LLMs as Judges in the Context of LALM Outputs: A Unified Plan for Open-Ended Audio Evaluation and Future Reward Modeling

Minhal Khan [redacted] **Lasha Koroshinadze** [redacted] **Chanakya Maddineni** [redacted]

1 Problem statement

The rapid advancement of Large Audio-Language Models (LALMs) has enabled sophisticated open-ended audio understanding, ranging from captioning to complex reasoning. However, evaluating these models remains a significant bottleneck. Traditional lexical metrics like BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), and METEOR (Banerjee and Lavie, 2005) correlate poorly with human judgment for open-ended tasks, as they rely on n-gram overlap and penalize semantically correct answers that differ in phrasing from the reference. Conversely, human evaluation, while the gold standard, is expensive, slow, and unscalable.

This project aims to develop a robust "LLM-as-a-Judge" system specifically for audio-grounded tasks. We propose fine-tuning a small, efficient language model (Qwen3-0.6B (Yang et al., 2025)) to predict human quality ratings (on a scale of 1-5) given an audio context, a question, and a candidate answer. Our research focuses on two primary questions: First, can small, generative decoder-only LLMs outperform strong encoder-only discriminative baselines, such as DeBERTa-based variants of BERTScore (Zhang et al., 2019)? Second, does incorporating synthetic Chain-of-Thought (CoT) (Wei et al., 2023) reasoning and data augmentation significantly improve alignment with human preference, thereby providing a scalable and reliable metric for the audio domain?

2 What we accomplished

In comparison to our proposal, the following goals were accomplished:

- **Generative model training:** We successfully fine-tuned a small LLM using both Supervised Fine-Tuning (SFT) and Group Relative Policy Optimization (GRPO). We also taught the model to generate custom CoT reasoning before assigning a final score, achieving good correlation with human raters.
- **Discriminative baseline:** We trained a DeBERTa-based regression model to act as a pure scoring judge, which outperformed standard metrics and established a strong upper bound for correlation performance.
- **Synthetic data engineering:** To address dataset scarcity, we expanded the training data fivefold through augmentation, paraphrasing rationales and candidate answers. Additionally, we generated reasoning traces using GPT-4o-mini.
- **Rigorous evaluation:** We implemented a full evaluation suite comparing lexical metrics (such as BLEU, ROUGE, METEOR), semantic metrics (BERTScore, SBERT (Reimers and Gurevych, 2019)), discriminative models and generative judges against human ground truth.

3 Related work

Several recent papers study LLMs-as-judges, specifically in audio captioning and speech evaluation. Closest to our work is CLAIR-A (Wu et al., 2025), which proposes a zero-shot, LLM-based metric for automated audio captioning that asks GPT-4o (and other LLMs) to output a numeric similarity score between a candidate caption and reference captions, along with a natural-language justification. CLAIR-A shows that this simple in-context evaluation strategy correlates more strongly with human preferences than both general-purpose metrics text overlap or embedding metrics such as BLEU, METEOR, BERTScore and audio-specific metrics such as

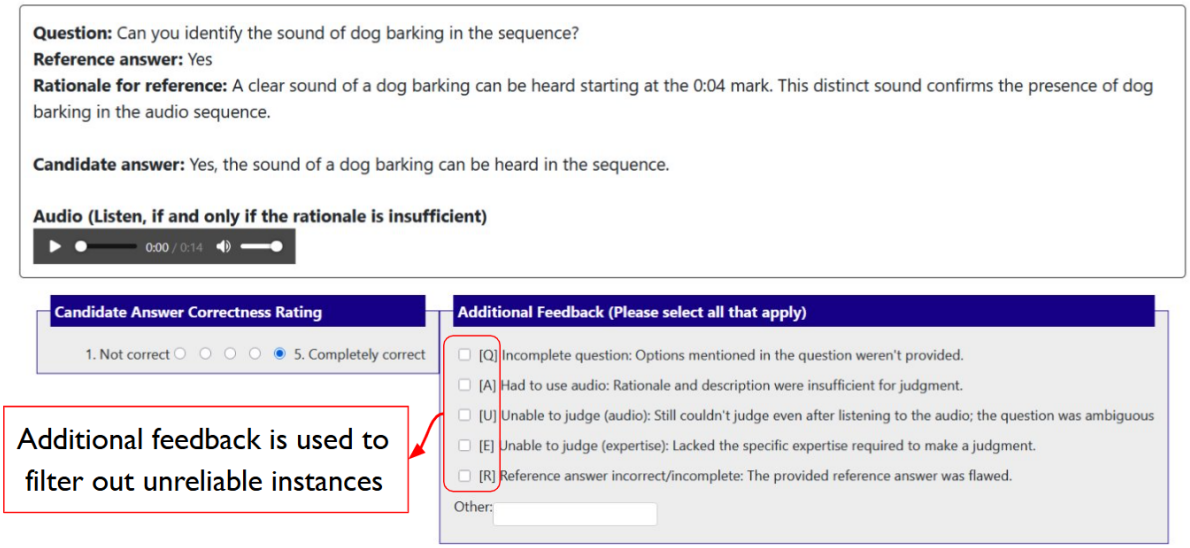


Figure 1: An example of how human annotations were collected using POTATO (Pei et al., 2023)

FENSE (Zhou et al., 2022), which combines semantic similarity with a fluency penalty, and ACES (Wijngaard et al., 2023), which compares semantic sound descriptor categories derived from captions. Whereas CLAIR-A focuses on audio captioning and uses zero-shot inference with large proprietary LLMs, we train a small open model as a supervised judge for open-ended audio question answering and reasoning, explicitly conditioning on questions, rationale, and transcripts.

Beyond captioning, AudioJudge (Manakul et al., 2025) investigates Large Audio Models (LAMs) as judges for speech quality and intelligibility, analyzing what works and fails when using LAM-based judges for tasks such as pronunciation, rate, and overall speech quality evaluation. The authors study how well LAM judges track human ratings, as well as their robustness to acoustic perturbations and biases. AudioJudge is complementary to our work: it concentrates on paralinguistic and perceptual speech evaluation, while we focus on semantic correctness and reasoning quality in audio-grounded QA, but both works highlight that task and domain-specific judge design matters for aligning with human preferences.

Furthermore, there exists a broader literature on using LLMs as general-purpose evaluators. Bavaresco et al. (2025) introduce JUDGEBENCH, a benchmark spanning multiple NLP datasets and LLMs, and show that while LLMs can match non-expert annotators on some graded tasks, their agreement with human judgments varies widely across properties, datasets, and models, leading the authors to recommend task-specific validation before deploying LLM judges as automatic evaluators. Complementary surveys on LLM-as-a-judge (Gu et al. (2025), Li et al. (2024)) systematically review this paradigm across functionality, methodology, applications, and limitations, emphasizing issues such as bias, robustness, and the need for meta-evaluation benchmarks to assess the reliability of judge models. Our work fits within this emerging landscape; rather than relying on a large, generic judge, we explicitly fine-tune a small, task-specific evaluator on human scores for audio-grounded QA, and we report standard correlations to make its alignment with human raters transparent.

4 Dataset and Preprocessing

We utilize two primary benchmarks for audio understanding: the MMAU (Sakshi et al., 2024) mini test set and MMAR (Ma et al., 2025).

4.1 Data Source

The core dataset for this project was created during the JSALT 2025 workshop. While it builds upon the existing MMAU and MMAR benchmarks, its distinct contribution is a new layer of human quality annotations (see Figure 1 for the annotation interface). The dataset contains a total of 3,613 evaluation items, consisting of 1,804 items from MMAU (covering 935 unique questions) and 1,809 items from MMAR (covering 937 unique questions). The total number of items exceeds the number of unique questions because

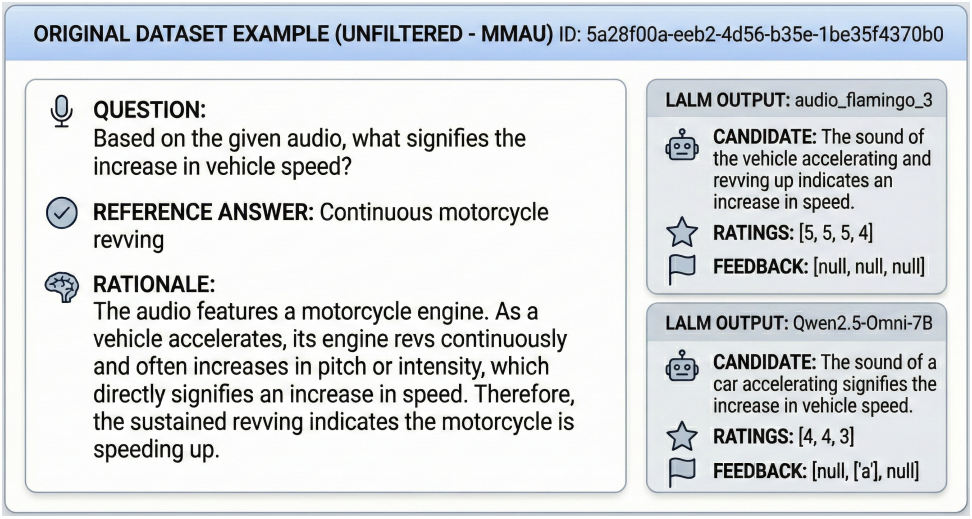


Figure 2: An example entry from the original dataset, showcasing the question, reference answer, rationale, and multiple LALM candidate outputs with human ratings.

up to 15 different LALMs were used to generate open-ended responses, including: SALMONN (Tang et al., 2024), Qwen2-Audio (Chu et al., 2024), and Audio Flamingo 3 (Goel et al., 2025). These responses were then rated by humans on a scale of 1 to 5. The majority of ratings (∼70%) were clustered at the extremes (1s and 5s).

In addition to numerical scores, annotators used feedback flags to identify specific issues, such as incomplete questions ([Q]), flawed references ([R]), or ambiguity preventing judgment ([U]), as detailed in Figure 1.

As shown in Figure 2, each data entry contains:

- **Question**: The prompt given to the model.
- **Reference Answer**: The ground truth answer.
- **Rationale**: A textual justification for the reference answer, generated by Gemini-2.5-flash.
- **Transcription**: A speech-to-text transcript generated by Whisper Large v3 (where applicable).
- **Candidate Answer**: The output from a specific LALM.
- **Human Ratings**: A list of scores provided by human annotators.

4.2 Preprocessing

The raw data was originally structured by question ID, with nested dictionaries for each model's output. We developed a cleaning pipeline to flatten this structure into a standard JSONL format: {`Question, Reference, Rationale, Transcript, Candidate, Mean Human Score`}. To ensure the reliability of our ground truth, we filtered the dataset by removing any evaluation instance where annotators selected *any* feedback flag (e.g., [Q], [A], [U], [E], [R]). This step eliminated ambiguous or flawed examples, resulting in a high-quality final dataset of approximately 3,400 items. Finally, we performed a random 90%/10% split to create our training and held-out test sets, ensuring that the evaluation results reported below reflect generalization to unseen questions.

4.3 Data Augmentation

Training a robust judge model on only ∼3,100 examples poses a significant risk of overfitting to specific sentence structures rather than learning semantic evaluation. To mitigate this and improve generalization, we implemented a synthetic data augmentation pipeline using GPT-4o-mini, expanding the training set by a factor of 5x. We used two primary strategies:

Linguistic Diversity. To prevent the model from overfitting to specific phrasings, we generated semantically equivalent but linguistically distinct versions of the rationales and candidate answers. For example, in an emotion detection task where the reference emotion is "Angry":

- **Original Rationale:** "The man's voice is loud and gruff, with a sharp, impatient tone. He repeatedly shouts 'Not now!'..."

- **Augmented Variation:** ”The man speaks in a loud, rough voice that carries an impatient quality. His repeated insistence of ’Not now!’ highlights his irritation...”

This ensures the model learns to recognize the underlying concept of the critique rather than memorizing keywords.

Negative Mining. We also created ”hard negatives” to teach the judge model to detect contradictions. By retaining a correct rationale (the evidence) but pairing it with a clearly incorrect candidate answer, we force the model to penalize answers that ignore the provided evidence.

- **Context (F1 Car):** The rationale describes ”a remarkably loud, piercing engine noise, typical of a high-speed racing car.”
- **Negative Pair:** The pipeline pairs this description with the candidate *”An alarm.”* (Target Score: 1.0).

This forces the model to spot when the audio evidence (rationale) explicitly contradicts the conclusion (candidate).

4.4 Synthetic Chain-of-Thought (CoT)

We hypothesized that treating evaluation as a simple regression task treats the judgment process as a ”black box,” limiting performance. To address this, we integrated synthetic CoT data to force the model to explicitly reason before assigning a score.

We used GPT-4o-mini to generate reasoning traces for every training example. The teacher model was provided with the full context and the ground truth human score, and was tasked with generating a step-by-step justification for that score. This transformed our training targets into the structured format: `<think> [Reasoning Trace] </think> [Score].`

For instance, in the ”F1 Car” example where a human rated a candidate 4.0 (instead of 5.0), the CoT teaches the model to articulate the specific deficiency:

> `<think>`
> The candidate answer states that ’the sound in the audio is most likely produced by a race car...’ This aligns well with the reference answer... However, the candidate answer is slightly less specific than the reference, as it does not explicitly mention ’F1 car’... This omission prevents the answer from being perfectly aligned... Nevertheless, the central idea remains intact... Given these factors, the score of 4.0 is justified...
> `</think>`
> **4.0**

This moves the model beyond simple regression to reasoned evaluation, improving explainability and alignment.

5 Baseline

We use a script to establish a baseline performance for our project. This calculates how well traditional, simpler metrics correlate with human judgements and serves as a point of comparison for our LLM-as-a-judge model.

Two main classes of metrics are evaluated:

- **Lexical metrics**: BLEU, ROUGE, and METEOR. These metrics rely on exact word overlaps between the candidate and reference answers. While they are computationally cheap, they often fail to capture semantic equivalences, especially in open-ended tasks where the correct answer can be phrased in multiple ways.
- **Semantic metrics**: We compute BERTScore and SBERT (Sentence BERT) cosine similarity. These metrics use pre-trained contextual embeddings to measure the similarity in meaning rather than just surface-level word matching.

To quantify the effectiveness of these baselines, we initially calculated the mean performance of each metric on our training dataset. We later extended our approach to include bucketed analysis, grouping human ratings into five quality tiers (”very poor” to ”excellent”) to observe if the baseline metrics monotonically increase with human quality judgements.

Rating	N	BLEU	ROUGE	MET	BERT	S-BERT
1 (V. Poor)	1740	.002	.039	.047	.831	.245
2 (Poor)	328	.002	.071	.089	.840	.324
3 (Neutral)	214	.000	.078	.086	.837	.336
4 (Good)	284	.004	.108	.137	.842	.399
5 (Excel.)	830	.018	.299	.298	.869	.534

Table 1: Mean metric scores stratified by human rating buckets. Each cell represents the average value of that metric for all examples assigned that specific human rating.

We refined our analysis by calculating the Pearson (r) and Spearman (ρ) correlations between

the semantic metrics and the ground-truth human ratings on the training set. BERTScore showed weak correlation ($r = 0.21$, $\rho = 0.24$), while S-BERT performed moderately better ($r = 0.51$, $\rho = 0.47$). These relatively low correlations, particularly for BERTScore, indicate the limitations of existing automated metrics for audio-grounded reasoning and help justify the need for a trained judge model.

5.1 Supervised Discriminative Baseline: DeBERTa scorer

To determine whether the complexity of a generative LLM is necessary for accurate scoring, we trained a discriminative baseline based on the DeBERTa-v3 architecture (He et al., 2023). Unlike the generative Qwen models, this is an encoder-only architecture fine-tuned to output a single scalar score based on the input.

Architecture: We utilized `microsoft/deberta-v3-base` as our backbone. The input sequence x is constructed by concatenating the relevant fields into a single string with explicit separators:

```
Question: q
Reference: a*
Rationale: r
Transcript: t
Candidate: â
```

The input includes the question, reference answer, rationale, transcript (where available), and the candidate answer. DeBERTa-v3 utilizes a disentangled attention mechanism, where content and relative position are encoded separately, allowing it to better model the semantic relationship between the reference and candidate answers over long sequences compared to standard BERT models.

Learnable Pooling Mechanism. Instead of relying solely on the standard `[CLS]` token embedding for classification, we implemented a custom learnable pooling head to aggregate information from the entire sequence. We initialize a learnable query vector $\mathbf{q}_{pool} \in \mathcal{R}^d$ (where $d = 768$). We then apply a Multi-Head Attention (MHA) layer where $\mathbf{q}_{pool}$ serves as the query, and the encoder's last hidden states $\mathbf{H} \in \mathbb{R}^{L \times d}$ serve as both keys and values:

$$\mathbf{h}_{pooled} = \text{MHA}(\text{query} = \mathbf{q}_{pool}, \text{key} = \mathbf{H}, \text{value} = \mathbf{H})$$

This mechanism allows the model to dynamically attend to the most relevant tokens in the sequence (e.g., specific discrepancies between the candidate and reference) regardless of their position, compressing the sequence into a single context-aware vector $\mathbf{h}_{pooled}$.

Regression Head and Objective. The pooled representation is passed through a two-layer Multi-Layer Perceptron (MLP) with ReLU activation and dropout ($p = 0.1$) to project it to a scalar logit z. A sigmoid function $\sigma(\cdot)$ maps this logit to the range $(0, 1)$, representing the normalized score:

$$\hat{y} = \sigma(\text{MLP}(\mathbf{h}_{pooled}))$$

The model was trained to minimize the Mean Squared Error (MSE) between the predicted score $\hat{y}$ and the normalized human ground truth score $y \in [0, 1]$ (derived from the original 1–5 Likert scale):

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} ||\hat{y}_i - y_i||^2$$

We trained for 20 epochs with a batch size of 16 and a learning rate of 2×10^{-5} using the AdamW optimizer. To address the label imbalance (predominance of 1s and 5s), we employed stratified sampling during training to ensure uniform exposure to all score levels.

6 Multi-stage Training

We adopted a two-stage training paradigm, moving from supervised imitation to reinforcement learning. This approach allows the model to first learn the structural requirements of the task (SFT) before optimizing for scoring accuracy and reasoning quality (GRPO).

6.1 Stage 1: Supervised Fine-Tuning (SFT)

The primary goal of the SFT stage was to bootstrap the model's ability to adhere to the required output formats. We used the `Qwen/Qwen3-0.6B` model as our base. It is

worth noting that Qwen3 natively supports Chain-of-Thought (CoT) reasoning; enabling this behavior typically requires setting a specific flag in the chat template or appending the token `/think` at the end of the prompt.

We explored two distinct SFT strategies:

1. **SFT with CoT:** The model is trained to generate a coherent reasoning trace enclosed in `<think>` tags before outputting the final scalar score.

2. **SFT No-Think (Direct Scoring):** The model is trained to simply output a single scalar number (1-5) with no prior reasoning, serving as a baseline for the effectiveness of the reasoning process.



Figure 3: Training loss for the SFT model with Chain-of-Thought (CoT) reasoning.

For both strategies, we utilized LoRA (Low-Rank Adaptation) rather than full fine-tuning. We targeted all linear layers in the transformer block (`q_proj`, `k_proj`, `v_proj`, `o_proj`, `gate_proj`, `up_proj`, `down_proj`) with a rank $r = 16$ and alpha $\alpha = 32$.

The models were trained on the original training dataset ($N \approx 3,100$ examples) using standard causal language modeling cross-entropy loss. We trained for 8 epochs with a learning rate of 10^{-4}, a cosine scheduler, and an effective batch size of 32. To accommodate the lengthy audio transcripts and reasoning traces, we set the maximum sequence length to 8192 tokens.

Training efficiency varied significantly between the two approaches due to the output length. The **SFT with CoT** model required approximately **30 minutes** to train, whereas the **SFT No-Think** model completed training in just **10 minutes**.

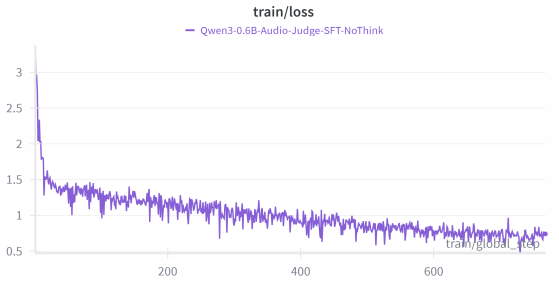


Figure 4: Training loss for the SFT model trained on direct scoring (No-Think).



Figure 5: Comparative validation loss between the CoT and No-Think models. The No-Think model exhibits signs of overfitting (rising validation loss) much earlier than the CoT model.

6.2 Stage 2: Group Relative Policy Optimization (GRPO)

While SFT teaches the model *how* to speak, it does not necessarily teach it *what* constitutes a good judgment. To align the model's scoring behavior with human intuition, we proceeded with Group Relative Policy Optimization (GRPO) on the full augmented dataset ($N \approx 15,500$). This stage was computationally intensive, requiring approximately **6.5 hours**.

GRPO eliminates the need for a separate value function model, making it highly efficient for our compute constraints. For every training prompt q, the model samples a group of outputs $\{o_1, o_2, ..., o_G\}$ (where $G = 8$). The temperature is set to 0.7 and effective batch size to 8. The optimization objective maximizes the likelihood of outputs that receive high rewards relative to the group average. We designed two specific reward functions:

1. **Format Reward (Max 0.6):** A structural reward composed of three incremental components: matching the CoT tag pattern (+0.2), ensuring minimum reasoning length (+0.2), and producing a valid parseable score (+0.2).

This ensures the model maintains the required output structure.

2. **Accuracy Reward (Max 1.0):** A distance-based reward derived from the ground truth human score y_{human}:

$$R_{acc} = 1.0 - \frac{|y_{pred} - y_{human}|}{4.0}$$

This function provides a continuous signal, offering partial credit for near-misses while linearly penalizing deviations, scaling from 0.0 (max error) to 1.0 (perfect match).



Figure 6: Training reward during GRPO. The reward consistently fluctuates between 0.6 and 1.6, indicating that the prior SFT stage successfully stabilized the format adherence (the 0.6 baseline). The variance primarily reflects the model's exploration in optimizing the accuracy component.

We utilized a significantly lower learning rate of 10^{-6} for this stage to prevent catastrophic forgetting of the SFT knowledge. We set the KL-divergence coefficient $\beta = 0.001$. As shown in Figure 7, the KL term steadily increases, indicating that the policy is successfully exploring and deviating from the reference model to maximize rewards, even though recent research suggests the KL term might be optional in GRPO frameworks.

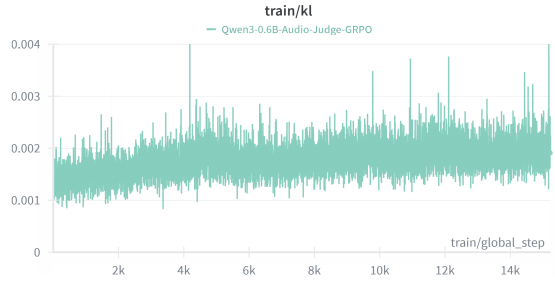


Figure 7: KL Divergence over time. The steady increase indicates the model is learning new behaviors and deviating from the initial SFT policy.

6.3 Hardware and Implementation

All training experiments were conducted on a single NVIDIA H200 GPU.

We used the `trl` library for the training loop and integrated `vLLM` for generation rollouts during the GRPO stage. This hybrid setup allowed for extremely fast inference during RL training.

7 Evaluation metrics

To rigorously assess the performance of our LLM-as-a-Judge, we compare its predicted scores against the ground truth human ratings using the following metrics:

- **Pearson Correlation (r):** Measures the linear correlation between the predicted scores ($\hat{y}$) and human ratings (y).

$$r = \frac{\sum(\hat{y}_i - \bar{\hat{y}})(y_i - \bar{y})}{\sqrt{\sum(\hat{y}_i - \bar{\hat{y}})^2 \sum(y_i - \bar{y})^2}}$$

- **Spearman's Rank Correlation (ρ):** Measures the monotonic relationship by assessing the correlation of rank variables, where d_i is the difference between the two ranks of each observation:

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)}$$

- **Mean Absolute Error (MAE)**: Calculates the average absolute difference between the predicted scores and the ground truth human ratings. This provides a direct measure of the prediction error magnitude on the 1-5 scale.

- **Accuracy**: Measures the percentage of instances where the model's predicted score (rounded to the nearest integer) exactly matches the rounded human rating. This treats the evaluation as a classification problem to assess precise alignment.

8 Results

We evaluated the Qwen model against unsupervised metrics and a supervised baseline, comparing performance before and after fine-tuning. The evaluations were performed on the held-out original test set ($N = 338$), the results of which are summarized in the table below:

The zero-shot base Qwen model with no CoT reasoning failed completely, achieving near zero

Model / Method	Pearson (r)	Spearman (ρ)	MAE	Acc
Baselines				
Base DeBERTa (BERTScore)	0.210	0.239	-	-
Base (S-BERT)	0.506	0.472	-	-
Fine-tuned (DeBERTa)	**0.867**	**0.823**	**0.4727**	**70.18%**
Qwen-Base				
Direct Scoring	-0.008	-0.038	2.012	15.6%
CoT Reasoning	0.440	0.399	1.377	44.6%
Qwen-Fine-Tuned				
SFT (Direct Scoring)	0.834	0.802	0.514	66.7%
SFT (CoT Reasoning)	0.823	0.787	0.526	66.9%
GRPO (CoT Reasoning)	0.824	0.798	0.525	67.8%

Table 2: Performance comparison. *Direct Scoring* indicates the model output a score immediately; *CoT* indicates the model generated reasoning first.

Pearson correlation and an accuracy of 15.6%, confirming that a small 0.6B parameter model has no inherent capability to judge audio-grounded QA pairs without specific instruction tuning. Moreover, this also calls attention to the non-triviality of the task and its inability to be solved by the base model's pre-training knowledge alone. Prompting the model to use CoT reasoning yielded a significant improvement, raising the Pearson correlation to $r = 0.44$. However, this is still notably lower than the S-BERT baseline's $r = 0.506$. This indicates that while reasoning helps, the base model lacks the domain-specific knowledge required to outperform simple embedding metrics.

The supervised DeBERTa baseline significantly outperforms all unsupervised metrics and Qwen models across all categories, demonstrating that the task is learnable and that a specialized encoder model can effectively rank answer quality. The bidirectional attention mechanism of DeBERTa allows it to cross-reference the candidate answer against the rationale and reference simultaneously, making it effective at detecting semantic discrepancies. As mentioned earlier in the report, however, the downside is that it remains a "black box" that cannot explain its decisions to a human user.

Our fine-tuned generative models bridge this gap between performance and interpretability. The SFT (Direct Scoring) model achieved a high Pearson $r = 0.834$ and a relatively low MAE (0.514), suggesting that the model internalized the scoring rubric. Most importantly, the GRPO model (with CoT) achieved a high classification accuracy (67.8%). While its linear correlation of $r = 0.824$ is slightly lower than the pure regression baseline set by DeBERTa, it offers the advantage of providing reasoning traces while maintaining respectable accuracy over the task of score prediction.

The success of the GRPO model suggests that reinforcement learning optimized its ability to serve as an accurate and interpretable judge to some extent, but its performance relative to the discriminative baseline highlights that there is still room for improvement. We isolate and analyze the success and failure cases from our models in the next section to better understand which of their aspects need to be addressed in detail.

9 Error Analysis

9.1 Analysis of DeBERTa-based judge

To understand the behavior of our strongest baseline, we analyzed instances where the fine-tuned DeBERTa model's predicted score had the largest deviation (Mean Absolute Error) from the human ground truth. This analysis revealed two distinct and informative patterns.

In cases of significant **underestimation**, the model was much harsher than human raters. This typically occurred when a candidate answer was semantically correct but failed to use a specific keyword present in the reference. For example, when the reference answer was "Chiropractic," a candidate describing the scene as "The woman is engaging in conversation with a man who is explaining his pain, and afterward, she checks him." was rated 4.0 by a human but only 1.8 by the model. This suggests the model over-weights the presence of key terms learned during training and is less robust to valid paraphrasing. This suggests the model over-weights the presence of key terms learned during training and is less robust to valid paraphrasing.

9.2 Analysis of Qwen3-based judge

To understand where and how the models falter when rating answers, we conducted a qualitative analysis of specific examples where the model's score differed from the human rating by at least 2 points. This section highlights characteristic behaviors: both failures and successes, across the different training stages.

The untrained **base model (No-Think)** frequently struggles with basic quality assessment. As shown in Figure 8, the model sometimes conflates the length or complexity of a candidate answer with its correctness, leading to significant overestimations of factually wrong responses.

When forced to use reasoning without fine-tuning, the **base model (CoT)** often produces fractured or contradictory logic. Figure 9 illustrates a

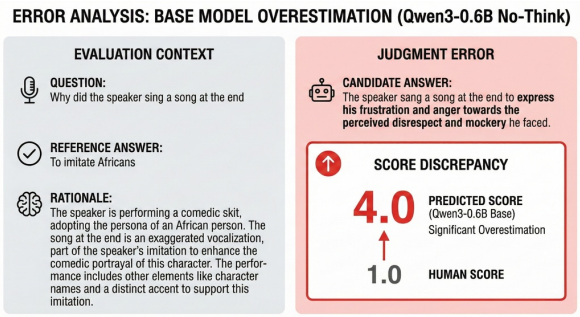


Figure 8: The Base model (No-Think) often hallucinates quality in verbose but incorrect answers, overestimating them significantly.

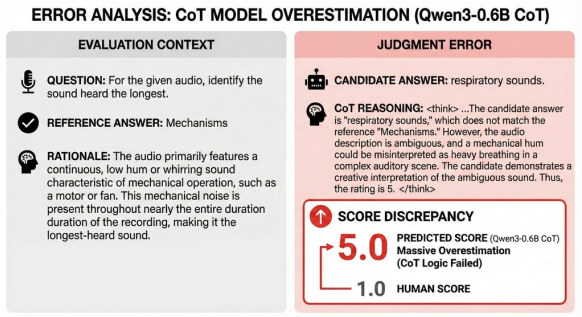


Figure 9: When forced to use CoT, the Base model's reasoning often breaks down, leading to logical contradictions and incorrect scores.

case where the model's reasoning correctly identifies the candidate as incorrect but then inexplicably assigns a high score, highlighting a disconnect between its reasoning and scoring mechanisms.

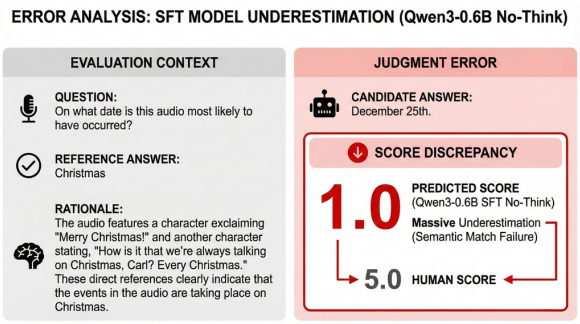


Figure 10: The SFT model (No-Think) shows improved accuracy but can still be semantically rigid, failing to recognize synonyms or alternative phrasings.

Supervised Fine-Tuning significantly improves performance, but the **SFT model (No-Think)** still exhibits semantic rigidity. Figure 10 shows a case of massive underestimation where the model fails to recognize that the candidate answer is a semantic equivalent of the reference, demonstrating a lack of robustness to surface-level variations.

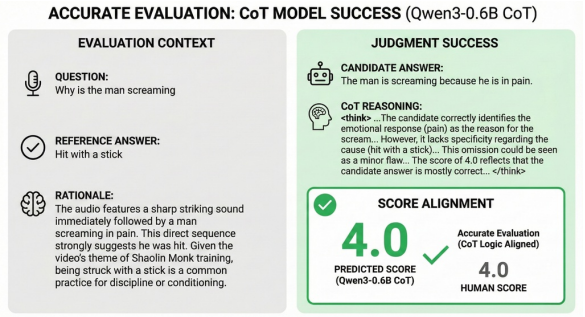


Figure 11: The SFT model (CoT) demonstrates the ability to align its reasoning with its score, accurately identifying partially correct answers.

With Chain-of-Thought enabled, the **SFT model (CoT)** is better able to articulate the nuances of its judgment. Figure 11 showcases a success story where the model's reasoning correctly identifies a candidate answer as mostly correct but lacking specific details, aligning perfectly with the human rater's score.

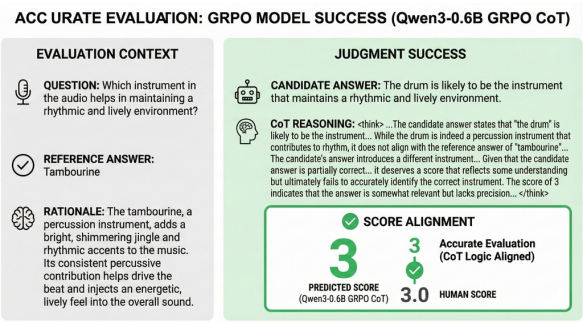


Figure 12: The final GRPO model (CoT) shows the highest alignment with human judgment, correctly identifying the core concept even when the specific term differs from the reference.

Our final **GRPO model (CoT)** demonstrates the highest level of sophistication. As seen in Figure 12, the model is able to look beyond exact word matching, correctly identifying that the candidate's mention of a "drum" captures the core concept of a percussion instrument maintaining rhythm, even if the reference specified a "tambourine." Its reasoning clearly justifies this nuanced judgment, aligning perfectly with the human score.

From these qualitative examples, it is clear that while GRPO and CoT significantly improve the model's judgment capability, fundamental challenges in semantic understanding persist.

10 Conclusion

Our project successfully demonstrates that small, specialized language models can serve as effective judges for open-ended audio reasoning tasks, provided they are trained with the right methodology. Our experiments revealed that while lexical methods and base metrics like BERTScore are insufficient for this domain, a fine-tuned Qwen3-0.6B model can achieve high alignment with human ratings ($r > 0.82$) while offering the crucial advantage of interpretability.

One surprising result was the effectiveness of the fine-tuned DeBERTa, which outperformed our generative models with regards to pure linear correlation. This indicates that decoder-only LLMs are not necessarily superior for all evaluation tasks; rather, compact, task-specific discriminative models remain highly competitive when the objective is constrained to pure scoring. Our GRPO model received a respectable accuracy score (67.8%), supporting the notion that reinforcement learning helps the model internalize the specific decision boundaries of the 1-5 rating scale better than simple regression.

Despite these successes, some challenges still remain. Our error analysis uncovered that the judge is still susceptible to certain biases, such as favoring large, complex-sounding answers when they are nonsensical, or failing to differentiate between synonyms. These issues indicate that while the model has learned the *form* of a good critique, it has yet to master the actual *substance* of semantic equivalence.

Continuing this work would involve following two potential main directions. First, we would have to refine our reinforcement learning methods (by incorporating DPO, for instance), to address the failure modes outlined in the error analysis section earlier in order to improve model accuracy. Second, we could explore a hybrid architecture where a lightweight encoder (such as DeBERTa) predicts the score and a generative model (such as Qwen) provides the rationale, combining the best of both worlds to achieve maximum correlation and interpretability.

11 AI Disclosure

The conceptualization, outline and direction of the project were purely handled by the team. We mainly used Gemini 3 Pro and ChatGPT while writing scripts, proofreading written text, and to double-check our methodologies at each stage of the project. Certain images in this report were also generated using Gemini to provide easy visualizations and facilitate easier skimming of complex concepts. Although the code generated by these models had solid foundations, some required manual intervention in the form of optimizations and bug-fixes, such as with data parsing or training scripts. All model outputs were ultimately vetted by each of the team members, and, to the best of our knowledge, no outputs were ever clearly wrong or irrelevant to our needs.

References

Banerjee, S. and Lavie, A. (2005). Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Bavaresco, A., Bernardi, R., Bertolazzi, L., Elliott, D., Fernández, R., Gatt, A., Ghaleb, E., Giulianelli, M., Hanna, M., Koller, A., Martins, A. F. T., Mondorf, P., Neplenbroek, V., Pezzelle, S., Plank, B., Schlangen, D., Suglia, A., Surikuchi, A. K., Takmaz, E., and Testoni, A. (2025). Llms instead of human judges? a large scale empirical study across 20 nlp evaluation tasks.

Chu, Y., Xu, J., Yang, Q., Wei, H., Wei, X., Guo, Z., Leng, Y., Lv, Y., He, J., Lin, J., Zhou, C., and Zhou, J. (2024). Qwen2-audio technical report.

Goel, A., Ghosh, S., Kim, J., Kumar, S., Kong, Z., gil Lee, S., Yang, C.-H. H., Duraiswami, R., Manocha, D., Valle, R., and Catanzaro, B. (2025). Audio flamingo 3: Advancing audio intelligence with fully open large audio language models.

Gu, J., Jiang, X., Shi, Z., Tan, H., Zhai, X., Xu, C., Li, W., Shen, Y., Ma, S., Liu, H., Wang, S., Zhang, K., Wang, Y., Gao, W., Ni, L., and Guo, J. (2025). A survey on llm-as-a-judge.

He, P., Gao, J., and Chen, W. (2023). Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing.

Li, H., Dong, Q., Chen, J., Su, H., Zhou, Y., Ai, Q., Ye, Z., and Liu, Y. (2024). Llms-as-judges: A comprehensive survey on llm-based evaluation methods.

Lin, C.-Y. (2004). Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Ma, Z., Ma, Y., Zhu, Y., Yang, C., Chao, Y.-W., Xu, R., Chen, W., Chen, Y., Chen, Z., Cong, J., Li, K., Li, K., Li, S., Li, X., Li, X., Lian, Z., Liang, Y., Liu, M., Niu, Z., Wang, T., Wang, Y., Wang, Y., Wu, Y., Yang, G., Yu, J., Yuan, R., Zheng, Z., Zhou, Z., Zhu, H., Xue, W., Benetos, E., Yu, K., Chng, E.-S., and Chen, X. (2025). Mmar: A challenging benchmark for deep reasoning in speech, audio, music, and their mix.

Manakul, P., Gan, W. H., Ryan, M. J., Khan, A. S., Sirichotedumrong, W., Pipatanakul, K., Held, W., and Yang, D. (2025). Audiojudge: Understanding what works in large audio model based speech evaluation.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Pei, J., Ananthasubramaniam, A., Wang, X., Zhou, N., Sargent, J., Dedeloudis, A., and Jurgens, D. (2023). Potato: The portable text annotation tool.

Reimers, N. and Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks.

Sakshi, S., Tyagi, U., Kumar, S., Seth, A., Selvakumar, R., Nieto, O., Duraiswami, R., Ghosh, S., and Manocha, D. (2024). Mmau: A massive multi-task audio understanding and reasoning benchmark.

Tang, C., Yu, W., Sun, G., Chen, X., Tan, T., Li, W., Lu, L., Ma, Z., and Zhang, C. (2024). Salmonn: Towards generic hearing abilities for large language models.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q., and Zhou, D. (2023). Chain-of-thought prompting elicits reasoning in large language models.

Wijngaard, G., Formisano, E., Giordano, B. L., and Dumontier, M. (2023). Aces: Evaluating automated audio captioning models on the semantics of sounds. In *2023 31st European Signal Processing Conference (EUSIPCO)*, pages 770–774. IEEE.

Wu, T.-H., Gonzalez, J. E., Darrell, T., and Chan, D. M. (2025). Clair-a: Leveraging large language models to judge audio captions.

Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., Zheng, C., Liu, D., Zhou, F., Huang, F., Hu, F., Ge, H., Wei, H., Lin, H., Tang, J., Yang, J., Tu, J., Zhang, J., Yang, J., Yang, J., Zhou, J., Zhou, J., Lin, J., Dang, K., Bao, K., Yang, K., Yu, L., Deng, L., Li, M., Xue, M., Li, M., Zhang, P., Wang, P., Zhu, Q., Men, R., Gao, R., Liu, S., Luo, S., Li, T., Tang, T., Yin, W., Ren, X., Wang, X., Zhang, X., Ren, X., Fan, Y., Su, Y., Zhang, Y., Zhang, Y., Wan, Y., Liu, Y., Wang, Z., Cui, Z., Zhang, Z., Zhou, Z., and Qiu, Z. (2025). Qwen3 technical report.

Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., and Artzi, Y. (2019). Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Zhou, Z., Zhang, Z., Xu, X., Xie, Z., Wu, M., and Zhu, K. Q. (2022). Can audio captions be evaluated with image caption metrics? In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 981–985. IEEE.