WORDS AT PLAY: BENCHMARKING AUDIO PUN UNDERSTANDING IN LARGE AUDIO-LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Puns represent a typical linguistic phenomenon that exploits polysemy and phonetic ambiguity to generate humour, posing unique challenges to the understanding of natural languages. Compared with text and images, spoken language plays a more central role in human communication. However, datasets and systematic resources for audio puns remain scarce, leaving this important modality largely underexplored. In this paper, we present APUN-Bench, the first benchmark dedicated to evaluating large audio language models (LALMs) on audio pun understanding. Our benchmark contains 4,447 audio samples annotated across three stages: pun recognition, pun word location and pun meaning inference. We conduct a deep analysis of APUN-Bench by systematically evaluating 10 state-of-the-art LALMs, uncovering substantial performance gaps in recognizing, localizing, and interpreting audio puns. Motivated by our analyses of positional biases in pun localization and error patterns in pun meaning inference, we further propose a prompt-based strategy and a fine-tuning framework to improve audio pun understanding. Our findings provide actionable insights toward advancing humour-aware audio intelligence.

1 INTRODUCTION

Puns are a common linguistic phenomenon that exploit lexical ambiguity to create humour through the interaction of phonetic, orthographic, and semantic features (Partington, 2009). As an important form of humorous expression, they reflect human creativity and communicative flexibility (Aleksandrova, 2022), and thus being widely applied to education, daily communication, and entertainment (Lems, 2011; Heidari-Shahreza, 2024). Nevertheless, understanding puns remains challenging because it must integrate multiple linguistic cues with contextual knowledge and reason over competing interpretations to identify the intended humorous meaning. Consequently, pun understanding has attracted increasing attention as a rigorous test of natural language understanding and reasoning capabilities in artificial intelligence (Morris et al., 2024).

Audio puns highlight the central role of speech-related cues and pose unique challenges for pun understanding (Karpenko, 2017; Palmann & Miller, 2025). As the primary medium of human communication, speech conveys linguistic content together with rich acoustic information, such as pronunciation, prosody, and phonetic variation. Audio puns leverage these speech-specific cues to activate multiple plausible interpretations through phonetic ambiguity, pronunciation similarity, and acoustic variability (Benzeghiba et al., 2007). Accordingly, understanding audio puns requires models not only to recognize spoken words but also to resolve competing meanings by integrating acoustic evidence with contextual information. Existing studies suggest that this difficulty arises primarily from speech-level ambiguity and contextual reinterpretation rather than from vocabulary rarity (Zheng & Wang, 2022). These characteristics make audio puns a valuable testbed for investigating human language and cognition, while also posing important challenges for audio understanding and multimodal AI.

Benchmarking audio puns is essential for uncovering the limitations of current LALMs and advancing the higher-level audio understanding capabilities of multimodal AI systems. Existing pun benchmarks primarily focus on textual and visual modalities (Ouyang et al., 2024; Xu et al., 2024; Zhang et al., 2024), thereby overlooking the speech-specific mechanisms that are often central to pun interpretation like phonetic similarity. Conversely, existing audio benchmarks mainly evaluate

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

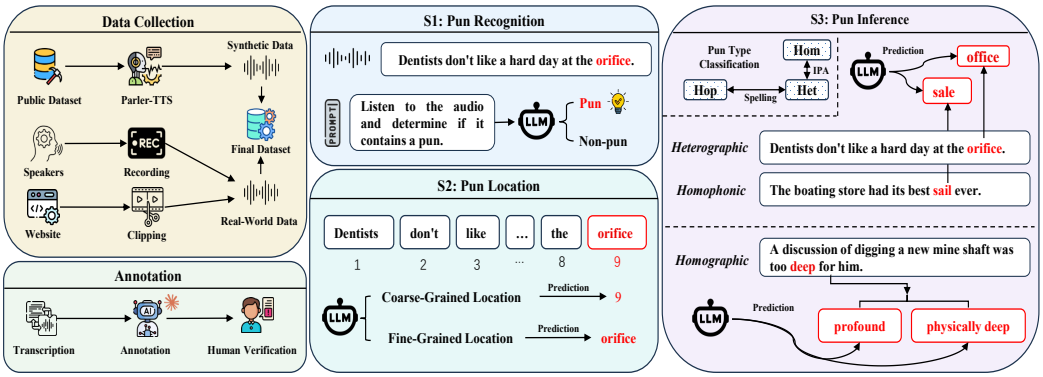

Figure 1: The Overview of APUN-Benchmark, covering data construction, human annotation and three evaluation stages: (S1) Pun Recognition, detecting the presence of an audio pun; (S2) Pun Location, identifying the specific pun word in the audio; and (S3) Pun Inference, inferring its dual meanings.

fundamental capabilities such as speech question answering and general multimodal comprehension (Yang et al., 2024; Sakshi et al., 2024; Wang et al., 2024; Gao et al., 2024; Zhang et al., 2025; Wang et al., 2025), but rarely include sophisticated linguistic phenomena such as puns that require models to reconcile acoustic cues with contextual and pragmatic information. As a result, benchmarking audio puns extends model assessment beyond basic speech perception and semantic comprehension toward the interpretation of ambiguity, implicit intent, and context-dependent humour.

This paper introduces APUN-Bench, the first benchmark for audio pun understanding, comprising 4,447 fully annotated audio samples across three key stages: pun recognition, pun word location, and pun meaning inference. Specifically, the first two stages evaluate whether a model can detect the presence of a pun and accurately identify the punning word, while the third assesses its ability to infer the intended secondary meanings of heterographic, homographic, and homophonic puns. APUN-Bench consists of both synthetic and real-world speech data. The synthetic portion is created by converting existing text-based pun corpora into speech, whereas the real-world portion is collected from publicly available audio resources to capture naturally occurring spoken puns. All audio samples and annotations are carefully verified to ensure high quality and annotation consistency. Extensive experiments on APUN-Bench reveal that current LALMs still exhibit substantial limitations in audio pun understanding. To address these challenges, we further propose two targeted improvement methods: a prompt-based approach for pun word location and a training-based framework for pun meaning inference. Experimental results demonstrate that both methods consistently improve model performance on audio pun understanding. Our main contributions are summarized as follows:

- We propose APUN-Bench, the first benchmark for audio pun understanding, including 4,447-sample dataset combining synthetic and real-world speech with semi-automatic annotation and human verification.
- We introduce a pipeline for audio pun understanding and conduct a systematic evaluation of 10 LALMs and 4 cascaded systems, providing comprehensive analyses of their strengths and limitations.
- We conduct an in-depth analysis of model performance and propose both prompt-level and model-level approaches to improve audio pun understanding, achieving an average accuracy improvement of 8.86%.

2 Related Work

2.1 Benchmarks for LALMs

General-purpose resources such as Audiobench Wang et al. (2024) and Airbench Yang et al. (2024) focus on recognition and generative comprehension across diverse acoustic conditions. More recent

Dataset	Heterographic	Homophonic	Homographic	Negative	Overall
Synthetic Dataset	794	304	1298	1,152	3,548
- Segment Clip (s)	4.18	4.36	3.88	3.47	3.97
- Sentence Length	12.21	13.89	11.68	8.50	11.57
Real-World Dataset	223	73	287	316	899
- Segment Clip (s)	6.36	6.16	5.34	6.73	6.14
- Sentence Length	11.12	11.40	11.55	10.88	11.23

Table 1: APUN-Bench data statistics, including average audio clip duration and sentence length.

work extends this scope: Sakshi et al. (2024) and Wang et al. (2025) assess multi-step spoken understanding and reasoning, while Zhang et al. (2025) highlights challenges in natural conversational settings. Other studies investigate specific dimensions, including cross-modal hallucination (Sung-Bin et al., 2024), dialogue ambiguity (Gao et al., 2024), and reasoning strategies for structured inference (Ma et al., 2025; Xie et al., 2025).

2.2 Pun Studies

Existing research on pun understanding has largely concentrated on recognition (Miller et al., 2017; Diao et al., 2018; Zhou et al., 2020; Jentzsch & Kersting, 2023; Thaniserikaran & Harikrishnan, 2026), location (Zou & Lu, 2019), and explanation (Sun et al., 2022a), but has also begun to explore multimodal extensions such as visual puns (Zhang et al., 2024). Beyond understanding, some researchers have further investigated automatic pun generation (Yu et al., 2018; Luo et al., 2019; Sun et al., 2022b; Su et al., 2025) and systematic evaluation of the ability of large-scale language models to understand puns (Xu et al., 2024). However, as a central form of human communication, speech remains largely unexplored in pun research, and there is an urgent need for systematic studies.

3 APUN-Bench

Our goal is to systematically evaluate the ability of LALMs to understand audio puns. Since existing benchmarks focus primarily on textual puns, we introduce APUN-Bench to assess pun detection and semantic inference in speech. Figure 1 presents an overview of the benchmark, including data collection, annotation, and task design.

Overview of APUN-Bench. Inspired by prior text-based pun benchmarks, APUN-Bench evaluates audio pun understanding through three stages: *pun recognition*, *pun word location*, and *pun meaning inference*. Specifically, the first two stages assess whether a speech segment contains a pun and identify the corresponding pun word, while the third evaluates the inference of secondary meanings for heterographic, homographic, and homophonic puns (Su et al., 2025). Compound puns containing multiple punning words are excluded. Overall, APUN-Bench comprises 4,447 audio clips, including 2,979 annotated puns and 1,468 negative samples.

3.1 Dataset Construction

The dataset construction consists of two main components: (1) Synthetic Audio Dataset: Audio samples synthesized from existing pun text data; (2) Real-world Audio Dataset: Audio data collected from real-world sources, including human recordings and publicly available websites, followed by editing and annotation. The details of data collection can be found at Appendix C. We provide detailed descriptions for 3 different stages, comprising a total of 6 subtasks.

Stage 1: Pun Recognition. This task assesses the model's ability to detect the presence of puns in audio, which serves as a prerequisite for more fine-grained comprehension and reasoning tasks. Inspired by Xu et al. (2024), we construct the question format that includes both pun and non-pun to mitigate potential biases, whose primary question format is "Determine whether the given audio is a pun/non-pun".

Stage 2: Pun Location. This task evaluates the ability of LALMs to localize pun words within spoken sentences. We investigate whether a model that can already detect the presence of a pun in

audio may still struggle to accurately identify the specific pun word. To enable more granular evaluation, we divide this task into two levels: (1). Coarse-grained Location: The model is evaluated for its ability to accurately locate the approximate location of pun words in the audio, without requiring the semantic accuracy of the identified words. Even if the model locates a semantically inaccurate word, as long as its location makes sense, it is considered correct. (2). Fine-Grained Location: this subtask further requires the model to accurately identify actual pun word in the audio.

For real-world speech data, manually labeling each audio segment in fine-grained detail is time-consuming. Previous research has demonstrated that leveraging LLMs to assist in dataset construction is both efficient and feasible (Gong et al., 2023; Liu et al., 2023; Hyun et al., 2024). Xu et al. (2024)) demonstrated that Claude ranks among the best existing models for understanding text-based puns. Accordingly, we first transcribe the audio into text using the Whisper-large model (Radford et al., 2022), and then employ Claude-Opus (particularly Claude-Opus-4-1) (Anthropic, 2025) to identify and locate the puns. To enhance its task comprehension and annotation accuracy, we prompt Claude-Opus with a small set of manually annotated examples along with feedback. More details can be found in the appendix D.

Stage 3: Pun Inference. This task evaluates the ability of LALMs to infer the intended meaning of puns in spoken sentences. Based on the characteristics, puns are classified into three categories for evaluation (Su et al., 2025): heterographic, homophonic, and homographic: (1) Heterographic puns involve words with similar but not identical pronunciations. For this type, the model needs to infer the correct pronunciation and the corresponding intended meaning substitute based on the given pun; (2) Homophonic puns involve words with identical pronunciations but different spellings. It is similar to heterographic puns, where the task is to identify the intended alternative word. (3) Homographic puns involve words with the same written form but different meanings. The model is required to generate phrases that capture both meanings and compare them with the gold standard, thereby assessing its ability to understand polysemy.

Based on the characteristics of different pun categories, we adopt a differentiated classification strategy. For homographic puns, which rely on lexical polysemy without spelling differences, we classify them by checking the consistency of word spellings. For homophonic and heterographic puns, whose defining feature lies in the phonetic differences between the punning word and its substitute, we transcribe the target words into International Phonetic Alphabet (IPA) symbols and determine their category based on whether their pronunciations match.

For heterographic and homophonic puns, we follow the same procedure as in Stage 2, using Claude-Opus to infer and generate the corresponding alternative words. For homographic puns, which involve lexical ambiguity, we derive the two gold-standard senses directly from WordNet (Miller, 1995). Finally, all annotations are verified for consistency using Claude-Opus and subsequently reviewed and refined by human annotators to ensure annotation accuracy.

Human Annotation. During benchmark construction, we conduct multiple rounds of manual development and reviews. In the raw data collection phase, each audio clip is rigorously examined for pun presence, pun words, categories, and replacement words to ensure the quality of the audio pun data. Audio samples with substandard quality are re-collected to ensure data reliability. In addition, when inferring double meanings for homographic puns in the synthetic speech dataset, we manually review the generated polysemous outputs, revise inappropriate content, and correct errors to obtain gold-standard annotations. The manual error correction rate is approximately 6%, and all checks are performed by the authors. Furthermore, the final human verification process is conducted to ensure the overall reliability of the dataset. More details can be found at Appendix E.

4 EVALUATION

Baseline Models. We evaluate APUN-Bench on a range of 10 recent LALMs, widely cited in the literature and capable of jointly processing text and audio information, including Qwen2-Audio-Instruct (Chu et al., 2024), Audio-Reasoner (Xie et al., 2025), Audio Flamingo 3 (Goel et al., 2025), Qwen-2.5-omni (Jin Xu et al., 2025), MiniCPM (Yao et al., 2024), Gemini 2.0 Flash (Jaech et al., 2024), SALMONN (Tang et al., 2023), GPT4o-Audio (Hurst et al., 2024), Omni-R1 (Rouditchenko et al., 2025) and MERaLiON2 (MERaLiON Team, 2024; Huang et al., 2025). Furthermore, we evaluate cascaded systems combining SenseVoiceSmall (An et al., 2024) or Whisper-Large (Radford

et al., 2022) with GPT-4o or Gemini 2.0 Flash. Each model is configured with a temperature of 0 to ensure deterministic outputs.

Evaluation Strategy. We adopt appropriate evaluation metrics for different tasks. For pun recognition tasks, we use accuracy, precision, recall, and F1 score as evaluation criteria, following the common practice in previous pun recognition research (Gepalova et al., 2024; Xu et al., 2024). Inspired by (Sung-Bin et al., 2024), we also report the proportion of "pun" responses produced by each model to examine potential biases in pun recognition.

For the pun word location stage, the coarse-grained setting requires only identifying the approximate location of the pun within the speech. Specifically, we first transcribe the target audio and obtain the IPA representation of each word, then compute the character-level edit distance between these representations and the IPA of the pun word predicted by the model. In addition, we incorporate phonetic characteristics by defining sets of vowels and consonants, mapping phonemes into feature vectors, and calculating their cosine similarity. We then combine these two similarity measures to select the word position with the highest overall similarity as the prediction result, as shown below:

$$S_{\text{edit}}(w_i) = 1 - \frac{d_{\text{edit}}(\phi(w_i), \phi(\hat{w}))}{\max(|\phi(w_i)|, |\phi(\hat{w})|)}, \tag{1}$$

$$S_{\cos}(w_i) = \cos\big(v(\phi(w_i)), v(\phi(\hat{w}))\big), \tag{2}$$

$$S(w_i) = S_{\text{edit}}(w_i) + S_{\cos}(w_i), \tag{3}$$

where $\phi(\cdot)$ denotes the IPA representation of a word, d_{edit} is the Levenshtein distance, and $v(\cdot)$ maps phonemes into feature vectors. For fine-grained location, an exact match is required to identify puns within speech. We first perform lemmatization on the words and then directly match them with the pun words predicted by the model to obtain the final evaluation outcome.

Following the evaluation protocol of the SemEval shared task Miller et al. (2017), we adopt different evaluation metrics for pun meaning inference. Heterographic and homophonic puns are evaluated using exact matching of the inferred alternative word, whereas homographic puns are evaluated based on the semantic similarity between the predicted meaning and the gold-standard meaning. Detailed evaluation protocols and threshold settings are provided in Appendix G.

4.1 Main Results

We systematically evaluate baseline models across the three stages of APUN-Bench on both synthetic and real-world datasets, with the results summarized in Table 2. In the pun recognition stage, most baseline models perform above chance level (50% accuracy). However, with the exception of GPT4o-Audio, Gemini2.0-Flash, and MERaLiON2, their recall rates are generally low, suggesting that the models tend to be overly conservative in identifying puns and are prone to false negatives. In the pun location stage, models are able to identify the approximate positions of more than half of the puns in the coarse-grained location task. Nevertheless, their performance drops substantially in the fine-grained detection task when recovering the exact pun words, revealing LALMs' limited ability to accurately transcribe specific pun words. In the pun inference stage, models achieve markedly better results on homophonic puns than on heterographic puns, consistent with findings that homophony constitutes only about 2–3% of the English lexicon (Marian et al., 2012), whereas a four-letter English word has on average about 10.33 phonological neighbors.

Moreover, proprietary models consistently outperformed open-source speech models in APUN-Bench, especially in pun location and inference, underscoring the gap between the two in deep semantic understanding. We further conduct McNemar's test to examine whether the results are statistically significant. For example, GPT4o-Audio achieves a highly significant advantage over MiniCPM-o-2.6 in fine-grained pun location ($p < 1 \times 10^{-10}$). Overall, although closed-source models exhibit advantages across tasks, current auditory language models still face significant challenges in detecting and understanding puns, with performance remaining unsatisfactory.

We further report the performance of cascaded models on audio pun understanding. Overall, cascaded systems underperform LALMs on pun recognition, but outperform them on pun word location and pun meaning inference. This suggests that stronger ASR can reduce transcription errors in word-level tasks, enabling better pun localization and inference. In addition, GPT-4o outperforms

Model	Size	Pun Recognition					Pun Location		Pun Inference		
		Acc. (↑)	Pre (↑)	Rec (↑)	F1 (↑)	Pun (%)	Coa (↑)	Fin (↑)	Heg (↑)	Hog (↑)	Hop (↑)
Synthetic Dataset											
Qwen2-Audio-Instruct	7B	45.78	71.70	39.45	50.90	38.06	50.81	31.77	13.47	17.91	26.97
SALMONN	13B	51.91	72.51	52.60	60.97	51.81	34.41	22.35	15.23	24.63	25.32
Audio-Reasoner	7B	55.27	72.62	59.81	65.59	54.47	50.39	32.31	23.05	27.25	37.50
Audio-Flamingo-3	7B	37.51	59.25	40.02	47.78	48.24	36.37	27.83	33.25	29.35	56.91
Qwen-2.5-omni	7B	64.04	78.71	68.07	73.00	61.76	60.98	48.01	40.55	32.38	60.19
MiniCPM-o-2.6	8.7B	62.31	78.00	65.77	71.37	27.92	59.94	44.62	39.54	37.36	67.43
Omni-R1	8.9B	65.28	77.39	72.58	74.91	66.97	63.49	49.43	44.20	39.73	63.48
MERaLiON2	10B	73.40	75.66	92.56	83.26	**87.37**	60.44	43.78	50.88	29.43	79.93
Gemini 2.0 Flash ✯	-	80.00	**91.04**	79.86	85.08	62.39	65.80	48.34	52.39	53.29	92.11
GPT4o-Audio ✯	-	78.04	78.48	**95.92**	**86.33**	83.87	75.97	56.63	59.06	57.81	85.19
SenseVoiceSmall+GPT4o	-	79.44	84.52	87.16	85.82	73.57	75.04	56.05	**67.50**	67.79	98.02
SenseVoiceSmall+Gemini2.0-Flash	-	78.65	84.12	86.64	85.36	73.47	74.08	55.46	55.16	60.10	78.28
Whisper-Large+GPT4o	-	**80.29**	87.28	84.69	85.97	68.91	**79.19**	**62.78**	67.00	**70.49**	**98.35**
Whisper-Large+Gemini2.0-Flash	-	77.47	84.37	84.58	84.47	71.46	77.56	60.86	55.54	62.94	89.80
Real-world Dataset											
Qwen2-Audio-Instruct	7B	58.64	95.11	37.61	53.90	25.53	49.05	24.95	23.87	21.59	26.02
SALMONN	13B	44.86	57.07	60.41	58.69	68.56	28.82	14.11	18.38	23.15	16.43
Audio-Reasoner	7B	70.99	89.97	61.94	73.37	41.14	48.71	25.98	34.97	25.70	38.35
Audio-Flamingo-3	7B	63.65	97.39	45.09	61.64	29.99	45.07	31.78	37.66	25.96	50.68
Qwen-2.5-omni	7B	71.68	**99.09**	56.79	72.21	37.12	59.21	42.85	50.00	35.78	57.53
MiniCPM-o-2.6	8.7B	74.35	98.08	61.62	75.68	40.80	54.73	38.03	43.69	34.86	54.79
Omni-R1	8.9B	83.50	97.37	76.59	85.74	50.95	58.86	41.31	44.59	35.00	57.53
MERaLiON2	10B	**88.18**	87.05	96.04	**91.32**	**71.46**	54.56	35.45	49.54	32.28	62.50
Gemini 2.0 Flash ✯	-	87.51	84.82	**98.51**	91.15	60.98	67.52	51.03	62.78	47.36	82.85
GPT4o-Audio ✯	-	85.36	87.02	92.56	89.71	66.89	66.95	50.08	60.98	50.70	76.71
SenseVoiceSmall+GPT4o	-	84.94	96.45	79.69	87.27	53.51	67.29	43.78	71.74	**67.36**	**87.67**
SenseVoiceSmall+Gemini2.0-Flash	-	83.61	92.56	81.41	86.63	56.97	67.12	44.06	50.06	49.47	86.30
Whisper-Large+GPT4o	-	85.20	98.26	78.47	87.25	51.28	**76.41**	**55.76**	**72.19**	67.01	84.93
Whisper-Large+Gemini2.0-Flash	-	83.08	95.34	77.89	85.74	52.73	69.53	49.74	52.46	52.98	82.19

Table 2: Evaluation Results of the Synthetic and Real-World Datasets on APUN-Bench in various LALMs and cascaded systems. ✯ represents the proprietary LALMs and the symbol '–' indicates that the model size is not publicly disclosed. Coa and Fin represent the accuracy of coarse-grained and fine-grained locations. Hog, hop and heg denote the accuracy of homographic puns, homophonic puns and heterographic puns.

other text-based models, highlighting the importance of downstream reasoning capability. To further disentangle speech recognition errors from higher-level pun reasoning errors, we report ASR word error rates in Appendix C.

4.2 Pun Transcript vs Full Audio?

To better understand the respective contributions of textual and acoustic information to audio pun understanding, we compare text-only and audio-only models on the real-world subset of APUN-Bench, as shown in Figure 2. Overall, text-only models generally outperform audio-only models on pun recognition (PR) and pun word location (PL), with the largest performance gap observed in PL. For example, GPT4o-Text achieves 93.03% on PL, substantially outperforming GPT4o-Audio (50.08%). This suggests that the primary bottleneck of LALMs lies in accurately perceiving and transcribing pun words from speech. Since text-only models operate on error-free transcripts, their superior performance mainly reflects the elimination of speech perception and transcription errors.

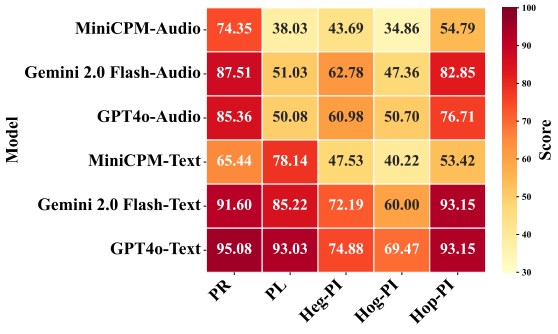

Figure 2: Comparison of text-only and audio-only models on the real-world subset of APUN-Bench across pun recognition (PR), pun location (PL), and three pun meaning inference subtasks: heterographic (Heg-PI), homographic (Hog-PI), and homophonic (Hop-PI).

In contrast, the performance gap is considerably smaller across the three pun meaning

inference subtasks. For GPT4o, the gap decreases from 42.95% on PL to 13.90%, 18.77%, and 16.44% on Heg-PI, Hog-PI, and Hop-PI, respectively. Moreover, MiniCPM-o-2.6 slightly outperforms its text-only counterpart on homophonic inference (54.79% vs. 53.42%). These findings suggest that although speech introduces transcription challenges, it also preserves pronunciation cues that are unavailable in text, providing complementary evidence for resolving phonetic ambiguity.

4.3 What is the performance of LALMs in Identifying the Position of Pun Words?

According to the incongruity theory of humour, punchlines typically appear at the end of a sentence (Shahaf et al., 2015). Building on this, we analyze the positional preferences of different models in identifying puns and compared them with manual annotation results, as shown in Figure 3. Sentence Interval refers to the normalized sentence position; for example, 0–0.2 denotes the first 20% of the whole sentence. The annotation data indicate that the vast majority of puns occur within the last 40% of a sentence, with fewer than 10% located in the first 60%, further supporting the incongruity theory. In contrast, most model predictions exhibit a more uniform distribution: although the last 20% still contains the highest proportion of predicted puns, more than 20% appear in the first 60%. These findings suggest that guiding models to focus more on the latter half of a sentence may improve their performance in pun location.

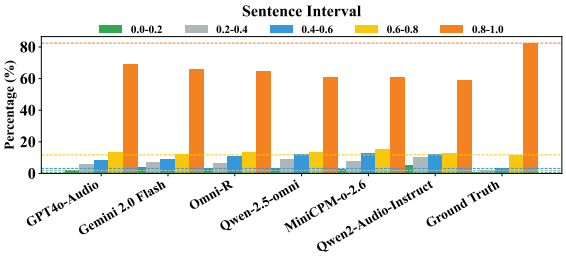

Figure 3: Statistical comparison of pun positions predicted by different LALMs and the ground truth in APUN-Bench, showing their distributions across sentence intervals (beginning, middle, and end).

4.3.1 Improving the Performance of Pun Location in Sentence

To test this hypothesis, we introduce a **Pun** **P**osition-**A**ware **P**rompt-based (**PunPAP**) strategy, which explicitly encourages the model to pay greater attention to the latter half of the sentence during pun localization, with the revised prompt segment: *...Identify the pun word in the audio, paying particular attention to the latter half of the sentence...*

Model	Method	Coa	Fin
Qwen2.5-Omni	Vanilla	60.98	48.01
Qwen2.5-Omni	PunPAP	**63.49**	**49.56**
MiniCPM-o-2.6	Vanilla	59.94	44.62
MiniCPM-o-2.6	PunPAP	**62.32**	**46.42**

Table 3: Performance improvements after prompt refinement for Qwen2.5-Omni and MiniCPM-o-2.6 on pun location.

We conduct evaluations on Qwen-2.5-omni-7B and MiniCPM-o-2.6, achieving 63.49% and 62.32% accuracy in coarse-grained pun location, respectively, surpassing the baseline models by 2.51% and 2.38% percentage points, as shown in Table 3. For fine-grained location, the models achieve 49.56% and 46.42% accuracy, outperforming the baselines by 1.55% and 1.80% percentage points, respectively. While the overall improvements are limited, the results indicate that emphasizing the latter part of a sentence may improve pun localization and provide initial insights for future research.

4.4 Audio Pun Inference: Where do LALMs Fall Short?

Figure 4 presents the distribution of error types for Omni-R1 and GPT4o-Audio over 450 randomly sampled instances, including 150 examples from each pun category. The definition details are provided in Table 4. By comparing model predictions with gold-standard annotations and corresponding transcriptions, we manually identify recurring error patterns and summarize their distribution, with evaluation metrics used as auxiliary evidence for categorization. For heterographic and homophonic puns, the dominant error type in both models is WIE (including Similar and No Similar), accounting for more than one-third of all errors. In contrast, for homographic puns, the dominant

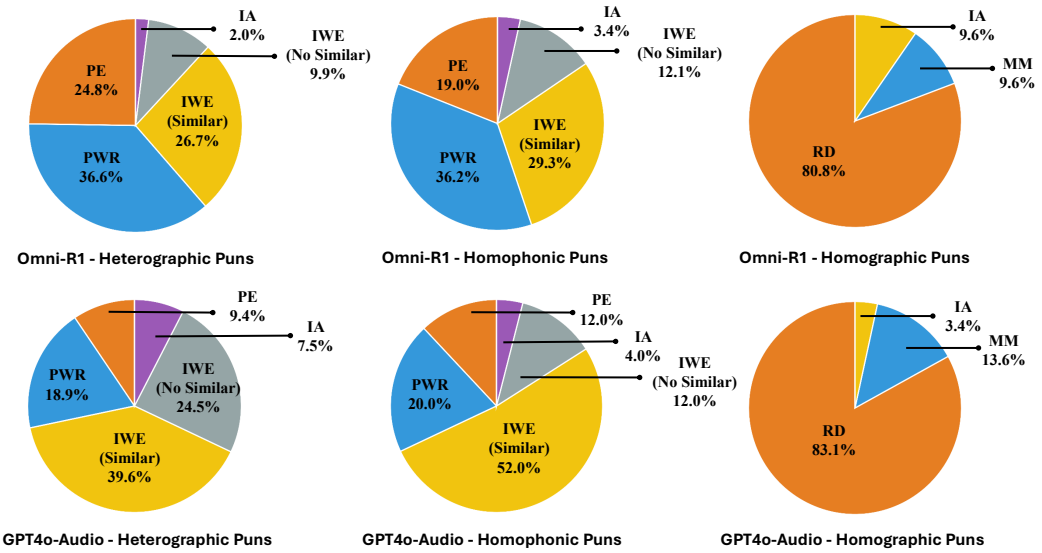

Figure 4: The distribution of error types in pun inference for Omni-R1 and GPT4o-Audio across heterographic, homographic, and homophonic puns.

error shifts to RD, which constitutes over three-quarters of the errors, while MM accounts for only around 10%. This suggests that although the models are generally capable of perceiving speech and identifying one layer of pun meaning, they still struggle to recover the secondary meaning required for full interpretation. Moreover, GPT4o-Audio exhibits a notably lower proportion of PWR errors than Omni-R1 on heterographic and homophonic puns, indicating that larger-scale models are less likely to rigidly reproduce the pun word itself as output. Finally, IA is the least frequent error type across all categories and both models, suggesting that most models understand the task requirements and attempt pun-related inference rather than producing unrelated responses.

4.4.1 Pun Inference Alignment

To enhance audio pun inference, we propose **PunAlign**, a two-stage training framework consisting of supervised fine-tuning (SFT) followed by direct preference optimization (DPO). First, Qwen2.5-Omni and MiniCPM-o-2.6 are fine-tuned using SFT to improve instruction following and reduce errors such as IA. We then construct preference pairs for DPO based on the observed error categories. For each input, the reference answer or a correct model response is used as the chosen response, while an incorrect response generated by the model serves as the rejected response. For instances initially answered correctly, additional rejected responses exhibiting PE or PWR are constructed to augment the preference data. For model training, the synthetic and real-world datasets are combined and randomly split into training and validation subsets with an 8:2 ratio. Benchmark results are reported on a separate held-out test set, excluded from model training. Detailed training hyperparameters are provided in Appendix H.

4.4.2 Improvement Results

Table 5 presents the ablation results of the SFT and DPO fine-tuning strategies on Qwen2.5-Omni and MiniCPM-o-2.6. The results show that SFT substantially enhances the models' speech pun reasoning ability. For instance, SFT improves MiniCPM-o-2.6 by 7.30% and 17.51% percentage points on the Heterographic+Homophonic and Homographic tasks, respectively. Applying DPO with error type-based preference data further improves performance, while combining SFT and

Model	SFT	DPO	Heg+Hop	Hog
Qwen2.5-Omni	-	-	42.27	32.38
	-	✓	42.73	36.82
	✓	-	55.00	46.15
	✓	✓	**58.64**	**49.22**
MiniCPM-o-2.6	-	-	46.12	33.07
	-	✓	47.03	34.24
	✓	-	53.42	50.58
	✓	✓	**55.25**	**53.31**

Table 5: Performance improvements of Qwen2.5-Omni and MiniCPM-o-2.6 on pun inference with fine-tuning strategy.

Error Type	Definition	Example (Audio Transcription)	Prediction	Ground Truth
Heterographic Puns & Homophonic Puns				
Position Error (PE)	The model fails to provide the correct word corresponding to the given pun word and may instead return a word from another position in the sentence.	When asked to picture the perfect modern defensive weapon the Claymore springs to **mine**.	Claymore	mind
Pun Word Repeat (PWR)	The model outputs the given pun word itself rather than its intended meaning.	Did you hear about the nervous preacher ? He had sweaty **psalms**.	psalms	palm
Word Inference Error (WIE) (Similar)	The model infer an incorrect word, often manifested as another word with a similar pronunciation.	One ear of corn said to the other'You're getting **husky**'.	hoarse	husk
Word Inference Error (WIE) (No Similar)	The model infer an incorrect word, often manifested as another word with a completely different pronunciation.	George Westinghouse was a refrigerator **magnate**.	entrepreneur	magnet
Irrelevant Answer (IA)	The model's answer is unrelated to the pun inference for the pun word.	She was only a Butcher's daughter, but there wasn't much more she could **loin**.	line human given an audio ...	learn
Homographic Puns				
Reasoning Disorder (RD)	The model can only infer a single layer of meaning, where one pair of matches shows high similarity while the other pair fails to align.	How could I trust the ceiling fan installer when I knew he was always **screwing up**.	making mistakes AND causing trouble	making mistakes AND installing screws
Meaning Mismatch (MM)	The model generates double-meaning words or phrases that deviate from the intended meanings defined in the gold standard.	Can honeybee abuse lead to a **sting** operation?	steamboat AND steamship	undercover police activity AND painful insect bite
Irrelevant Answer (IA)	The model's answer is irrelevant style of inference meaning or incorrect styles for the pun word.	Then there was the occasion I spotted a health - oriented cafe displaying a sign reading' California **Shakes**.' Obviously, I thought.	The two meanings of 'shake' in this context	earthquake AND milkshake

Table 4: Error types of pun inference in heterographic, homophonic and homographic puns.

DPO achieves the best results. For example, Qwen2.5-Omni gains 16.37% and 16.84% percentage points over the baseline on the two tasks, respectively. These results demonstrate the complementarity of SFT and DPO, with SFT improving task adaptation and DPO further refining reasoning behavior.

5 LIMITATIONS

While this research offers new perspectives and insights into the understanding of audio puns, several limitations remain: (1) The scope of pun types extends beyond the three categories examined in this study, with more complex forms such as recursive and compound puns remaining outside its consideration. (2) Puns often arise in more complex speech environments, such as multi-turn dialogues, whereas our dataset is restricted to single-sentence instances. (3) The size of the real-world corpus still remains limited, which constrains the statistical robustness of our findings. However, despite these limitations, this paper represents the first systematic study of audio pun understanding in LALMs, and its data and analyses offer valuable references for subsequent research on the intersection of language and speech understanding.

6 CONCLUSION

In this paper, we introduce APUN-Bench, the first benchmark for comprehensively evaluating audio pun understanding in LALMs. APUN-Bench comprises the first audio pun dataset, an automatic evaluation protocol, and three progressively challenging tasks: pun recognition, pun word location, and pun meaning inference. Extensive experiments on 10 open-source and proprietary LALMs, together with 4 cascaded systems, reveal that current models still struggle with fine-grained pun localization and meaning inference, particularly for heterographic and homophonic puns. Motivated by these findings, we further propose PunPAP and PunAlign methods, which consistently improve pun localization and inference performance. We hope APUN-Bench will facilitate future research on humour-aware audio understanding and advanced speech reasoning.

AI USE STATEMENT

In this work, we used generative AI tools to assist in implementing the method and evaluation code, to synthesize audio data for dataset construction under the authors' guidance, and to refine research hypotheses, assess their feasibility, and assist in interpreting results. All AI-generated audio data were subsequently reviewed and verified by the authors as part of the dataset construction and human verification process. We did not use generative AI tools to propose the research idea, design the experiments, or develop the theoretical account of the method. Additionally, we used generative AI tools for editing the text for readability. We reviewed all AI-assisted work: the code behind every reported run was checked and verified by the authors, and the text was written and revised by the authors. We take responsibility for the final content of this work, including text, claims, data, or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

The Institutional Review Board (IRB) of our institution has approved the human studies presented in this paper. In addition, we have obtained permission to use data collection from the public pun website *O. Henry Museum Pun-Off*.

REPRODUCIBILITY STATEMENT

The code and APUN-Bench dataset, including audio samples and annotations, will be released upon publication. Details of dataset construction, annotation, and human verification are provided in Section 3 and Appendices E. The implementation details and training hyperparameters of the proposed improvement methods are provided in Appendix H.

REFERENCES

Elena Aleksandrova. Pun-based jokes and linguistic creativity. *The European Journal of Humour Research*, 10(1):88–107, 2022.

Keyu An, Qian Chen, Chong Deng, Zhihao Du, Changfeng Gao, Zhifu Gao, Yue Gu, Ting He, Hangrui Hu, Kai Hu, et al. Funaudiollm: Voice understanding and generation foundation models for natural interaction between humans and llms. *arXiv preprint arXiv:2407.04051*, 2024.

Anthropic. System card: Claude opus 4 & claude sonnet 4. `https://www-cdn.anthropic.com/4263b940cabb546aa0e3283f35b686f4f3b2ff47.pdf`, May 2025.

Mohamed Benzeghiba, Renato De Mori, Olivier Deroo, Stephane Dupont, Teodora Erbes, Denis Jouvet, Luciano Fissore, Pietro Laface, Alfred Mertins, Christophe Ris, et al. Automatic speech recognition and speech variability: A review. *Speech communication*, 49(10-11):763–786, 2007.

Yunfei Chu, Jin Xu, Qian Yang, Haojie Wei, Xipin Wei, Zhifang Guo, Yichong Leng, Yuanjun Lv, Jinzheng He, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen2-audio technical report. *arXiv preprint arXiv:2407.10759*, 2024.

Yufeng Diao, Hongfei Lin, Di Wu, Liang Yang, Kan Xu, Zhihao Yang, Jian Wang, Shaowu Zhang, Bo Xu, and Dongyu Zhang. Weca: A wordnet-encoded collocation-attention network for homographic pun recognition. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pp. 2507–2516, 2018.

Kuofeng Gao, Shu-Tao Xia, Ke Xu, Philip Torr, and Jindong Gu. Benchmarking open-ended audio dialogue understanding for large audio-language models. *arXiv preprint arXiv:2412.05167*, 2024.

Arina Gepalova, Adrian-Gabriel Chifu, and Sébastien Fournier. Clef 2024 joker task 1: exploring pun detection using the t5 transformer model. In *Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2024). CEUR Workshop Proceedings*, pp. 1857–1861, 2024.

Arushi Goel, Sreyan Ghosh, Jaehyeon Kim, Sonal Kumar, Zhifeng Kong, Sang-gil Lee, Chao-Han Huck Yang, Ramani Duraiswami, Dinesh Manocha, Rafael Valle, et al. Audio flamingo 3: Advancing audio intelligence with fully open large audio language models. *arXiv preprint arXiv:2507.08128*, 2025.

Yuan Gong, Hongyin Luo, Alexander H Liu, Leonid Karlinsky, and James Glass. Listen, think, and understand. *arXiv preprint arXiv:2305.10790*, 2023.

Mohammad Ali Heidari-Shahreza. Humour integrated language learning (hill): defining, defending and developing an emerging field. *The European Journal of Humour Research*, 12(2):176–196, 2024.

Xin Huang, Tarun Kumar Vangani, Minh Duc Pham, Xunlong Zou, Bin Wang, Zhengyuan Liu, and Ai Ti Aw. Meralion-textllm: Cross-lingual understanding of large language models in chinese, indonesian, malay, and singlish, 2025. URL https://arxiv.org/abs/2501.08335.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

Lee Hyun, Kim Sung-Bin, Seungju Han, Youngjae Yu, and Tae-Hyun Oh. SMILE: Multimodal dataset for understanding laughter in video with language models. In *Findings of the Association for Computational Linguistics: NAACL 2024*. Association for Computational Linguistics, 2024.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

Sophie Jentzsch and Kristian Kersting. Chatgpt is fun, but it is not funny! humor is still challenging large language models. *arXiv preprint arXiv:2306.04563*, 2023.

Zhifang Guo Jin Xu, Ting He Jinzheng He, Hangrui Hu, Jialin Wang Shuai Bai, Keqin Chen, Bin Zhang Yang Fan, Kai Dang, Yunfei Chu Xiong Wang, and Junyang Lin. Qwen2.5-omni technical report. *arXiv preprint arXiv:2503.20215*, 2025.

Hanna Karpenko. Humor as a part of interpersonal communication. *Journal of Vasyl Stefanyk Precarpathian National University. Series of social and human sciences*, (4, no. 1):195–199, 2017.

Yoach Lacombe, Vaibhav Srivastav, and Sanchit Gandhi. Parler-tts. https://github.com/huggingface/parler-tts, 2024.

Kristin Lems. Pun work helps english learners get the joke. *The reading teacher*, 65(3):197–202, 2011.

Zhenyu Li, Minghao Zhao, Tian Xu, Rui Chen, Xinyu Yang, Hao Wu, and Shanshan Wang. Meralion-audiollm: Bridging audio and language with large language models. *arXiv preprint arXiv:2412.09818*, 2024.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023.

Fuli Luo, Shunyao Li, Pengcheng Yang, Baobao Chang, Zhifang Sui, Xu Sun, et al. Pun-gan: Generative adversarial network for pun generation. *arXiv preprint arXiv:1910.10950*, 2019.

Dan Lyth and Simon King. Natural language guidance of high-fidelity text-to-speech with synthetic annotations, 2024.

Ziyang Ma, Zhuo Chen, Yuping Wang, Eng Siong Chng, and Xie Chen. Audio-cot: Exploring chain-of-thought reasoning in large audio language model. *arXiv preprint arXiv:2501.07246*, 2025.

Viorica Marian, James Bartolotti, Sarah Chabal, and Anthony Shook. Clearpond: Cross-linguistic easy-access resource for phonological and orthographic neighborhood densities. 2012.

MERaLiON Team. Meralion-audiollm: Bridging audio and language with large language models, 2024. URL https://arxiv.org/abs/2412.09818.

George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11): 39–41, 1995.

Tristan Miller, Christian F Hempelmann, and Iryna Gurevych. Semeval-2017 task 7: Detection and interpretation of english puns. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pp. 58–68, 2017.

Meredith Ringel Morris, Jascha Sohl-Dickstein, Noah Fiedel, Tris Warkentin, Allan Dafoe, Aleksandra Faust, Clement Farabet, and Shane Legg. Position: Levels of agi for operationalizing progress on the path to agi. In *Forty-first International Conference on Machine Learning*, 2024.

Kun Ouyang, Yuanxin Liu, Shicheng Li, Yi Liu, Hao Zhou, Fandong Meng, Jie Zhou, and Xu Sun. Punchbench: Benchmarking mllms in multimodal punchline comprehension. *arXiv preprint arXiv:2412.11906*, 2024.

Anna Palmann and Tristan Miller. What's in a pun? assessing the relationship between phonological distance and perceived funniness of punning jokes. *HUMOR*, (0), 2025.

Alan Scott Partington. A linguistic account of wordplay: The lexical grammar of punning. *Journal of Pragmatics*, 41(9):1794–1809, 2009.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision, 2022. URL `https://arxiv.org/abs/2212.04356`.

Andrew Rouditchenko, Saurabhchand Bhati, Edson Araujo, Samuel Thomas, Hilde Kuehne, Rogerio Feris, and James Glass. Omni-r1: Do you really need audio to fine-tune your audio llm? *arXiv preprint arXiv:2505.09439*, 2025.

S Sakshi, Utkarsh Tyagi, Sonal Kumar, Ashish Seth, Ramaneswaran Selvakumar, Oriol Nieto, Ramani Duraiswami, Sreyan Ghosh, and Dinesh Manocha. Mmau: A massive multi-task audio understanding and reasoning benchmark. *arXiv preprint arXiv:2410.19168*, 2024.

Dafna Shahaf, Eric Horvitz, and Robert Mankoff. Inside jokes: Identifying humorous cartoon captions. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1065–1074, 2015.

Yuchen Su, Yonghua Zhu, Ruofan Wang, Zijian Huang, Diana Benavides-Prado, and Michael Witbrock. A survey of pun generation: Datasets, evaluations and methodologies. *arXiv preprint arXiv:2507.04793*, 2025.

Jiao Sun, Anjali Narayan-Chen, Shereen Oraby, Alessandra Cervone, Tagyoung Chung, Jing Huang, Yang Liu, and Nanyun Peng. Expunations: Augmenting puns with keywords and explanations. *arXiv preprint arXiv:2210.13513*, 2022a.

Jiao Sun, Anjali Narayan-Chen, Shereen Oraby, Shuyang Gao, Tagyoung Chung, Jing Huang, Yang Liu, and Nanyun Peng. Context-situated pun generation. *arXiv preprint arXiv:2210.13522*, 2022b.

Kim Sung-Bin, Oh Hyun-Bin, JungMok Lee, Arda Senocak, Joon Son Chung, and Tae-Hyun Oh. Avhbench: A cross-modal hallucination benchmark for audio-visual large language models. *arXiv preprint arXiv:2410.18325*, 2024.

Yunxin Tang, Yusheng Liu, Zhiyong Wu, Jiajun Li, Qian Wang, Zhongkai Meng, Zheng Zhang, Lei Xie, and Ming Wang. Salmonn: Towards generic hearing abilities for large language models. *arXiv preprint arXiv:2309.07985*, 2023.

Adith Santosh Thaniserikaran and Govind Harikrishnan. Phonetic cues improve llm-based pun detection in short text. In *Proceedings of the 2nd Workshop on Computational Humor (CHum 2026)*, pp. 72–80, 2026.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouédec. TRL: Transformers Reinforcement Learning, 2020. URL `https://github.com/huggingface/trl`.

Bin Wang, Xunlong Zou, Geyu Lin, Shuo Sun, Zhuohan Liu, Wenyu Zhang, Zhengyuan Liu, AiTi Aw, and Nancy F Chen. Audiobench: A universal benchmark for audio large language models. *arXiv preprint arXiv:2406.16020*, 2024.

Dingdong Wang, Jincenzi Wu, Junan Li, Dongchao Yang, Xueyuan Chen, Tianhua Zhang, and Helen Meng. Mmsu: A massive multi-task spoken language understanding and reasoning benchmark. *arXiv preprint arXiv:2506.04779*, 2025.

Zhifei Xie, Mingbao Lin, Zihang Liu, Pengcheng Wu, Shuicheng Yan, and Chunyan Miao. Audio-reasoner: Improving reasoning capability in large audio language models, 2025. URL https://arxiv.org/abs/2503.02318.

Zhijun Xu, Siyu Yuan, Lingjie Chen, and Deqing Yang. " a good pun is its own reword": Can large language models understand puns? *arXiv preprint arXiv:2404.13599*, 2024.

Qian Yang, Jin Xu, Wenrui Liu, Yunfei Chu, Ziyue Jiang, Xiaohuan Zhou, Yichong Leng, Yuanjun Lv, Zhou Zhao, Chang Zhou, et al. Air-bench: Benchmarking large audio-language models via generative comprehension. *arXiv preprint arXiv:2402.07729*, 2024.

Yuan Yao, Tianyu Yu, Ao Zhang, Chongyi Wang, Junbo Cui, Hongji Zhu, Tianchi Cai, Haoyu Li, Weilin Zhao, Zhihui He, et al. Minicpm-v: A gpt-4v level mllm on your phone. *arXiv preprint arXiv:2408.01800*, 2024.

Zhiwei Yu, Jiwei Tan, and Xiaojun Wan. A neural approach to pun generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1650–1660, 2018.

Jian Zhang, Linhao Zhang, Bokai Lei, Chuhan Wu, Wei Jia, and Xiao Zhou. Wildspeech-bench: Benchmarking audio llms in natural speech conversation. *arXiv preprint arXiv:2506.21875*, 2025.

Tuo Zhang, Tiantian Feng, Yibin Ni, Mengqin Cao, Ruying Liu, Katharine Butler, Yanjun Weng, Mi Zhang, Shrikanth S Narayanan, and Salman Avestimehr. Creating a lens of chinese culture: A multimodal dataset for chinese pun rebus art understanding. *arXiv preprint arXiv:2406.10318*, 2024.

Wei Zheng and Xiaolu Wang. Contextual support for less salient homophones and pun humor appreciation: evidence from eye movements in reading chinese homophone puns. *Frontiers in Psychology*, 13:875479, 2022.

Yichao Zhou, Jyun-Yu Jiang, Jieyu Zhao, Kai-Wei Chang, and Wei Wang. " the boating store had its best sail ever": Pronunciation-attentive contextualized pun recognition. *arXiv preprint arXiv:2004.14457*, 2020.

Yanyan Zou and Wei Lu. Joint detection and location of english puns. *arXiv preprint arXiv:1909.00175*, 2019.

A APPENDIX A: BASELINE MODELS

Qwen2-Audio-Instruct. Chu et al. (2024) present Qwen2-Audio, a large audio–language model that integrates an audio encoder with the Qwen LLM to process spoken queries, dialogues, and instruction-following tasks. The instruct variant evaluated here follows the official release, focusing on robust audio understanding across speech and environmental sound domains.

SALMONN. Tang et al. (2023) introduce SALMONN, which equips LLMs with general "hearing" abilities. It combines speech, audio event, and music encoders with a frozen LLM backbone, supporting tasks ranging from ASR and speech translation to audio captioning. SALMONN has been widely adopted as an open-source baseline for broad auditory competence.

Audio-Reasoner. Xie et al. (2025) propose Audio-Reasoner, a reasoning-enhanced model trained with structured chain-of-thought supervision and a large corpus of annotated audio–text pairs (CoTA). It is designed to strengthen multi-step deduction and temporal reasoning, offering explicit reasoning traces alongside competitive performance.

Laura	Gary	Jon	Lea
Karen	Rick	Brenda	David
Eileen	Jordan	Mike	Yann
Joy	James	Eric	Lauren
Rose	Will	Jason	Aaron
Naomie	Alisa	Patrick	Jerry
Tina	Jenna	Bill	Tom
Carol	Barbara	Rebecca	Anna
Bruce	Emily		

Table 6: List of the 34 speakers used in Parler-TTS.

Qwen-2.5-Omni. Jin Xu et al. (2025) extend the Qwen family with an omni-modal series capable of processing text, audio, and vision within a unified interface. The 3B and 7B variants are optimised for instruction following, open-domain spoken QA, and multimodal reasoning, providing strong baselines at different parameter scales.

MiniCPM. Yao et al. (2024) present MiniCPM-V, a lightweight multimodal model optimised for efficient deployment on mobile devices. Despite its compact scale, MiniCPM demonstrates performance comparable to GPT-4V on several multimodal tasks. Its audio-capable versions support real-time speech understanding for instruction-following scenarios.

Audio Flamingo 3. Goel et al. (2025) introduce Audio Flamingo 3 (AF3), a fully open state-of-the-art LALMs capable of multi-turn, multi-audio dialogue and long-context audio processing, enabling stronger cross-domain capabilities over speech, sound, and music.

Omni-R1. Rouditchenko et al. (2025) explore whether direct audio fine-tuning is necessary for competitive performance. Omni-R1 leverages large-scale instruction tuning on text paired with synthetic audio, achieving strong spoken understanding and reasoning without explicit raw-audio fine-tuning.

MERaLiON-AudioLLM. Li et al. (2024) introduce MERaLiON-AudioLLM, a multilingual and multimodal large audio–language model. The family emphasises robustness to multilingual inputs, accents, and code-switching. In our evaluation, we reference the 10B variant (*MERaLiON2*), which extends this line with improved large-scale instruction-following capabilities.

GPT-4o-Audio. Hurst et al. (2024) document the GPT-4o family; we evaluate the audio-enabled version that accepts spoken input for transcription, dialogue, and multimodal reasoning. As a closed-source model, GPT-4o-Audio sets a strong commercial baseline for real-time, latency-sensitive applications.

Gemini 2.0 Flash. The Gemini Flash models are proprietary systems released by Google DeepMind as part of the Gemini family Jaech et al. (2024). They are optimised for low-latency instruction following with speech input. We include Gemini 2.0 as a strong closed-source baseline to contextualise open-source performance.

B APPENDIX B: DATA COLLECTION

The dataset construction consists of two main components: (1) Synthetic Audio Dataset: Audio samples synthesized from existing pun text data; (2) Real-world Audio Dataset: Audio data collected from real-world sources, including human recordings and publicly available websites, followed by editing and annotation.

B.1 SYNTHETIC AUDIO DATASET.

One challenge in constructing the benchmark is collecting audio data and performing fine-grained annotations. To address this, we used the existing SemEval 2017 pun text dataset (Miller et al., 2017), which comprises 2,396 positive instances (puns) and 1,152 negative instances (non-puns). The positive set comprises instances annotated with both the punning word and its corresponding replacement. Then, we employ state-of-the-art speech synthesis Parler-TTS (Lacombe et al., 2024;

Model	Heterographic		Homophonic		Homographic	
	Coa (↑)	Fin (↑)	Coa (↑)	Fin (↑)	Coa (↑)	Fin (↑)
SALMONN	32.37	17.12	35.86	8.55	35.32	28.81
Qwen2-Audio-Instruct	52.14	29.47	44.41	13.81	51.51	37.41
Audio-Reasoner	50.50	29.34	39.47	11.18	52.90	39.11
Qwen-2.5-omni	53.53	36.39	56.91	21.38	66.54	61.42
MiniCPM	56.30	34.13	62.50	20.06	61.58	56.85
Omni-R1	58.06	38.16	60.53	21.71	67.54	62.89
MERaLiON2	55.16	31.36	62.50	17.10	63.21	57.70
Gemini 2.0 Flash ✯	60.96	36.90	70.72	35.19	67.62	58.48
GPT4o-Audio ✯	**74.06**	**44.20**	**76.64**	**42.43**	**76.99**	**67.62**

Table 7: Evaluation results of the Synthetic Dataset on APUN-Bench. We evaluate various LALMs on our proposed APUN-Bench. ✯ represents the proprietary LALMs.

Lyth & King, 2024), which is capable of generating high-quality, natural-sounding speech conditioned on various speaker characteristics, such as gender, pitch, and speaking style, to randomly assign each text to one of 34 distinct synthetic speakers, ensuring diversity in speaker characteristics, as shown in Table 6. Finally, a manual filtering step is conducted to screen the missing words or unclear pronunciations to improve the overall quality of the speech data.

B.2 Real-world Audio Dataset.

In addition to the synthetic audio dataset, we collect publicly available videos and manually segment the speech content from the *O. Henry Museum Pun-Off* [1], a long-standing global pun competition where contestants must create puns within a limited time to compete for the championship. To address the limited availability of public audio data, we further collect data from pun websites [2] and create realistic pun speech by manual recording. Furthermore, we constructed negative samples from real speech. The dataset contains 583 positive samples and 316 negative samples. These annotated data effectively introduce the challenge of understanding puns in audio, requiring models to accurately detect and interpret them under complex conditions. In total, our dataset comprises 3,548 synthetic and 899 real-world audio samples, ensuring diversity across data sources.

C Appendix C: Supplementary Analysis

C.1 Are LALMs Biased Toward Certain Types of Pun Word Location?

In the pun location stage, different models show notable performance gaps between coarse and fine-grained locations, with differences of around 20% being common across models. While this indicates that the models are generally consistent in transcribing and recognizing audio puns, we further explore their recognition capabilities across specific pun categories. Table 7 presents the performance of various models on three categories of puns, evaluated through both coarse-grained and fine-grained tasks in synthetic dataset. The results reveal that most models perform significantly better on homographic puns, excelling in both location detection and exact matching. For heterographic and homophonic puns, although their accuracy in coarse location is comparable, the accuracy of fine-grained location for homophonic puns is substantially lower. This finding highlights that existing models still encounter considerable difficulties in accurately reproducing fully homophonic puns.

C.2 Part-of-Speech Distributions

Additionally, we analyze the part-of-speech distributions of pun words in both the synthetic and real-world subsets using SpaCy, identifying the top five most frequent categories as shown in Table 8.

[1] https://www.punoffatx.brushsquaremuseums.org/

[2] https://pun.me/

Model	NOUN	ADV	VERB	ADJ	PROPN
Synthetic Dataset	954	312	246	202	146
Real-world Dataset	208	36	131	81	93

Table 8: The part-of-speech distributions between Synthetic and Real-world Datasets.

ASR Model	Synthetic			Real-world		
	Heg (↓)	Hog (↓)	Hop (↓)	Heg (↓)	Hog (↓)	Hop (↓)
SenseVoiceSmall	0.144	0.126	0.151	0.177	0.134	0.269
Whisper-Large	0.111	0.097	0.123	0.154	0.093	0.221

Table 9: Word Error Rate (WER) of SenseVoiceSmall and Whisper-Large on synthetic and real-world datasets.

The findings indicate that nouns and verbs are more likely to serve as pun-bearing words, implying that certain lexical categories are inherently more productive in pun construction. This observation provides insight into the lexical bias underpinning pun formation.

C.3 Word Error Rate

We report the word error rate (WER) of the SenseVoiceSmall and Whisper-Large ASR models on both synthetic and real data to quantify their transcription errors on audio puns. The results are shown in Table 9. Overall, the WER on real-world data is higher than that on synthetic data, indicating that more complex environmental noise, speaker variability, and natural pronunciation variations in real speech interfere with the accuracy of audio transcription. Further analysis from the perspective of pun type reveals that, under both data settings, the WER for homographic puns is consistently lower than that for heterographic and homophonic puns. Compared to phonetic ambiguity caused by near-homophones, the result indicates that ASR models are more robust in handling words that involve only semantic ambiguity without introducing substantial phoneme confusion; in contrast, for puns with pronounced audio-level ambiguity (such as heterographic and homophonic puns), existing transcription models still have considerable room for improvement.

C.4 Time Consumption Analysis

We statistically analyze the time consumption of cascaded systems and end-to-end speech models in audio pun dataset, as shown in Table 10. To prevent the influence of additional factors such as network latency, we only conduct experiments on local open-source LALMs and cascaded systems on real-world datasets. The experimental setup used an A100-SXM4-80GB, and we select MiniCPM and Qwen-2.5-omni as LALMs, along with a combination of four cascaded systems. Experimental results show that compared to end-to-end speech models, cascaded systems require longer inference time when processing puns. This is mainly due to the fact that cascaded systems need to sequentially execute the speech transcription and text processing stages, introducing additional computational overhead. In contrast, LALMs reduce the overall inference latency by modeling the speech-to-semantic mapping end-to-end. This result demonstrates that using LALMs for speech pun understanding has a clear efficiency advantage.

D Appendix D: Annotation Prompts

We use the following prompt to instruct Claude-Opus (Claude-Opus-4-1) to identify and locate puns in transcribed speech. The prompt includes a clear task description, output specification, and a few manually annotated examples with corrective feedback to improve the accuracy of the auxiliary annotation. Some details are shown as Figure 5 and Figure 6.

Model	Architecture	ASR (s)	Reasoning (s)	Total (s)
MiniCPM	End-to-End	-	0.61	0.61
Qwen-2.5-omni	End-to-End	-	0.91	0.91
SenseVoiceSmall+MiniCPM	Cascaded	0.42	0.54	0.74
Whisper-Large+MiniCPM	Cascaded	1.31	0.54	1.85
SenseVoiceSmall+Qwen-2.5-Instruct	Cascaded	0.42	0.34	0.76
Whisper-Large+Qwen-2.5-Instruct	Cascaded	1.31	0.34	1.65

Table 10: Time consumption analysis in real-world dataset between open-source LALMs and cascaded systems.

Model	Pun Recognition		Pun Location		Pun Inference		
	Acc. (↑)	Pun (%)	Coa (↑)	Fin (↑)	Heg (↑)	Hog (↑)	Hop (↑)
SALMONN	63.33	50.00	43.33	23.33	18.33	28.33	18.33
Qwen2-Audio-Instruct	51.66	43.33	58.33	33.33	30.00	23.63	28.33
Audio-Reasoner	58.33	50.00	63.33	40.00	31.67	34.61	35.42
Qwen-2.5-omni	78.33	58.33	60.00	36.67	43.33	31.66	65.00
MiniCPM	53.33	33.33	53.33	30.00	46.67	44.06	58.33
Omni-R1	75.00	65.00	56.67	33.33	45.00	44.82	63.89
MERaLiON2	73.33	**86.67**	56.67	26.67	51.67	36.66	65.00
Gemini 2.0 Flash ✯	**81.67**	63.33	70.00	41.66	51.67	53.33	81.67
GPT4o-Audio ✯	73.33	86.67	73.33	46.67	53.33	52.54	78.33
Human	76.67	41.67	**76.67**	**60.00**	**78.33**	**73.33**	**90.00**

Table 11: Comparison with Human Evaluation and LALMs. ✯ represents the proprietary LALMs.

E APPENDIX E: HUMAN INVOLVEMENT

E.1 ANNOTATION

For real-world data recordings, we recruit 10 adult anonymous participants with strong English-speaking backgrounds, all of whom hold a college education and represent diverse majors from different countries. To ensure diversity in speech, each participant records approximately 30–60 pun sentences, and we provide a compensation of $0.3 per sentence recording.

E.2 HUMAN VERIFICATION

Human verification is carried out by two experts, each with a master's or doctoral degree and strong English proficiency. The review process is divided into two parts, including both synthetic dataset and real-world dataset:

Audio Quality: Experts examine the audio files to identify any missing or under-recorded segments and to ensure that the recordings are clear and intelligible. Each audio clip averages no more than 8 seconds in length. To preserve the authenticity of natural speech, accent variation is not restricted. For quality assessment, we randomly sample 300 audio files from the dataset. Approximately 1.5% of the samples require re-recording due to defects, yielding a low overall defect rate and confirming the high quality of the audio data.

Annotation Verification: After the initial annotation stage, we conduct a systematic quality check. Experts confirm the presence of puns, specify alternatives for heterographic and homophonic puns, and validate the accuracy of dual meanings in homographic puns. A random sample of 300 data points is selected for cross-validation. On average, reviewers spend 45–60 seconds evaluating each speech sample. Each entry is independently verified by each expert, and inter-annotator agreement is measured using *Cohen's K*, which reaches 0.92, indicating substantial consistency across annotators.

E.3 HUMAN EVALUATION

We randomly sample 200 data points for manual evaluation from each stage to compare the performance of current mainstream LALMs with that of human participants. Task-specific questionnaires are designed and distributed to three anonymous volunteers via the scientific research platform Prolific [3], targeting primarily native English speakers from the UK and the USA. Each participant receives a reward of $5.4 per completed questionnaire. Table 11 presents the evaluation results, indicating that while some state-of-the-art LALMs achieve performance comparable to human evaluators in certain pun understanding tasks, they still lag significantly behind humans in others, such as pun inference. In addition, we observe that human evaluators exhibit a conservative bias when identifying puns, tending to classify utterances as non-puns. However, it is worth noting that pun understanding relies on rich linguistic and world knowledge, representing a high-level language comprehension task. Therefore, this task calls for larger-scale and more diverse human evaluations, which we plan to pursue in future work.

F APPENDIX F: THE ROLE OF ACOUSTIC CUES

We first design a controlled comparison on the real-world subset under different speaker conditions. Specifically, we randomly select two speakers (one male and one female) and evaluate them separately on the pun location task. Table 12 show consistent performance differences between the two speakers, indicating that model behavior varies with acoustic factors such as voice timbre and pronunciation characteristics. This finding suggests that understanding spoken puns does not rely solely on textual transcription but is indeed related to audio-level features. Based on this consideration, during data construction we introduce multi-speaker speech (covering different genders, timbres, and speaking styles) and include both synthetic and real-world audio sources to enhance acoustic diversity and better reflect real-world application scenarios.

We include the cascaded system as a baseline to reveal the capability gap between current LALMs and the "ASR+LLM" pipeline. As shown in Table 2, the cascaded system achieves better overall performance. This result indicates, on the one hand, that mature ASR systems have an advantage in word-level transcription accuracy. On the other hand, it suggests that current end-to-end LALMs do not yet fully exploit acoustic information when processing audio signals and still show notable weaknesses in resolving ambiguities that rely on acoustic cues.

G APPENDIX G: EVALUATION DETAILS

During the pun inference phase, results for heterographic and homophonic puns are evaluated by exact matching between the model-predicted pun words and the ground-truth annotations, as illustrated in the formula below:

$$\text{Eval}(p) = \begin{cases} 1, & \hat{w} = w^*, \\ 0, & \text{otherwise}, \end{cases} \tag{4}$$

where $\hat{w}$ is the model-predicted pun word and w^* is the ground-truth alternative words. For homographic puns, as described in Dataset Construction, we first construct a gold standard capturing the dual meanings of each pun. The model's predicted meanings are then compared with this gold standard using similarity matching, and the two pairs with the highest similarity are selected. The formula is illustrated as below:

$$\text{Eval}(p) = \begin{cases} 1, & S(\hat{m}_{j_1}, m^*_{i_1}) > \tau \ \wedge \ S(\hat{m}_{j_2}, m^*_{i_2}) > \tau, \\ 0, & \text{otherwise}. \end{cases} \tag{5}$$

where m^*_i are the gold-standard meanings of a homographic pun, $\hat{m}_j$ are the model-predicted meanings, $S(\cdot, \cdot)$ denotes the similarity function, and τ is the threshold. Based on empirical experiments, pairs with similarity scores above 0.3 (τ = 0.3) are retained and regarded as positive examples for this task.

[3] https://www.prolific.com/

/* Definition */
You are an expert linguist with specialized knowledge of pun research. Your role is to annotate transcribed spoken sentences by identifying the pun word. A pun word is a word or phrase that exploits multiple meanings, sound similarities, or semantic ambiguities to create humor or wordplay.

/* Instruction */
For each given sentence, identify the pun word. If there is no pun word, output "None". Only output the pun word (or the minimal pun phrase), without explanations or additional text. Be consistent with the format shown in the examples.

/* Examples */
Pun Sentence: Dentists don't like a hard day at the orifice.
Output: orifice
Pun Sentence: A discussion of digging a new mine shaft was too deep for him.
Output: deep
...

/* Test Data */
Pun Sentence: He didn't tell his mother that he ate some glue. His lips were sealed.
Output:

Figure 5: Prompts used to guide Claude-Opus-4-1 in pun location for auxiliary annotation.

/* Definition */
You are an expert linguist with specialized knowledge of pun research. Your role is to annotate transcribed spoken sentences by inferring the alternative word that the pun word replaces or plays upon. A pun word is a word or phrase that exploits multiple meanings, sound similarities, or semantic ambiguities to create humor or wordplay.

/* Instruction */
For each given pun sentence, infer the intended alternative word. Only output the alternative word (or minimal phrase) without explanations or additional text. Ensure consistency with the format shown in the examples.

/* Examples */
Pun Sentence: Dentists don't like a hard day at the orifice.
Pun word: orifice
Output: office
...

/* Test Data */
Pun Sentence: Geologists can be sedimental about their work.
Output: sedimental
Output:

Figure 6: Prompts used to guide Claude-Opus-4-1 in pun inference for auxiliary annotation (for heterographic and homophonic puns).

Model	Speaker1 (Coa)	Speaker1 (Fin)	Speaker2 (Coa)	Speaker2 (Fin)
MiniCPM	57.25	37.50	50.00	28.26
Qwen2.5-Omni	58.00	40.00	52.17	32.60

Table 12: Performance comparison of MiniCPM and Qwen2.5-Omni on coarse-grained (Coa) and fine-grained (Fin) pun word location across two speakers.

H APPENDIX H: TRAINING DETAILS

We fine-tuned MiniCPM-o-2.6 using the TRL framework von Werra et al. (2020) with LoRA adapters. For SFT, we trained for one epoch using a learning rate of 1×10^{-5}, a per-device batch size of 1, and 8 gradient accumulation steps, resulting in an effective batch size of 8. For DPO, we used two training epoch, a learning rate of 5×10^{-7}, and $\beta = 0.02$, while retaining the same batch configuration. LoRA rank, scaling factor, and dropout were set to 16, 32, and 0.05, respectively. Training used BF16 precision and gradient checkpointing. The dataset was randomly divided into training and validation subsets at an 80:20 ratio using seed 42, and the same validation IDs were retained across model variants to ensure comparability. Hyperparameters and prompts were fixed before training rather than selected through validation-set search. Checkpoints were saved every 100 optimization steps, and the final checkpoint after one epoch was used for evaluation; no best-checkpoint selection based on validation performance was performed.