xmu.edu.cn

ABSTRACT

This report describes the method used in our submission for DCASE 2026 Task 5. The system uses the Qwen2.5-Omni base checkpoint as the audio-language backbone and applies an Audio-TaH-style all-audio rerun mechanism for audio-dependent multiple-choice question answering. Given an audio clip, a question, and answer choices, the model first builds an audio-conditioned prompt context with Qwen2.5-Omni. Before decoding, we perform an additional thinker pass that retains all audio-token positions. This selection-free rerun avoids hard-audio chunk selection and is intended to preserve access to the full set of audio evidence before final answer generation.

***Index Terms*—** audio-dependent question answering, Qwen2.5-Omni, Audio-TaH, audio-language model, audio reasoning

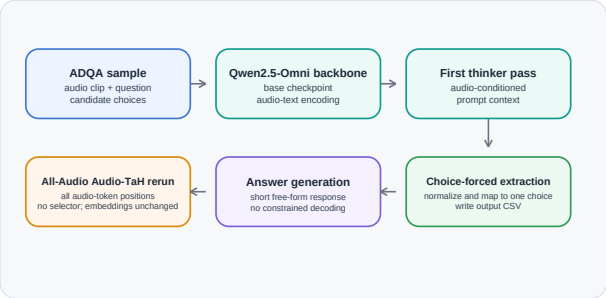

Figure 1: Overview of the proposed ADQA method. The system uses Qwen2.5-Omni as the audio-language backbone, applies an all-audio Audio-TaH rerun over audio-token positions, and obtains the final prediction through choice-forced answer extraction.

1. SYSTEM OVERVIEW

The system targets audio-dependent multiple-choice question answering. Given an audio clip, a question, and answer choices, the Qwen2.5-Omni based model encodes the audio and text prompt into an audio-conditioned context. On top of this backbone, we apply an Audio-TaH-style latent rerun mechanism so that the model can revisit audio-token positions before producing the final answer.

The submitted system consists of two main components: a Qwen2.5-Omni audio-language backbone and an all-audio Audio-TaH rerun procedure. The backbone provides audio-text understanding ability, while the rerun procedure increases latent computation over audio-token positions during answer generation. We use the Qwen2.5-Omni base checkpoint as the backbone, rather than an additional supervised fine-tuned AQA checkpoint.

2. ALL-AUDIO AUDIO-TAH RERUN

The main method component is a selection-free Audio-TaH rerun. In the first thinker pass, Qwen2.5-Omni processes the audio tokens together with the question and answer choices, producing an audio-conditioned hidden context. Before answer generation, we perform an additional prompt-level thinker pass that includes all audio-token positions. The rerun therefore uses the complete audio-token set instead of selecting a sparse subset of hard audio chunks.

This design differs from a hard-audio selection strategy. Instead of first deciding which audio chunks are difficult or important, the method reruns all audio-token positions. This removes the need for hard-audio labels or a learned token selector, and avoids discarding potentially useful audio evidence at a separate selection stage.

The rerun keeps the original audio-token embeddings unchanged. Therefore, the method does not introduce an additional audio encoder or directly modify the audio representation itself. The core operation is repeated latent computation inside the Qwen2.5-Omni thinker. The main trade-off is that full audio-token rerun increases evidence coverage at the cost of additional computation compared with sparse rerun.

3. ANSWER GENERATION AND EXTRACTION

After the all-audio Audio-TaH rerun, the model generates a short response for the multiple-choice question. Constrained choice decoding is not used during generation. Instead, the final prediction is obtained by extracting the answer span from the model output and mapping it back to one of the provided choices.

The post-processing removes option prefixes such as A/B/C/D, normalizes whitespace and punctuation, and compares the extracted response with the candidate choices. For the official no-answer evaluation set, the submitted answer is forced to be one of the provided choice strings before writing the output CSV. This choice-forced post-processing preserves the multiple-choice submission protocol and reduces errors caused by paraphrases, missing articles, or superficial formatting differences.