AUDIO QUESTION ANSWERING WITH GRPO-BASED FINE-TUNING AND CALIBRATED SEGMENT-LEVEL PREDICTIONS

Technical Report

Marcel Gibier, Nolwenn Celton, Raphaël Duroselle, Pierre Serrano, Olivier Boeffard, Jean-François Bonastre

Inria Paris, LR2
Paris, France
prenom.nom@inria.fr

ABSTRACT

In this report, we describe our submission to Track 5 of the DCASE 2025 Challenge for the task of Audio Question Answering (AQA). Our system leverages the SSL backbone BEATs to extract frame-level audio features, which are then processed by a classification head to generate segment-level predictions of acoustic events, following the Audioset ontology. These segment-level predictions are subsequently calibrated before producing event-level predictions. Finally, these predictions are incorporated into a structured prompt, along with the question and candidate answers. This prompt is then fed to a fine-tuned version of Qwen2.5-7B-Instruct, trained using the GRPO algorithm with a simple reward function. Our method achieves an accuracy of 62.6 % on the development set, demonstrating the effectiveness of combining acoustic event reasoning with instruction-tuned large language models for AQA.

***Index Terms*—** Audio question answering, Sound event detection, Classification, Calibration, GRPO

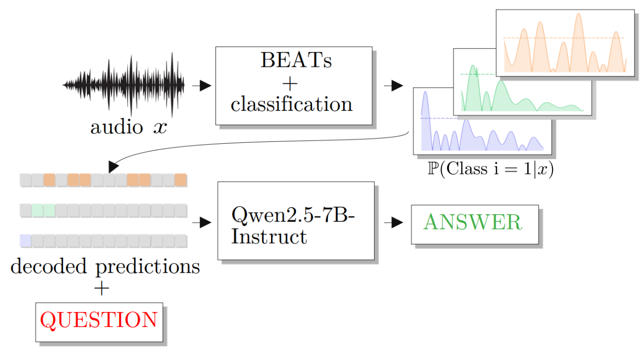

Figure 1: Our audio question-answering system is based on two main steps: (i) we decode the class presence probabilities output by the sound event detection model to obtain a time-localized, textual sound description of the audio; (ii) this text, along with the question, is then provided to a language model, allowing it to answer without directly analyzing the audio.

1. INTRODUCTION

The task of Audio Question Answering (AQA) involves providing an answer to a question about an audio clip. The question can be open-ended or closed, but for the first edition of this challenge, it exclusively consists of multiple-choice questions with several possible answers each. High-performing solutions for this task typically involve multimodal models, and three such models have been proposed as baselines: Qwen2-Audio-7B [1], AudioFlamingo2 [2], and Gemini-2.0-Flash.

These models typically extract audio representations from each frame using an encoder, such as an Audio Spectrogram Transformer [3]. The representations are then projected into an embedding space matching the dimensionality of the language model's textual inputs, which can be either shared or separate. Audio can be provided either as learned queries through a Q-Former [4], as a prefix to the text tokens, or via a cross-attention mechanism within the language model [5], which is trained using causal language modeling to answer questions about the audio using datasets such as OpenAQA [6]. If the language model's output does not exactly match one of the four expected answers, techniques such as regular expressions or similarity scoring with Sentence-BERT can be used to evaluate the response.

In our 2025 proposal, we hypothesize that many questions regarding audio content can be addressed using solely a timestamped list of detected acoustic events. This approach presents two major challenges. First, it requires a model capable of linguistic reasoning from this structured representation to formulate relevant responses. Second, the reliability of these responses heavily depends on the confidence in the scores provided by the acoustic event detection module. Essentially, good reasoning depends on accurate and well-calibrated detection.

We present two contributions in this work: (i) we incorporate a calibration method to ensure the reliability of the sound events provided to the language model, (ii) we investigate the benefits of fine-tuning using the Group Relative Policy Optimization (GRPO) algorithm as an alternative to conventional Supervised Fine-Tuning (SFT) for the AQA task. Regarding the inference strategy, we evaluate which answer option makes the complete sequence most probable by estimating the token-by-token likelihood, considering only the tokens corresponding to each candidate option.

Our single submitted system achieved an accuracy of 62.6%, significantly outperforming the best baseline score of 52.5% obtained by Gemini-2.0-Flash.

2. METHOD

2.1. Calibration of segment-level sound event predictions

Our system relies on a sound event detection (SED) system that outputs event-level predictions constituted of an event label, and start and end timestamps. We use the pretrained audio transformers proposed by [7]. The SED system is constituted of several components: an audio encoder, a classification head at the segment-level that outputs posterior probabilities for each event class, and a post-processing algorithm that produces event-level predictions (median filtering and threshold on segment-level posterior probabilities).

We observed that the values of the posterior probabilities on the DCASE corpus did not seem to reflect the relative proportion of target and non target segments. This property is called miscalibration. It can lead to suboptimal event-level predictions because the same threshold applied to different classes can correspond to different compromises between false alarm and false rejection.

2.1.1. Logistic regression

Rather than calibrating the posterior probabilities directly — which are often biased by class frequency imbalances in the evaluation data — we chose to calibrate the likelihood ratios (LR). These scores provide a measure that is independent of prior distribution assumptions.

$$\mathrm{LR}(x) = \left(\frac{P(x \mid y=1)}{P(x \mid y=0)}\right) \tag{1}$$

To this end, we trained a separate logistic regression model for each class, aimed at transforming the raw LRs produced by the BEATs model into calibrated scores. We used the implementation of [8]. The reduction of the calibration error can be evaluated by comparing the values of a proper scoring rule such as C_{llr} before and after calibration [9].

$$\begin{aligned} C_{llr} = \frac{1}{2}\Bigg(& \frac{1}{N_{y=1}} \sum_i \log_2\left(1 + \frac{1}{\mathrm{LR}_{(y=1),i}}\right) \\ & + \frac{1}{N_{y=0}} \sum_j \log_2\left(1 + \mathrm{LR}_{(y=0),j}\right)\Bigg) \end{aligned} \tag{2}$$

with $\{y=1\}$ and $\{y=0\}$ indicating the segments with and without the target event.

2.1.2. Priors adjustment

After calibration, we estimated the class prior probabilities $P(y = 1)$ using the class distribution observed in the training subset. This allowed us to convert the calibrated LLRs back into posterior probabilities, which were then evaluated with reliability curves.

$$P(y=1|x) = \frac{LR \times P(y=1)}{LR \times P(y=1) + (1 - P(y=1))} \tag{3}$$

2.2. Using GRPO for Question Answering

Reinforcement Learning from Human Feedback (RLHF) is a fine-tuning method used to align a language model's responses with human preferences. This approach is based on reinforcement learning, where the goal is to optimize a policy, that is, the generative model itself, by maximizing a reward function. To guide this optimization, a reward model trained on human feedback is used to evaluate the quality of the model's responses. Traditionally, a reinforcement learning architecture also includes a critic, a separate neural network responsible for estimating the expected value of the actions taken by the policy. The critic helps stabilize training by providing a more accurate estimate of the advantage, which represents the difference between the received reward and the expected reward.

We adopt Group Relative Policy Optimization (GRPO) for fine-tuning via RLHF [10]. In contrast to other reinforcement learning methods, GRPO removes the need for a separate critic model by using the average reward across multiple generations per prompt from the policy itself to establish a baseline for advantage estimation. This choice is also motivated by the fact that RLHF, particularly using GRPO, significantly outperforms supervised fine-tuning on downstream tasks with limited data [11, 12].

For each question q, in the format described in Box 2.2, the model produces a set $G = \{o_i\}$ of candidate answer tokens, where each $o_i \in \{\mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{D}\}$ corresponds to one of the multiple-choice options. We denote $\pi_\theta(o_i \mid q) = P(\text{token} = o_i \mid q)$ as the probability distribution over the elements of G where θ represents the model's parameters.

We define a simple reward function that assigns a value of 1 if the output o_i is the correct answer, and 0 otherwise. Our objective is to assess whether the response o_i outperforms the group's average response. To encourage the model to increase the probability $\pi_\theta(o_i)$, the advantage function is defined as follows:

$$\hat{A}_i^{\text{GRPO}} = \frac{R_i - \bar{R}}{\sigma_R} \tag{4}$$

where R_i is the reward for the i-th response, $\bar{R}$ is the mean reward across the group, and σ_R is the corresponding standard deviation.

In the end, the goal is to increase the contribution of responses with high advantage, so we aim to maximize the following quantity, which is a bit simpler than the original DeepSeek-R1 method:

$$\begin{aligned} \mathcal{J}_{\text{GRPO}}(\theta) = E_{q \sim P(Q)} \Bigg[& \frac{1}{|G|} \sum_{i=1}^{|G|} \\ & \min\left(r_i \hat{A}_i^{\text{GRPO}},\ \text{clip}(r_i,\ 1-\epsilon,\ 1+\epsilon) \hat{A}_i^{\text{GRPO}}\right)\Bigg] \\ & - \beta\, D_{\text{KL}}\left[\pi_\theta \parallel \pi_{\text{ref}}\right] \end{aligned} \tag{5}$$

where:

- $r_i = \frac{\pi_\theta(o_i|q)}{\pi_{\theta_{\text{old}}}(o_i|q)}$ is the probability ratio, which quantifies how much the new policy π_θ differs from the old policy $\pi_{\theta_{\text{old}}}$ for the predicted token o_i. The current policy π_θ is optimized relative to the previous policy $\pi_{\theta_{\text{old}}}$, which serves as the fixed reference from the last optimization step, while a fixed reference policy π_{ref}, typically the pretrained model, ensures stability.
- $D_{\text{KL}}[\cdot]$ denotes the KL divergence penalty, used to regularize the policy updates.
- The clipping operation and the minimum function prevent drastic policy changes, thereby ensuring training stability.

Audio question answering prompt used to query language model

Answer the question using **ONLY** the timestamped events. Your answer must be **EXACTLY** one of the provided options (A/B/C/D).

—

[Timestamped Events]
`<EVENTS>`
—
[Question]
`<QUESTION>`
—
[Options]
`<CHOICES>`
—

Rules:

1. Use **ONLY** the provided events.
2. If unsure, choose the option most consistent with the evidence — **never** output "N/A" or "N".
3. Output **ONLY** the letter of the correct option (A/B/C/D), with **NO** punctuation, text, or explanations.

Box 2.2. This prompt provides the model with strict instructions to select only the letter identifying the correct answer, using the provided timestamped events as support.

3. SUBMISSIONS AND RESULTS

3.1. Audio question answering dataset

The dataset provided for the challenge [13] consists of three types of audio/question pairs: Bioacoustics QA [14], Temporal Soundscapes QA, and Complex QA [15]. Table 1 summarizes the proportions of these benchmarks.

The first part (Part 1) evaluates how well the models can adapt to diverse acoustic conditions. The second part (Part 2) assesses the model's ability to detect the start, end, and transitions of different acoustic events. The final part (Part 3) evaluates the model's capability to answer complex questions involving a diverse set of real-world audio scenarios.

Split	Part1	Part2	Part3	Total
train	740	1038	6443	8221
dev	224	609	1633	2466
eval	1480	1004	2400	4884

Table 1: Dataset splits for each QA category

3.2. Sound event detection dataset

The BEATs modelhave been finetuned on AudioSet strong train set by the authors of [7]. The logistic regression models are trained and evaluated on subsets of the AudioSet strong evaluation set [16].

Calibration models are trained on a restricted subset of the AudioSet evaluation set, consisting of 13,596 audio clips (approximately 1,142,064 samples, retaining 84 frames of 40 ms per audio out of the 250 frames typically used in the BEATs model). These audio samples were selected randomly, which may introduce a class distribution bias. Furthermore, the evaluation was conducted on another limited subset of only 1,700 audio clips (142,800 samples).

3.3. Evaluation metrics

We optimize our submitted system based on the accuracy, which corresponds to the overall evaluation accuracy (i.e., accuracy computed according to the proportion of the three parts).

3.4. Submitted system

Our single submission is based on a cascaded model architecture comprising two main modules, totaling 7.7 billion parameters.

First, we use the BEATs audio encoder [17], finetuned for strong event detection [7] to extract posterior probabilities for each sound event class of the Audioset ontology [18] at the segment level. These probabilities are then calibrated, and a threshold of 0.1 is applied to derive event-level predictions. Each prediction is then converted into a text string following the format: `{class}_{start_time}_{end_time}`. These strings are incorporated into a prompt, alongside the corresponding question and answer choices.

The resulting prompt is processed by a language model, specifically Qwen2.5-7B-Instruct [19], which was fine-tuned using LoRA (Low-Rank Adaptation) [20]. In our setup, LoRA injects low-rank trainable matrices into the attention projection layers (q_{proj}, k_{proj}, v_{proj}, and o_{proj}), allowing us to efficiently adapt the model with minimal parameter updates. We use a moderate rank of 16 and scaling factor of 32 to balance adaptation capacity and training stability. We fine-tune our model using GRPO and $|G| = 8$. Optimization is performed using a 8-bit quantized AdamW optimizer, with an initial learning rate of 3×10^{-5} and a weight decay coefficient of 0.01. Training is conducted over a single epoch, using a batch size of 8, on a Nvidia A100 GPU with 80 GB of memory.

Model	Part1	Part2	Part3	Total
Qwen2-Audio-7B	30.0	39.2	49.6	45.0
AudioFlamingo2	53.9	31.7	49.5	45.7
Gemini-2.0-Flash	42.0	46.3	56.6	52.5
Ours	**50.4**	**54.0**	**67.5**	**62.6** (+10.1%)
Ours (with dev)	58.5	60.0	68.3	65.4 (+12.9%)

Table 2: Accuracy comparison between baselines and our method across the three parts and overall on the development set. In green is the difference compared to Gemini-2.0-Flash, the best baseline.

As shown in Table 2, our method outperforms the baselines provided by the challenge. The submitted system was trained using the same procedure on both the training and development datasets. Its performance on the development set (a subset of the training set) is reported in the figure, indicated by the label "with dev."

3.5. Calibration results

The results (Figure 2) confirm that the BEATs model exhibited poor overall calibration. It tends to be underconfident for certain classes, such as `Male speech, man speaking` and `Female speech, woman speaking` and conversely overconfident for others, such as `Mechanisms`.

3.5.1. *Focus on the example of a specific class : Male speech, man speaking*

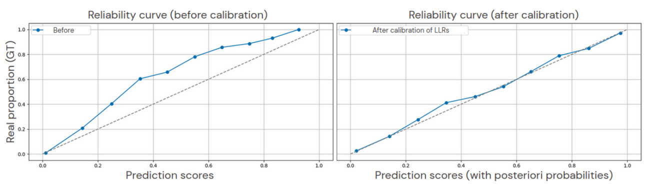

Figure 2: Experimental results of LLR calibration for the class `Male Speech, man speaking`. Distribution of log-likelihood ratios (LLRs) before (right) and after calibration (left), along with the corresponding reliability curves.

In the selected calibration example (figure 2) of the class `Male Speech, man speaking`, the reliability curve before calibration (bottom right) deviates significantly from the ideal diagonal, indicating poor model calibration. The predicted confidence scores are generally lower than the actual proportion of positive instances, reflecting under-confidence in the model's predictions. For instance, an average predicted score of 0.6 corresponds in practice to a class occurrence frequency above 0.8. After calibration (bottom left), the reliability curve aligns more closely with the diagonal, demonstrating a better match between predicted confidence and observed frequencies. This qualitative improvement is quantitatively supported by a significant decrease in the CLLR score, which drops from 0.653 to 0.266, indicating a notable reduction in calibration error.

These results demonstrate the effectiveness of LLR-based calibration combined with prior adjustment in enhancing both the interpretability and reliability of BEATs model predictions on the AudioSet dataset.

3.5.2. *Results on the 447 AudioSet classes*

Figure 3 represents the CLLR before and after calibration for each class in the AudioSet dataset. Each point represents a class, and its color shows the gain achieved through calibration (i.e., the CLLR before calibration minus the CLLR after calibration), with red tones indicating a substantial gain and blue tones indicating a small or negligible one. Most points lie below the diagonal, reflecting a systematic decrease in CLLR after calibration, which is an indication of an overall improvement in the quality of the prediction scores. Some classes that were initially very poorly calibrated (CLLR > 4) see their scores substantially reduced (to below 1), highlighting the ability of the calibration process to correct extreme cases. Moreover, the post-calibration CLLRs are predominantly below 0.5, suggesting good probabilistic adjustment. Finally, the post-calibration scores are much more tightly clustered, indicating a homogenization of performance across classes.

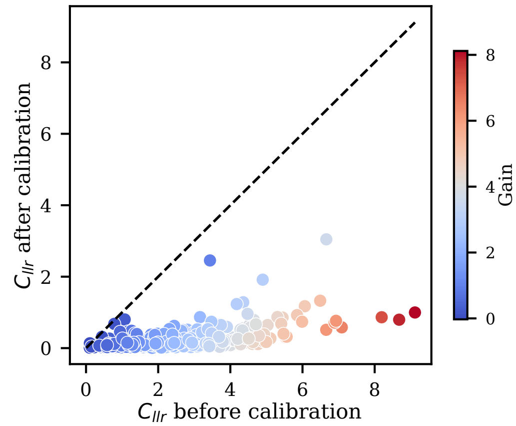

Figure 3: Class-wise CLLR Comparison on AudioSet – Before vs After Calibration

3.5.3. *Impact of calibration on our system*

Calibration is essential for selecting the optimal threshold to maximize the final evaluation metric. In general, calibration leads to improved performance on this challenge (see Table 3). A threshold of 0.1 was selected, as it yielded the highest score on the development set.

Model	Calibration	Threshold	Part 1	Part 2	Part 3
Qwen2.5-7B-Instruct + BEATS (frozen)	NO	0.05	27.2	46.5	45.1
Qwen2.5-7B-Instruct + BEATS (frozen)	NO	0.1	31.3	46.5	44.9
Qwen2.5-7B-Instruct + BEATS (frozen)	NO	0.2	31.3	41.5	43.2
Qwen2.5-7B-Instruct + BEATS (frozen)	YES	0.05	35.3	47.9	45.0
Qwen2.5-7B-Instruct + BEATS (frozen)	YES	0.1	35.7	51.2	44.1
Qwen2.5-7B-Instruct + BEATS (frozen)	YES	0.2	32.6	46.6	45.0
Qwen2.5-7B-Instruct + BEATS (fine-tuned)	NO	0.1	51.8	49.4	64.2
Qwen2.5-7B-Instruct + BEATS (fine-tuned)	YES	0.1	50.4	54.0	67.5

Table 3: Performance of our system without fine-tuning on the development set under different calibration and threshold settings

The last two lines compare the benefit of using calibration when the language model is fine-tuned. In Part 1, we observe a slight performance drop with calibration. This can be explained by the fact that the questions in this section primarily involve the classification and detection of marine mammals, which are not among the classes present in AudioSet. However, for the other two parts of the challenge, the questions involve audio samples from datasets whose ontologies are often closely aligned with that of AudioSet. This likely accounts for the performance gains observed when applying calibration.

3.6. What does GRPO truly encourage ?

Sometimes the correct answer is in the model's output, but extracting it in the expected format remains difficult. This raises the question of whether the gain due to fine-tuning with GRPO stem more from enforcing answer formatting than from actual improvements in reasoning about acoustic events.

To address this issue, we use the same base model (Qwen2.5-7B-Instruct), fine-tuned with GRPO under three different configurations (Table 4): (1) events are included neither during training nor inference, (2) events are included only at inference time, and (3) events are included during both training and inference.

When acoustic events are not included during training or inference, performance remains high Table. This suggests that some questions can be answered using the question alone or by eliminating wrong options. Adding events only at inference brings little improvement.

Training	Inference	Part 1	Part 2	Part 3
NO	NO	51.8	49.7	65.1
NO	YES	49.1	50.9	65.5
YES	YES	50.4	54.0	67.5

Table 4: Comparison of accuracy on the development set after 1 epoch of fine-tuning Qwen2.5-7B-Instruct. Training (respectively, Inference) indicates that events are included in the prompt during training (respectively, inference).

However, including them during training significantly boosts performance (except for Part 1; see Section 3.5.3). This shows that GRPO helps the model better use acoustic events to answer questions.

4. SUMMARY AND FUTURE WORK

We proposed a large language model for Task 5 of the DCASE 2025 Challenge. The model does not process audio directly but is fine-tuned using GRPO, which allows effective adaptation with limited data, from prompts that include timestamped acoustic events detected by a sound event detection model. We showed that calibrating the likelihood scores for each detected event class improves performance on the challenge. As future work, we suggest adapting the event priors based on the question being asked or on previously detected events.

5. REFERENCES

[1] Y. Chu, J. Xu, Q. Yang, H. Wei, X. Wei, Z. Guo, Y. Leng, Y. Lv, J. He, J. Lin, C. Zhou, and J. Zhou, "Qwen2-audio technical report," 2024. [Online]. Available: https://arxiv.org/abs/2407.10759

[2] S. Ghosh, Z. Kong, S. Kumar, S. Sakshi, J. Kim, W. Ping, R. Valle, D. Manocha, and B. Catanzaro, "Audio flamingo 2: An audio-language model with long-audio understanding and expert reasoning abilities," 2025. [Online]. Available: https://arxiv.org/abs/2503.03983

[3] Y. Gong, Y.-A. Chung, and J. Glass, "Ast: Audio spectrogram transformer," 2021. [Online]. Available: https://arxiv.org/abs/2104.01778

[4] J. Li, D. Li, S. Savarese, and S. Hoi, "Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models," 2023. [Online]. Available: https://arxiv.org/abs/2301.12597

[5] J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds, R. Ring, E. Rutherford, S. Cabi, T. Han, Z. Gong, S. Samangooei, M. Monteiro, J. Menick, S. Borgeaud, A. Brock, A. Nematzadeh, S. Sharifzadeh, M. Binkowski, R. Barreira, O. Vinyals, A. Zisserman, and K. Simonyan, "Flamingo: a visual language model for few-shot learning," 2022. [Online]. Available: https://arxiv.org/abs/2204.14198

[6] Y. Gong, H. Luo, A. H. Liu, L. Karlinsky, and J. Glass, "Listen, think, and understand," 2024. [Online]. Available: https://arxiv.org/abs/2305.10790

[7] F. Schmid, T. Morocutti, F. Foscarin, J. Schlüter, P. Primus, and G. Widmer, "Effective pre-training of audio transformers for sound event detection," in *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2025, pp. 1–5.

[8] A. Tanel Alumäe, "The TalTech systems for the VOiCES from a Distance Challenge," in *Interspeech (submitted)*, 2019.

[9] L. Ferrer and D. Ramos, "Evaluating posterior probabilities: Decision theory, proper scoring rules, and calibration," *arXiv preprint arXiv:2408.02841*, 2024.

[10] DeepSeek-AI, "Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning," 2025. [Online]. Available: https://arxiv.org/abs/2501.12948

[11] G. Li, J. Liu, H. Dinkel, Y. Niu, J. Zhang, and J. Luan, "Reinforcement learning outperforms supervised fine-tuning: A case study on audio question answering," 2025. [Online]. Available: https://arxiv.org/abs/2503.11197

[12] H. Zhou, X. Li, R. Wang, M. Cheng, T. Zhou, and C.-J. Hsieh, "R1-zero's "aha moment" in visual reasoning on a 2b non-sft model," 2025. [Online]. Available: https://arxiv.org/abs/2503.05132

[13] C.-H. H. Yang, S. Ghosh, Q. Wang, J. Kim, H. Hong, S. Kumar, G. Zhong, Z. Kong, S. Sakshi, V. Lokegaonkar, O. Nieto, R. Duraiswami, D. Manocha, G. Kim, J. Du, R. Valle, and B. Catanzaro, "Multi-domain audio question answering toward acoustic content reasoning in the dcase 2025 challenge," 2025. [Online]. Available: https://arxiv.org/abs/2505.07365

[14] J. Kim, H. Yun, S. H. Woo, C.-H. H. Yang, and G. Kim, "Wow-bench: Evaluating fine-grained acoustic perception in audio-language models via marine mammal vocalizations," 2025.

[15] S. Sakshi, U. Tyagi, S. Kumar, A. Seth, R. Selvakumar, O. Nieto, R. Duraiswami, S. Ghosh, and D. Manocha, "Mmau: A massive multi-task audio understanding and reasoning benchmark," in *ICLR*, 2025.

[16] S. Hershey, D. P. W. Ellis, E. Fonseca, A. Jansen, C. Liu, R. C. Moore, and M. Plakal, "The benefit of temporally-strong labels in audio event classification," 2021. [Online]. Available: https://arxiv.org/abs/2105.07031

[17] S. Chen, Y. Wu, C. Wang, S. Liu, D. Tompkins, Z. Chen, and F. Wei, "Beats: Audio pre-training with acoustic tokenizers," *arXiv preprint arXiv:2212.09058*, 2022.

[18] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2017, pp. 776–780.

[19] Qwen, "Qwen2.5 technical report," 2025. [Online]. Available: https://arxiv.org/abs/2412.15115

[20] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," 2021. [Online]. Available: https://arxiv.org/abs/2106.09685