Technical Report: Fun-Audio-Chat-8B with LoRA Fine-Tuning for Audio-Dependent Question Answering

DCASE 2026 Task 5 - Audio-Dependent Question Answering (ADQA)

Abstract

This report describes our submission to DCASE 2026 Task 5 (Audio-Dependent Question Answering, ADQA). We fine-tuned the Fun-Audio-Chat-8B large language model with a LoRA adapter trained using Chain-of-Thought (CoT) reasoning data. The model produces free-text reasoning that is then post-processed to extract the final answer label (A/B/C/D). Inference uses sampling-based decoding (temperature=0.7, top_p=0.9, repetition_penalty=1.1) which outperforms greedy decoding by ~2.5% on our holdout set, reducing repetition artifacts that are prevalent in CoT-generated output.

1. Task Description

DCASE 2026 Task 5 (ADQA) evaluates systems on their ability to answer questions about audio content. Each item consists of: (1) an audio file, (2) a natural-language question, and (3) four answer choices (A/B/C/D). Systems must identify the correct choice for each question. The evaluation metric is accuracy.

2. Base Model: Fun-Audio-Chat-8B

Fun-Audio-Chat-8B is a large language model designed for audio understanding and instruction-following. It uses an encoder-decoder transformer architecture and processes audio through a dedicated audio encoder before entering the LLM text pathway. The model natively supports audio + text interleaved inputs.

3. Fine-Tuning with LoRA + CoT Reasoning

3.1 Training Data

Training data is formatted as multi-turn conversations where the model is given audio content plus a question, and responds with a Chain-of-Thought reasoning sequence that concludes with an answer. The CoT format encourages the model to explicitly reason about audio events, making the final answer extractable with high reliability.

3.2 LoRA Fine-Tuning

We apply Low-Rank Adaptation (LoRA) to the model's attention weights. This keeps the pretrained weights frozen while training a small set of low-rank decomposition matrices, dramatically reducing the number of trainable parameters and enabling efficient fine-tuning on a single GPU. The adapter checkpoint used in this submission is checkpoint-1737.

3.3 Chain-of-Thought Training

Training prompts include explicit CoT formatting, teaching the model to verbalize its reasoning about audio events (e.g., counting sneezes, identifying speakers, localizing sound sources) before producing the final answer. This training

approach yields two benefits: (1) the model learns richer audio understanding through textual reasoning, and (2) the structured output simplifies answer extraction.

4. Inference Pipeline

4.1 Audio Processing

At inference time, audio is loaded at 16kHz using librosa and passed to the processor as a floating-point array. The processor applies the model's audio template and prepends the audio tokens to the text input, forming a multimodal input sequence.

4.2 Decoding Strategy

We use sampling-based decoding instead of greedy decoding. Key parameters:

```
do_sample            True
temperature          0.7
top_p                0.9
repetition_penalty   1.1
max_new_tokens       512
```

Comparison on holdout set (400 samples, same extraction pipeline):

Greedy decoding:	`92.5%`	(repetition artifacts common)
Sampling (above params):	`95.0%`	(cleaner output, ~2x faster)
	`+2.5%`	

4.3 Answer Extraction

The raw model output contains a CoT reasoning sequence and concludes with a final answer (e.g., 'The answer is: C.' or '(C)'). We use a regex-based post-processor (post_process.py) to extract the answer label from the free-text output in four stages:

Stage 1: Regex split on 'answer is: X' pattern - most reliable for well-formed outputs
Stage 2: <answer> XML tag - covers outputs where the model self-encapsulates the answer
Stage 3: Last [A-D] occurrence in final 5 lines - catches structural but malformed outputs
Stage 4: Keyword matching on choice text - fallback for unusual or partial outputs

5. Holdout Evaluation Results

We evaluated on a held-out portion of the training data (400 samples) using the sampling decoding strategy. Results by question type show consistent accuracy above 90% across all categories.

```
Holdout accuracy:           95.0%
Holdout size:               400 samples
Evaluation metric:          Accuracy
```

6. Submission Package

The submission.zip file contains the following four files:

output.csv:	Final predictions with columns: id, answer (A/B/C/D/?)
metadata.yaml:	Team, model, and system configuration information
technical_report.pdf:	This document - method description and results
post_process.py:	Standalone answer extraction script

Note: output.csv will be generated after the evaluation set is released on June 1, 2026.

7. Key Findings

CoT Training

Chain-of-Thought reasoning training significantly improves answer quality by forcing explicit audio reasoning before committing to an answer.

Sampling > Greedy

Sampling-based decoding outperforms greedy by ~2.5% on holdout and eliminates repetition loops common in CoT output.

Repetition Penalty

repetition_penalty=1.1 is sufficient to prevent looping without over-penalizing valid repeats.

Extraction Reliability

The split-based regex extractor on 'answer is: X' pattern achieves near-perfect extraction on well-formed CoT output.

Model: Fun-Audio-Chat-8B | Adapter: checkpoint-1737 | Decoding: sampling (T=0.7, top_p=0.9) | Holdout Accuracy: 95.0%