

---

# EmergentTTS-Eval: Evaluating TTS Models on Complex Prosodic, Expressiveness, and Linguistic Challenges Using Model-as-a-Judge

---

Ruskin Raj Manku   Yuzhi Tang   Xingjian Shi   Mu Li   Alex Smola

Boson AI, Santa Clara, CA 95054  
{ruskin, yuzhi, xingjian, mu, smola}@boson.ai

## Abstract

1       Text-to-Speech (TTS) benchmarks often fail to capture how well models handle  
2       nuanced and semantically complex text. Building on *EmergentTTS*, we introduce  
3       *EmergentTTS-Eval*, a comprehensive benchmark covering six challenging TTS  
4       scenarios: emotions, paralinguistics, foreign words, syntactic complexity, complex  
5       pronunciation (e.g. URLs, formulas), and questions. Crucially, our framework auto-  
6       mates both test-generation and evaluation, making the benchmark easily extensible.  
7       Starting from a small set of human-written seed prompts, we iteratively extend  
8       them using LLMs to target specific structural, phonetic and prosodic challenges,  
9       resulting in 1,645 diverse test samples. Moreover, we employ a model-as-a-judge  
10      approach, using a Large Audio Language Model (LALM) to assess the speech  
11      across multiple dimensions such as expressed emotion, prosodic, intonational,  
12      and pronunciation accuracy. We evaluate state-of-the-art open-source and propri-  
13      etary TTS systems, such as 11Labs, Deepgram, and OpenAI’s 4o-mini-TTS, on  
14      EmergentTTS-Eval, demonstrating its ability to reveal fine-grained performance  
15      differences. Results show that the model-as-a-judge approach offers robust TTS  
16      assessment and a high correlation with human preferences. We open source the  
17      evaluation code<sup>1</sup> and the dataset<sup>2</sup>

## 18   1 Introduction

19   Recent breakthroughs in generative modeling have led to significant advancements in Text-to-  
20   Speech (TTS) technology [6, 43, 18, 40, 14]. State-of-the-art proprietary systems now demonstrate  
21   remarkable naturalness and human-like quality when converting standard, well-formed text into  
22   spoken language. These systems are widely deployed in various applications, including virtual  
23   assistants [25], audiobooks [37, 26], navigation systems [20, 36], and accessibility tools [28, 24].  
24   However, as TTS technology becomes more integrated into real-world use cases, systems increasingly  
25   encounter complex and diverse text prompts that go beyond conventional reading tasks, such as code  
26   switching, or rendering complex technical character sequences.

27   Conversely, evaluation methodologies for TTS systems have not kept pace with the growing complex-  
28   ity of use cases. Current benchmarks exhibit several limitations: they often use restricted text domains  
29   [41], the lack diversity in linguistic phenomena [38], and they rely on costly, non-reproducible human  
30   evaluations that may vary significantly across different listener cohorts. Even worse, code-switching  
31   in multiple languages requires extremely polyglot evaluators (or many specialized ones). Thus, for  
32   reasons of practicality, many evaluations focus on voice cloning alone.

---

<sup>1</sup><https://github.com/boson-ai/EmergentTTS-Eval-public>

<sup>2</sup><https://huggingface.co/datasets/bosonai/EmergentTTS-Eval>

Real-world TTS applications encounter numerous challenges that remain difficult for current systems. These include accurately reflecting human emotions and sounds [8]-for example, when narrating various types of books like fantasy or children’s literature, TTS systems must realistically handle quoted dialogues and paralinguistic cues to keep listeners engaged. Another dimension involves more formal scenarios, such as syntactically complex text with nested clauses in legal and literary contexts, or scientific and academic texts containing special characters and equations that are difficult to pronounce. Additionally, there is a growing need for TTS systems to handle multilingual content [22, 11, 32] and properly intonate questions with contextually appropriate prosody [19], challenges that current evaluation frameworks fail to systematically address. An evaluation methodology is required that reliably captures TTS performance across all these scenarios, moving away from subjective human assessments of expressiveness and prosody.

To this end, we propose EmergentTTS-Eval, a comprehensive benchmark specifically designed to evaluate TTS performance across these challenging scenarios. Our benchmark covers six critical dimensions. Through an iterative refinement process, we are able to controllably generate increasingly more difficult utterances for TTS systems to synthesize. Furthermore, drawing parallel from the textual domain, where reward LLMs are widely used for judging output of other LLMs, we propose to use the model-as-judge paradigm for evaluating TTS systems. Our contributions are as follows:

- We create a benchmark with 1,645 samples for evaluating TTS systems across six challenging scenarios: Emotions, Paralinguistics, Syntactic Complexity, Questions, Foreign Words and Complex Pronunciation.
- We propose an iterative refinement strategy with LLMs that creates increasingly complex utterances for TTS, resulting in a multi-layered and diverse evaluation benchmark for evaluating all aspects of TTS performance.
- We are the first to use Large Audio Language Models (LALMs) as reward models for judging otherwise subjective dimensions of audio, like expressiveness, prosody, pausing, stress and pronunciation accuracy. We show its effectiveness through human correlation. The results are stable under the choice of judge LALMs.
- We evaluate leading open-source and closed-source TTS systems on our benchmark, showing how model-as-a-judge reveals finer-grained and systematic failures, and highlights the gap between closed-source and open-source models on specific aspects of speech generation.

## 2 Related Work

### 2.1 Text-to-Speech Model Evaluation Metrics

Traditional TTS evaluation rely on humans to provide a Mean Opinion Score (MOS) that is both costly and statistically noisy, due to its reliance on a changing pool of evaluators. Recent advances in automatic TTS evaluation typically rely on two metrics: the Word Error Rate (WER) [6, 43, 18, 14], as calculated by using an ASR model to convert the generated speech back into text and compare with the reference text; a speaker-similarity score (SIM) [6, 43, 18, 40], calculated by comparing the latent embeddings of generated vs. reference speech using an audio foundation model, such as WaveLM [15]. Recent works also explored the use of models to directly predict MOS (Sim-MOS) by training on datasets such as The Samsung Open MOS Dataset [23] and The VoiceMOS Challenge [16].

While these metrics serve to capture how natural or accurate a system sounds, their evaluation power is limited by the difficulty and expressivity of the voice dataset and cannot handle nuanced, context-sensitive phenomena such as emotional prosody or complex syntax. More recently, BASE-TTS [18] introduced an emergent abilities test suite that probes seven linguistically motivated phenomena, such as compound nouns, emotions, foreign words, paralinguistics, etc., using 20 hand-crafted prompts per category. Although BASE-TTS shifts the focus toward higher-order TTS capabilities, its dataset size is limited and reliance on human expert judges is costly.

Our work addresses these limitations by automating test-generation and expanding category coverage. In particular, we create progressively harder stimuli at scale to differentiate between high-performing TTS systems. Our framework thus bridges the gap between traditional metric-based evaluation and nuanced, reproducible benchmarking.

## 84 2.2 Model-as-a-Judge For Text-to-Speech Model Evaluation

85 A key weakness in previous benchmarks is the need for human judges. Recent years have seen a  
86 growing trend of integrating audio encoders with LLMs. This has resulted in large audio-language  
87 models (LALM) that excel at a variety of audio comprehension tasks [15, 35, 29, 34, 12]. SALMONN  
88 [34] uses finetuned LALM to predict MOS, SIM and A/B testing scores. Wang et al. [39] extends  
89 this by finetuning an LALM to also generate open-ended qualitative feedback, covering noisiness,  
90 distortion, prosody, etc., alongside scores. This approach leverages the LLM’s contextual knowledge  
91 to provide multi-dimensional evaluations more akin to a human expert. Chen et al. [10] compiled the  
92 first corpus of human-written TTS evaluations (with overall MOS plus detailed error annotations) and  
93 used it to train an audio-augmented GPT model. The resulting system can describe speech quality  
94 degradations and compare two samples in free-form language and outperforms prior state-of-the-art  
95 MOS prediction models. WavReward [17] employs a generalist reward model to score spoken  
96 dialogue quality across dimensions like clarity and expressiveness .

97 Our work not only use LALMs as judges but also to generate tests spanning categories of emergent  
98 TTS abilities. Our evaluation demonstrates that even out-of-the-box LALMs like Gemini-2.5 Pro are  
99 capable of evaluating emergent capabilities in SOTA TTS systems and produce A/B testing results  
100 that are highly-correlated with human preference.

## 101 3 EmergentTTS-Eval Benchmark

102 In this section, we describe how we construct the datasets in EmergentTTS-Eval, which covers  
103 6 categories of challenging real-world TTS scenarios with varying levels of complexity. We also  
104 describe how the evaluation process is scaled with the help of Large Audio Language Model (LALM).

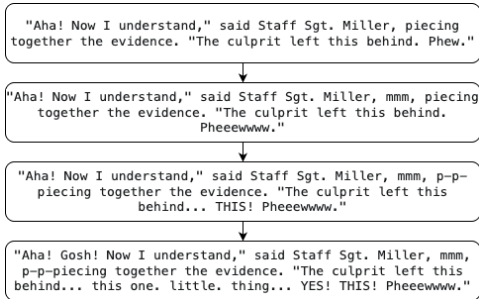


Figure 1: Paralinguistic example, refined and made more complex for TTS with increased number of cues.

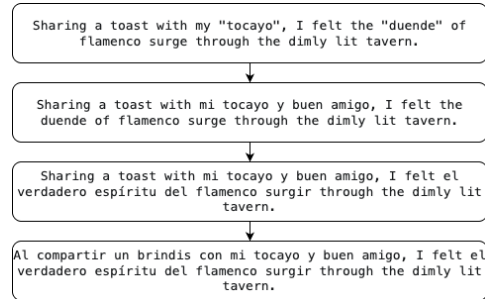


Figure 2: Foreign words example, refined and made more complex by adding idiomatically and prosodically rich foreign words.

### 105 3.1 Dataset Construction

106 We follow two key guidelines when constructing text prompts in EmergentTTS-Eval: (1) the  
107 dataset should encompass real-world challenges faced by TTS systems, and (2) it should exhibit  
108 varying levels of difficulty to enable fine-grained assessment of system capabilities. To this end,  
109 we begin with a diverse set of seed prompts and iteratively expand their scope (breadth) and  
110 complexity (depth). This reflects the approach of progressively increasing instruction difficulty used  
111 in instruction tuning [42]. Our seed prompts are derived from a collection of 140 human curated  
112 samples introduced in the BASE-TTS [18]. These samples span 7 challenging TTS categories  
113 commonly encountered in real-world scenarios: “Compound Nouns”, “Emotions”, “Foreign Words”,  
114 “Paralinguistics”, “Questions”, “Syntactic Complexities”, and “Punctuation”, with 20 samples per  
115 category. Some prompts pose challenges for TTS systems because generating realistic speech  
116 requires a deep understanding of the text’s semantic context. Consider the following text prompt from  
117 the “Emotions” category: A profound sense of realization washed over Beal as he  
118 whispered, "You’ve been there for me all along, haven’t you? I never truly  
119 appreciated you until now.". An effective TTS system should recognize the emotional  
120 context and appropriately render the quoted sentence as a whisper to reflect Beal’s sentiment.

Although the BASE-TTS proposed set is of high quality, it is limited in its ability to explore the depth within each category and lacks broad diversity, as it was curated by a small group of individuals. However, complexity and diversity are essential for a robust evaluation benchmark, as they help assess challenging scenarios where system failures can significantly impact user experience. For example, we want to evaluate on real-world scenarios like sequential interrogative questions, sustained emotion synthesis with natural shift to contrasting emotions, multi-code switching, etc. In addition to the categories defined in BASE-TTS, we introduce a new category called “Complex Pronunciation”, which contains prompts featuring unusual characters, numerals, and tongue-twisters. We exclude the “Compound-Nouns” category due to its limited scope and the strong performance of current TTS systems according to manual assessment. We also drop the “Punctuations” category, as punctuation-related challenges are inherently addressed within other categories such as “Paralinguistics”, “Syntactic Complexity”, and “Complex Pronunciation”.

To enrich the complexity of the text prompts and improving their diversity, we leverage LLMs to iteratively refine the initial utterances. The LLM is first tasked to curate samples that improve the dataset breadth-wise, guided by explicit diversity enhancement criteria embedded in the prompt and reinforced by strict structural constraints. Afterwards, we apply an iterative refinement process to construct a multi-tiered dataset encompassing utterances of varying complexities. In the process, we take the base utterance  $U_i$ , and create a deeper version  $U_{i+1}$  through a specific refinement method for each category.  $U_{i+1}$  can then be fed as input to the next refinement step, to get an even more challenging  $U_{i+2}$  and so on. According to our experiments, the LLMs are able to generate strong refinements if we provide detailed criteria in the instruction and three refinement steps are sufficient. We share the prompts we use for all the categories in the Appendix A, and an example refinement for Paralinguistics and Foreign Words category is shown in Figures 1 and 2, respectively. Here are the description of the six categories in the final dataset:

- Questions:** Contain sequential questions and statements. This evaluates the TTS system’s ability in generating interrogative and declarative prosody.
- Emotions:** Contains narrative text with long quoted dialogues of emotion intensification, followed by contrasting emotions.
- Paralinguistics:** Contains vocal interjections (Uhh, Hmmm), Onomatopoeia (Achoo, tick-tock, etc), Varied Emphasis markers (Capitalization, vowel elongation, syllable emphasis with hyphens), Pacing cues like ellipses (...) or punctuation (STOP.RIGHT.THERE), and stuttering (I-I-I d-didn’t, so so-so-so-sorry).
- Foreign Words:** Covers 15 unique languages with idiomatically and prosodically rich phrases placed in between english text.
- Syntactic Complexity:** Covers complex text with garden-path sentences, deep nested clauses with centre embeddings, homographs, and other forms of syntactic complexity.
- Complex Pronunciation:** Texts with emails, phone numbers, URLs, Street Addresses, Location references, STEM equations, units and notations, Abbreviations-Both initialisms (pronounced letter by letter) and acronyms (pronounced as word), and tongue twisters.

**Dataset Statistics:** For five of the categories that we use from BASE-TTS, we curate a total of 70 seed utterances by appending 50 breadth-wise expanded sentences along with 20 curated by human, after this, we perform three iterative refinement steps, resulting in additional  $70 * 3 = 210$  samples. This results in 280 samples each for these five categories. For “Complex Pronunciation”, we curate 60 breadth-wise diverse samples from scratch, which are turned into 240 samples after three rounds of refinement. Subsequently, we add five complex short tongue twisters, each repeated multiple times. Based on our manual observation, TTS systems often struggle with repeated articulation-where a single slip can lead to a cascading effect. We report these findings in Section 4.2. Total sample count thus comes out to  $280 * 5 + 240 + 5 = 1645$ . Category-wise statistics are shown in the Appendix A.7.

### 3.2 Large-Audio-Language Model as Judge

Synthesizing speech for all 1,645 benchmark utterances results in approximately 420 minutes (or 7 hours) of audio per TTS system. Evaluating this volume of audio through human raters is both time-consuming and resource-intensive, with limited reproducibility, and the need for specialized linguistic knowledge. To address these limitations, we employ Large-Audio-Language Models (LALMs) as automatic judges. Our benchmark specifically targets aspects of speech synthesis-such as prosody, pausing, expressiveness, and pronunciation-that are not adequately captured by traditional metrics like

Word Error Rate (WER) or MOS-based quality estimators. Accurately assessing these dimensions requires a general-purpose, high-capacity audio understanding model.

We choose **Gemini 2.5 Pro** as our primary LALM-based judge due to its strong performance on established audio reasoning benchmarks such as MMAU [30] (See Appendix B for performance comparison of different LALMs on MMAU). Notably, it leverages inference-time scaling [13, 21] before producing outputs, which aligns well with the complexity of our evaluation tasks.

To evaluate a candidate TTS system  $T_i$ , we compare it against a strong reference system  $T_j$ , chosen to have low WER to ensure high-fidelity synthesis of the benchmark utterances. For each evaluation instance, both systems generate speech for the same input, and the outputs are randomly assigned as  $T_1$  and  $T_2$  to avoid positional bias. The LALM judge is provided with the original text, the associated category label, and a structured evaluation prompt that includes the target evaluation dimension (e.g., prosody, emotion), scoring rubric, and detailed category-specific reasoning guidelines. The model is then presented with the audio from  $T_1$ , followed by a separation marker, and then the audio from  $T_2$ .

The LALM returns a structured json response containing natural language justifications for the performance of each system, a comparative analysis highlighting key differences-annotated as either subtle or significant-a scalar score in the range  $[0, 3]$  for each system, and a final winner label: 0 for a tie, 1 if  $T_1$  is preferred, and 2 for  $T_2$ . The prompt is designed to elicit chain-of-thought reasoning with time-stamp based analysis, and encourages the model to resolve borderline cases by articulating fine-grained distinctions and predict human-based preferences. The full judge prompts used for each category are shared in the Appendix C.3.

We adopt a win-rate-based metric to summarize performance. Let  $W(T_i)$  denote the win-rate of system  $T_i$  relative to the baseline  $T_j$ . This is computed as:

$$W(T_i) = \frac{\sum(\text{winner} = \text{index}_i) + 0.5 \cdot \sum(\text{winner} = 0)}{n}$$

where  $\text{index}_i \in 1, 2$  corresponds to the randomized label assigned to  $T_i$ , and  $n$  is the total number of comparisons. A score of 0.5 reflects parity with the baseline, while deviations indicate relative superiority or inferiority.

This evaluation protocol enables robust, interpretable, and reproducible TTS comparison at scale. Unlike human raters, the LALM offers consistent judgments across multilingual and prosodically rich utterances, and its outputs include timestamp-grounded rationales that support fine-grained diagnostic analysis as evidenced by examples provided in the Appendix E. Our experiments in Section 4.6.2 show that the judge-based win-rate has high correlation with human preference as well.

## 4 Experiments

### 4.1 Setup

**Models Evaluated** We evaluate three open-source models: **Sesame-1B (TTS)** [33], **Qwen2.5 Omni (LALM)** [43], and **Orpheus-TTS (TTS)** [9]. In addition, we benchmark four closed-source systems using their flagship models: **ElevenLabs’ multilingual-v2 (TTS)**, **Deepgram’s Aura-2 (TTS)**, **HumeAI’s Octave (TTS)**, and OpenAI’s **GPT-4o** suite, which includes both TTS and audio reasoning variants-gpt-4o-mini-tts, gpt-4o-audio-preview-2024-12-17, and gpt-4o-mini-audio-preview-2024-12-17. For models that are fine-tuned on specific voices, we pre-select some of these voices to show the main results. As we show later in Section 4.4, the final win-rate can be sensitive to the voice. We considered other open-source models as well, the results for which we report in the Appendix D. In addition to the win-rate as described in Section 3.2, we follow standard practice by computing WER using **Whisper-v3-large** [27], and MOS scores are calculated using a fine-tuned wav2vec2.0 model [7].

**Prompting** For pure TTS models such as Sesame1B, Orpheus-TTS, Aura-2, and Eleven Multilingual v2, we directly provide the utterance text. For other models, we compare a basic prompting setup (utterance only) with a **Strong Prompting** strategy, where the input is augmented with category-specific instructions (e.g., “be emotionally expressive” for the Emotions category). For HumeAI and GPT-4o-mini-tts, these instructions are passed via optional style descriptors; for LALMs like Qwen 2.5 Omni and GPT-4o audio variants, they are included in the user message.

We calculate the win-rate of all evaluated models against gpt-4o-mini-tts(Alloy voice), judge temperature is set to 0.0. More details about hyper-parameters for the judge and evaluated models, along with the full prompting templates used to generate audios are provided in the Appendix C.

## 4.2 Benchmark Performance

**Overall Results:** Model performance, summarized in Table 1, reveals a broad spectrum of win-rates ranging from 15.96% to 65.17%. GPT-4o-Audio (Ballad voice) achieves the highest overall performance, with particularly strong results in the expressiveness-focused categories-88.84% in *Emotions* and 82.14% in *Paralinguistics*. Notably, only GPT-4o-mini-tts with strong prompting surpasses the 50% win-rate in the *Complex Pronunciation* category, suggesting targeted optimization by OpenAI for this capability. HumeAI ranks as the second-best closed-source system, outperforming Deepgram’s Aura-2 (Thalia) and ElevenLabs’ Multilingual v2 (Brian). The low performance of Aura-2 in multilingual settings aligns with its lack of explicit multilingual support; when the *Foreign Words* category is excluded, its win-rate rises to approximately 35%, slightly above ElevenLabs. Among open-source models, Orpheus-TTS performs best, with Qwen 2.5 Omni following closely. In contrast, Sesame1B exhibits significant performance deficits, particularly in the *Emotions* and *Complex Pronunciation* categories. We observe that **strong prompting** consistently enhances performance for all models where both prompted and unprompted evaluations are available. For example, GPT-4o-mini-tts reaches a 56% win-rate under strong prompting, showing a clear improvement over its baseline configuration. A similar gain is observed for GPT-4o-audio-preview. Win rates and MOS scores show low correlation (Spearman Rank Correlation = 0.23), demonstrating they capture distinct aspects of speech quality. Judge parsing failures stemmed from two issues: incorrect JSON formatting and reaching maximum token limits when judges became trapped in repetitive reasoning loops.

**Depth-wise Performance Trends:** Figure 3 illustrates how model win-rates change across increasing refinement depths for each category. Models naturally cluster into high-performing (average win-rate > 50%) and low-performing groups. Although we might expect deeper utterances to widen this performance gap-with strong models excelling and weaker ones faltering-our findings reveal more nuanced patterns. At higher complexity levels, both models may encounter difficulties, increasing the likelihood of ties. Additionally, strong models sometimes reveal systematic weaknesses when challenged by greater complexity, while lower-performing models occasionally match or exceed the baseline by avoiding specific failure modes. Nevertheless, four of our six categories exhibit clear depth-sensitive performance trends. The exceptions are *Questions* and *Syntactic Complexity*, where more subtle prosodic expectations result in less pronounced differentiation across depths.

**Systematic Failures and Judge Insights:** Depth-wise analysis reveals consistent failure patterns and demonstrates our judge’s sensitivity to prosodic, phonetic, and semantic mismatches. Most open-source models handle *Questions* and *Syntactic Complexity* adequately, with Sesame1B being the notable exception due to flat intonation and poor pausing. Sesame1B particularly struggles with *Emotions*, often inserting random interjections or producing monotonous speech. All open-source models underperform on *Complex Pronunciation*, misreading decimals, dropping digits, and breaking down at higher complexity. For *Foreign Words*, Sesame substitutes non-English tokens with unrelated content, while Orpheus anglicizes pronunciation to the extent of being phonetically incorrect.

Commercial models show different limitations: ElevenLabs falters with *Complex Pronunciation*, while Deepgram Aura-2 degrades with longer utterances and struggles with expressive *Paralinguistics*. OpenAI models excel in emotional and multilingual content but still exhibit subtle issues-occasional mispronunciations, dropped dates, and synthesis breakdowns-that our judge successfully identifies. The judge effectively distinguishes emphatic renditions, recognizes homograph disambiguation, and rewards appropriate prosody, though subtle paralinguistic nuances and emotional shifts remain challenging to evaluate perfectly. We provide detailed analysis of these failure modes and judge behavior in the Appendix E.

## 4.3 Sensitivity to Judge

While Gemini 2.5 Pro achieves the highest performance on the MMAU [30] benchmark for audio understanding, we conducted an ablation study to assess how evaluation outcomes vary across

Table 1: Main results, WER↓ and Win-rate↑ over all categories with gpt-4o-mini-tts-allow as baseline, † represents Strong Prompted models

Model	Voice	Emotions		Foreign Words		Paralinguistics		Complex Pronunciation		Questions		Syntactic Complexity		Overall			
		WER	Win-Rate	WER	Win-Rate	WER	Win-Rate	WER	Win-Rate	WER	Win-Rate	WER	Win-Rate	WER	Win-Rate	Parsing Fail	MOS
gpt-4o-mini-tts(baseline)	Alloy	0.72	-	13.45	-	20.55	-	29.90	-	0.42	-	1.04	-	10.61	-	-	4.23
Sesame1B	-	17.07	7.32%	45.27	10.35%	49.63	18.92%	80.97	7.40%	2.74	31.78%	4.30	18.88%	32.32	15.96%	4	3.38
Qwen 2.5 Omni	Chelsie	1.22	41.18%	26.98	11.07%	57.48	17.44%	64.07	3.30%	12.77	49.28%	1.66	36.96%	26.58	27.07%	7	4.09
Qwen 2.5 Omni†	Chelsie	2.41	41.60%	26.77	11.42%	58.44	20.25%	49.51	6.12%	0.87	51.78%	3.47	38.57%	23.03	28.77%	1	4.09
Orpheus TTS	Tara	1.81	31.78%	22.31	17.5%	40.94	39.82%	41.04	10.61%	1.48	39.64%	1.63	38.92%	17.71	30.12%	0	3.76
DeepGram Aura-2	Thalia-en	3.45	29.28%	21.41	18.75%	23.73	21.14%	54.49	33.81%	1.24	48.21%	1.36	43.70%	16.83	32.44%	4	4.33
11Labs eleven multilingual v2	Brian	0.63	30.35%	14.44	35.53%	21.51	45.53%	31.44	14.48%	0.49	39.46%	1.15	35.53%	11.19	33.89%	0	3.55
HumeAI†	-	0.83	61.60%	21.05	34.64%	19.84	36.91%	37.14	34.28%	0.38	43.21%	0.93	44.64%	12.85	42.73%	1	4.18
gpt-4o-mini-tts†	Alloy	0.71	59.17%	12.07	57.32%	21.33	58.75%	31.57	52.44%	0.66	52.67%	0.84	57.14%	10.76	56.32%	2	4.20
gpt-4o-mini-audio-preview	Alloy	0.95	55.89%	14.48	59.82%	19.04	52.86%	32.27	30.61%	55.27	47.32%	0.88	48.75%	10.92	49.60%	1	4.19
gpt-4o-mini-audio-preview†	Alloy	9.34	59.13%	12.70	58.92%	20.92	62.59%	37.14	28.68%	0.74	48.21%	0.72	53.40%	13.09	52.31%	5	4.18
gpt-4o-audio-preview	Alloy	1.03	48.57%	14.72	60.17%	23.16	66.78%	35.89	40.81%	1.19	47.5%	1.25	57.14%	12.38	53.76%	0	4.09
gpt-4o-audio-preview†	Alloy	0.93	61.64%	13.75	62.5%	20.56	68.21%	36.92	49.59%	1.72	47.85%	1.26	56.85	12.00	57.95%	4	4.06
gpt-4o-audio-preview†	Ballad	1.82	88.84%	13.30	60.17%	21.15	82.14%	35.32	40.40%	1.38	56.96%	1.16	59.53%	11.87	65.17%	4	3.83

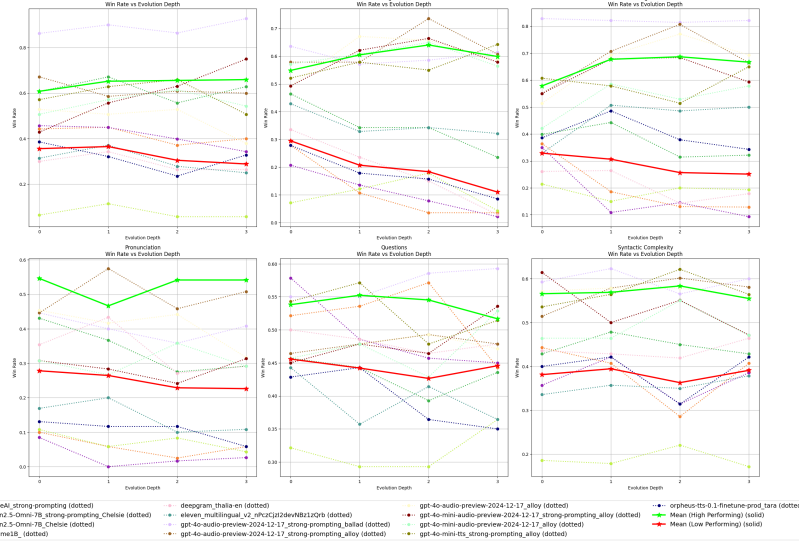


Figure 3: Win-rate chart for each category at different refinement depths. We also show the mean win-rate at each depth, computed collectively for high-performing models (average win-rate>50%) and low-performing models (average win-rate<50%).

Table 2: Win-rates based on judger used, † represents Strong Prompted Models

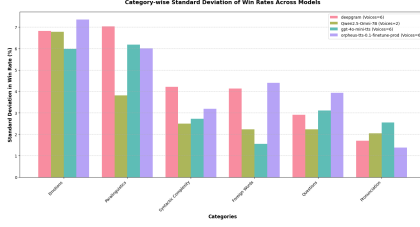
Judger Model → Evaluated Model ↓	Gemini 2.0 Flash		Gemini 2.5 Flash		Gpt-4o-mini-audio		Gpt-4o-audio		Qwen 2.5 Omni	
	Win-Rate	Parsing Fail	Win-Rate	Parsing Fail	Win-Rate	Parsing Fail	Win-Rate	Parsing Fail	Win-Rate	Parsing Fail
Sesame1B	25.30%	3	24.77%	6	28.60%	2	31.07%	2	41.39%	76
Qwen2.5 Omni Chelsie	38.06%	3	31.49%	8	42.67%	1	38.13%	2	47.12%	82
Qwen2.5 Omni Chelsie†	39.17%	0	32.09%	6	43.09%	2	39.38%	1	47.41%	77
Orpheus-TTS Tara	39.41%	1	38.02%	4	41.03%	0	41.33%	1	48.59%	74
DeepGram Thalia	40.79%	0	36.27%	2	43.10%	0	37.26%	0	47.84%	70
ElevenLabs Brian	44.79%	1	41.14%	0	48.93%	1	44.22%	0	48.98%	67
Hume.AI	47.99%	1	40.34%	3	46.20%	0	47.20%	1	49.42%	76
gpt-4o-mini-tts Alloy†	54.43%	0	53.43%	0	52.06%	0	51.51%	1	50.31%	63
gpt-4o-mini-audio-preview Alloy	48.08%	0	47.14%	0	48.63%	0	48.72%	1	50.28%	71
gpt-4o-mini-audio-preview† Alloy	51.18%	1	49.57%	0	47.29%	1	50.12%	0	49.10%	73
gpt-4o-audio Alloy	53.28%	0	53.65%	2	50.39%	0	53.03%	0	49.71%	81
gpt-4o-audio† Alloy	54.98%	1	57.06%	3	50.54%	0	54.74%	0	<b>50.69%</b>	73
gpt-4o-audio† Ballad	<b>58.78%</b>	1	<b>57.60%</b>	1	<b>55.80%</b>	1	<b>64.23%</b>	1	49.30%	68

different LALM judger models, both proprietary and open-source. Using identical audio inputs from candidate TTS systems, we varied the judger model across four closed-source and one open-source alternative. Results are shown in Table 2.

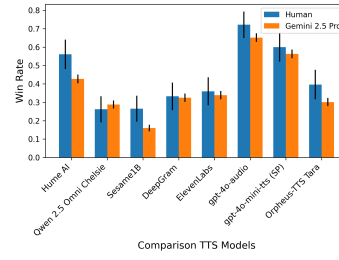
Our analysis reveals that Qwen 2.5 Omni performs poorly in the judging role, frequently producing parsing errors and yielding win-rates near 50% across the board-indicative of near-random behavior. In contrast, the remaining judger models (Gemini 2.0 Flash, Gemini 2.5 Flash, Gemini 2.5 Pro, GPT-4o-mini-audio, and GPT-4o-audio) exhibit strong agreement in their relative rankings, despite differences in absolute scores. This alignment is quantified by a high Kendall’s W coefficient of concordance ( $W = 0.97$ ), indicating near-perfect inter-model consistency and further validating the robustness of our evaluation framework.

#### 4.4 Understanding bias of specific voices

Most TTS models are tied to specific voices-either through fine-tuning or voice cloning-except for a few, such as Hume.AI and Sesame1B, which generate different voice for different utterances.



(a) Variance of win-rate by voice (gemini-2.0-flash as judge)



(b) Comparison of different TTS win-rates against baseline under human vs model (Gemini 2.5 Pro) with 95% CI

Figure 4: Left: Variance of win-rate by voice. Right: Human and model win-rate alignment.

(a) Performance difference for complex pronunciation with normalization techniques using GPT-4o-mini-TTS (Gemini-2.0-Flash judge), averaged over 6 runs.

Text Normalization Method	Win-rate $\uparrow$
No TN	51.69%
WeText TN	50.06%
GPT-4.1-mini TN	76.74%

(b) Spearman correlation between human and model judge rankings based on win-rate.

Model Judge	$\rho_{\text{Spearman}} \uparrow$
Gemini 2.5 Pro	90.5%
Gemini 2.0 Flash	90.5%
Gemini 2.5 Flash	90.5%
GPT-4o-audio	90.5%
Qwen 2.5 Omni	88.1%
GPT-4o-mini-audio	76.2%

Table 3: Left: Ablation study on the impact of different text normalization methods. Right: Correlation between human preference and different judge models.

To examine the impact of voice identity on performance, we measure the category-wise standard deviation in win-rate across multiple voices for four models: GPT-4o-mini-tts (6 voices: Alloy, Ballad, Ash, Coral, Nova, Onyx), Deepgram Aura-2 (6 voices: Thalia-en, Andromeda-en, Helena-en, Apollo-en, Arcas-en, Aries-en), Orpheus-TTS (Tara, Leah, Jess, Leo, Dan, Mia), and Qwen 2.5 Omni (2 voices: Chelsie and Ethan). Results are shown in Figure 4a. We find that *Emotions* and *Paralinguistics* exhibit the highest sensitivity to voice variation, reflected in elevated standard deviations. This is consistent with the fact that voice fine-tuning often emphasizes expressive rendering, which these categories demand. In contrast, *Pronunciation* shows the least variance across voices, as it depends more on the inherent ability of the model and not the voice characteristics, other categories also show low variance generally.

#### 4.5 Text Normalization

The main challenge of the complex pronunciation category lies in parsing uncommon characters and their groups, something that can be made easier by using Text Normalization(TN) techniques prior to sending the text to the TTS model. To this end, we do an ablation measuring the change in win-rate for various TN techniques. We also add the data point corresponding to an LLM (GPT-4.1-mini) acting as the TN, the results are in Table 3a.

We note that basic TN techniques do not always improve model performance on our benchmark and can make it worse. For instance, WeText converts '\$1,890.125375' to 'one thousand eight hundred and ninety point one dollars twenty five thousand three hundred and seventy five', which harms TTS quality. Similarly, '0' is sometimes normalized to the informal 'oh', which is not preferred in formal or decimal contexts. 'SQL' was correctly normalized to 'S Q L', but the baseline's pronunciation 'Sequel' was preferred. Using an LLM for TN resolves many of these issues and significantly improves win-rate, though some errors persist with the basic prompt that we used. We include more examples in the Appendix F.

#### 4.6 Human-Model Alignment

##### 4.6.1 Human Study Setup

We conducted human evaluation to measure the correlation between the model judges' preference to that of human judges. We created an online survey using Gradio, where human judges were presented with pairs of audio clips generated by a baseline TTS and a comparison TTS and instructed



to rate which is the better one (or tie). To ensure consistency in evaluation, participants were given instructions and evaluation criteria adapted from the prompts used for the model judges. The human preferences were then aggregated to compute the win-rate of each comparison model against the baseline, which was compared to the win-rates produced by model judges. For this study, we selected gpt-4o-mini-tts as the baseline and compared it against eight other models: Sesame1B, Deepgram, ElevenLabs, gpt-4o-mini-audio-preview, gpt-4o-mini-tts (SP), Hume AI, Orpheus-TTS Tara, and Qwen2.5-Omni Chelsie. These comparisons were evaluated by the following model judges: Gemini 2.5 Pro, Gemini 2.0 Flash, Gemini 2.5 Flash, GPT-4o-mini-audio, GPT-4o-audio, and Qwen 2.5 Omni.

A total of 512 audio pairs were sampled from these comparisons to ensure coverage across different categories and refinement depths. These were distributed among  $N = 8$  human judges, with each judge assigned between 149 and 150 audio pairs with some redundancy among the judges.

#### 4.6.2 Agreement Between Human and Model Judgements

To evaluate alignment between human and model judgments, we computed the Spearman correlation between the comparison models’ rankings based on win-rates derived from human ratings and those derived from each model judge. As shown in Table 3b, all judges achieved high correlation scores, suggesting that model judges closely mirrors human preferences in determining which TTS system performs better. We also analyzed the individual win rates of each comparison model (vs. the baseline) under both human and model evaluations. As shown in Figure 4b, most model win rates are closely aligned with human judgment (within 95% CI), though discrepancies exist in some cases (e.g., Hume AI, Sesame1B), where the model (Gemini 2.5 Pro) over-estimates performance compared to human preference.

## 5 Limitations and Conclusion

There are two main limitations to our work related to the dataset creation and the LALM-as-judge paradigm. First, LALMs have inherent biases that may manifest in our synthetic dataset, such as preferences for literary language and formal phrasing patterns. For categories like Foreign Words and Syntactic Complexity, refinement level of depth=3 produces grammatically correct but somewhat artificial utterances that occur infrequently in natural communication, but still act as a solid stress-test for TTS systems. Additionally, our multilingual evaluation focuses on Latin transcriptions rather than native character sets, which doesn’t fully capture the challenges of true multilingual TTS. Regarding evaluation, using Gemini 2.5 Pro incurs substantial costs-approximately \$50 per complete TTS system evaluation. Nevertheless, the strong ranking agreement observed across different judge models suggests opportunities for more economical alternatives without significant quality loss. We also observe that evaluating subjective aspects like emotions, prosody, and intonation can occasionally lead to LALM hallucinations, where judges incorrectly identify pronunciation issues. Despite these considerations, EmergentTTS-Eval represents a significant advancement in TTS evaluation methodology by addressing critical gaps in existing benchmarks. Our approach systematically challenges TTS systems across dimensions that conventional metrics overlook, while offering a scalable alternative to resource-intensive human evaluations. The strong correlation between our LALM judges and human preferences validates the approach, while the benchmark’s ability to reveal fine-grained performance differences demonstrates its practical utility for driving progress in creating more human-like synthetic speech.

## 6 Broader Impacts

EmergentTTS-Eval aims to accelerate the development of more expressive, accurate, and inclusive TTS systems, which can greatly benefit accessibility tools and enable more natural conversational interfaces across a variety of applications. However, highly convincing TTS systems could be used to perpetrate fraud or spread disinformation, and LALM judges may perpetuate biases. To mitigate these risks, we encourage pairing TTS systems with deepfake detectors or watermarking and auditing prompt and judge outputs for diverse representation.

## References

- [1] MiniCPM-o 2.6. [https://huggingface.co/openbmb/MiniCPM-o-2\\_6](https://huggingface.co/openbmb/MiniCPM-o-2_6).
- [2] sonoai/bark. <https://github.com/suno-ai/bark>.
- [3] tortoise-tts. <https://github.com/neonbjb/tortoise-tts>.
- [4] wenet-e2e/WeTextProcessing. <https://github.com/wenet-e2e/WeTextProcessing>.
- [5] ZypHra/Zonos-v0.1. <https://huggingface.co/ZypHra/Zonos-v0.1-transformer>.
- [6] Philip Anastassiou, Jiawei Chen, Jitong Chen, Yuanzhe Chen, Zhuo Chen, Ziyi Chen, Jian Cong, Lelai Deng, Chuang Ding, Lu Gao, Mingqing Gong, Peisong Huang, Qingqing Huang, Zhiying Huang, Yuanyuan Huo, Dongya Jia, Chumin Li, Feiya Li, Hui Li, Jiaxin Li, Xiaoyang Li, Xingxing Li, Lin Liu, Shouda Liu, Sichao Liu, Xudong Liu, Yuchen Liu, Zhengxi Liu, Lu Lu, Junjie Pan, Xin Wang, Yuping Wang, Yuxuan Wang, Zhen Wei, Jian Wu, Chao Yao, Yifeng Yang, Yuanhao Yi, Junteng Zhang, Qidi Zhang, Shuo Zhang, Wenjie Zhang, Yang Zhang, Zilin Zhao, Dejian Zhong, and Xiaobin Zhuang. Seed-tts: A family of high-quality versatile speech generation models, 2024.
- [7] Pavel Andreev, Aibek Alanov, Oleg Ivanov, and Dmitry Vetrov. Hifi++: A unified framework for bandwidth extension and speech enhancement. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, page 1–5. IEEE, June 2023.
- [8] Huda Barakat, Oytun Turk, and Cenk Demiroglu. Deep learning-based expressive speech synthesis: a systematic review of approaches, challenges, and resources. *EURASIP Journal on Audio, Speech, and Music Processing*, 2024(1), February 2024.
- [9] CanopyAI. Orpheus-tts. <https://github.com/canopyai/Orpheus-TTS>, 2025.
- [10] Chen Chen, Yuchen Hu, Siyin Wang, Helin Wang, Zhehuai Chen, Chao Zhang, Chao-Han Huck Yang, and Eng Siong Chng. Audio large language models can be descriptive speech quality evaluators. In *ICLR*, 2025.
- [11] Hyunjae Cho, Wonbin Jung, Junhyeok Lee, and Sang Hoon Woo. SANE-TTS: Stable and natural end-to-end multilingual text-to-speech. In *Interspeech 2022*, page 1–5. ISCA, September 2022.
- [12] Yunfei Chu, Jin Xu, Xiaohuan Zhou, Qian Yang, Shiliang Zhang, Zhijie Yan, Chang Zhou, and Jingren Zhou. Qwen-audio: Advancing universal audio understanding via unified large-scale audio-language models, 2023.
- [13] DeepSeek-AI. Deepseek-R1: Incentivizing reasoning capability in llms via reinforcement learning, 2025.
- [14] Zhihao Du, Qian Chen, Shiliang Zhang, Kai Hu, Heng Lu, Yexin Yang, Hangrui Hu, Siqi Zheng, Yue Gu, Ziyang Ma, Zhifu Gao, and Zhijie Yan. Cosyvoice: A scalable multilingual zero-shot text-to-speech synthesizer based on supervised semantic tokens, 2024.
- [15] Shujie Hu, Long Zhou, Shujie Liu, Sanyuan Chen, Lingwei Meng, Hongkun Hao, Jing Pan, Xunying Liu, Jinyu Li, Sunit Sivasankaran, Linquan Liu, and Furu Wei. Wavlm: Towards robust and adaptive speech large language model, 2024.
- [16] Wen-Chin Huang, Szu-Wei Fu, Erica Cooper, Ryandhimas E. Zezario, Tomoki Toda, Hsin-Min Wang, Junichi Yamagishi, and Yu Tsao. The voicemos challenge 2024: Beyond speech quality prediction, 2024.
- [17] Shengpeng Ji, Tianle Liang, Yangzhuo Li, Jialong Zuo, Minghui Fang, Jinzheng He, Yifu Chen, Zhengqing Liu, Ziyue Jiang, Xize Cheng, Siqi Zheng, Jin Xu, Junyang Lin, and Zhou Zhao. Wavreward: Spoken dialogue models with generalist reward evaluators, 2025.
- [18] Mateusz Łajszczak, Guillermo Cámara, Yang Li, Fatih Beyhan, Arent Van Korlaar, Fan Yang, Arnaud Joly, Álvaro Martín-Cortinas, Ammar Abbas, Adam Michalski, et al. BASE TTS: Lessons from building a billion-parameter text-to-speech model on 100k hours of data. *arXiv preprint arXiv:2402.08093*, 2024.

- [19] Javier Latorre, Kayoko Yanagisawa, Vincent Wan, Balakrishna Kolluru, and M.J.F. Gales. Speech intonation for tts: Study on evaluation methodology. 09 2014.
- [20] Jianli Liu and Jinying Chen. *The Application of Speech Synthesis in Car Warning System*, pages 657–662. 10 2014.
- [21] Zijun Liu, Peiyi Wang, Runxin Xu, Shirong Ma, Chong Ruan, Peng Li, Yang Liu, and Yu Wu. Inference-time scaling for generalist reward modeling, 2025.
- [22] Haowei Lou, Hye young Paik, Sheng Li, Wen Hu, and Lina Yao. Generalized multilingual text-to-speech generation with language-aware style adaptation, 2025.
- [23] Georgia Maniati, Alexandra Vioni, Nikolaos Ellinas, Karolos Nikitaras, Konstantinos Klapsas, June Sig Sung, Gunu Jho, Aimilios Chalamandaris, and Pirros Tsiakoulis. Somos: The samsung open mos dataset for the evaluation of neural text-to-speech synthesis. In *Interspeech 2022*, interspeech\_2022. ISCA, September 2022.
- [24] Bolaji Oladokun, Rexwhite Enakrire, Wole Olatokun, and Yusuf Ajani. Youth disability: Adopting text-to-speech and screen reader technologies for library and information services. 04 2024.
- [25] Bryan Patrick, Kaisha Stone, and Tommy Fred. Applications of voice quality technology in virtual assistants and call centers. 12 2024.
- [26] Charuta Pethe, Bach Pham, Felix D Childress, Yunting Yin, and Steven Skiena. Prosody analysis of audiobooks, 2025.
- [27] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision, 2022.
- [28] Sandra Raffoul and Lindsey Jaber. Text-to-speech software and reading comprehension: The impact for students with learning disabilities. *Canadian Journal of Learning and Technology*, 49(2):1–18, November 2023.
- [29] Paul K. Rubenstein, Chulayuth Asawaroengchai, Duc Dung Nguyen, Ankur Bapna, Zalán Borsos, Félix de Chaumont Quitry, Peter Chen, Dalia El Badawy, Wei Han, Eugene Kharitonov, Hannah Muckenhirn, Dirk Padfield, James Qin, Danny Rozenberg, Tara Sainath, Johan Schalkwyk, Matt Sharifi, Michelle Tadmor Ramanovich, Marco Tagliasacchi, Alexandru Tudor, Miha-jlo Velimirović, Damien Vincent, Jiahui Yu, Yongqiang Wang, Vicky Zayats, Neil Zeghidour, Yu Zhang, Zhishuai Zhang, Lukas Zilka, and Christian Frank. Audiopalm: A large language model that can speak and listen, 2023.
- [30] 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, 2024.
- [31] 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.
- [32] Thibault Sellam, Ankur Bapna, Joshua Camp, Diana Mackinnon, Ankur P. Parikh, and Jason Riesa. Squid: Measuring speech naturalness in many languages. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, page 1–5. IEEE, June 2023.
- [33] Sesame. Csm 1b. <https://huggingface.co/sesame/csm-1b>, 2025.
- [34] Changli Tang, Wenyi Yu, Guangzhi Sun, Xianzhao Chen, Tian Tan, Wei Li, Lu Lu, Zejun Ma, and Chao Zhang. Salmonn: Towards generic hearing abilities for large language models, 2024.
- [35] Gemini Team. Gemini: A family of highly capable multimodal models, 2025.
- [36] Attila Vékony. Speech recognition challenges in the car navigation industry. pages 26–40, 08 2016.

- 463 [37] Brendan Walsh, Mark Hamilton, Greg Newby, Xi Wang, Serena Ruan, Sheng Zhao, Lei He,  
464 Shaofei Zhang, Eric Dettinger, William T. Freeman, and Markus Weimer. Large-scale automatic  
465 audiobook creation, 2023.
- 466 [38] Siyang Wang and Éva Székely. Evaluating text-to-speech synthesis from a large discrete  
467 token-based speech language model, 2024.
- 468 [39] Siyin Wang, Wenyi Yu, Yudong Yang, Changli Tang, Yixuan Li, Jimin Zhuang, Xianzhao Chen,  
469 Xiaohai Tian, Jun Zhang, Guangzhi Sun, Lu Lu, Yuxuan Wang, and Chao Zhang. Enabling  
470 auditory large language models for automatic speech quality evaluation, 2025.
- 471 [40] Xinsheng Wang, Mingqi Jiang, Ziyang Ma, Ziyu Zhang, Songxiang Liu, Linqin Li, Zheng  
472 Liang, Qixi Zheng, Rui Wang, Xiaoqin Feng, Weizhen Bian, Zhen Ye, Sitong Cheng, Ruibin  
473 Yuan, Zhixian Zhao, Xinfu Zhu, Jiahao Pan, Liumeng Xue, Pengcheng Zhu, Yunlin Chen, Zhifei  
474 Li, Xie Chen, Lei Xie, Yike Guo, and Wei Xue. Spark-tts: An efficient llm-based text-to-speech  
475 model with single-stream decoupled speech tokens, 2025.
- 476 [41] Kangxiang Xia, Dake Guo, Jixun Yao, Liumeng Xue, Hanzhao Li, Shuai Wang, Zhao Guo, Lei  
477 Xie, Qingqing Zhang, Lei Luo, Minghui Dong, and Peng Sun. The iscslp 2024 conversational  
478 voice clone (covoc) challenge: Tasks, results and findings, 2024.
- 479 [42] Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and  
480 Daxin Jiang. Wizardlm: Empowering large language models to follow complex instructions.  
481 *arXiv preprint arXiv:2304.12244*, 2023.
- 482 [43] Jin Xu, Zhifang Guo, Jinzheng He, Hangrui Hu, Ting He, Shuai Bai, Keqin Chen, Jialin Wang,  
483 Yang Fan, Kai Dang, Bin Zhang, Xiong Wang, Yunfei Chu, and Junyang Lin. Qwen2.5-omni  
484 technical report, 2025.

## APPENDIX

The appendix is organized as follows:

**Section A:** Prompts and details for breadth and depth refinement of each category, along with final dataset statistics.

**Section B:** MMAU benchmarking of LALMs.

**Section C:** Evaluation related details, such as hyperparameters, audio generation prompts, and prompts for the judge.

**Section D:** Benchmarking other open-source models such as Suno Bark, Mini-CPM, Tortoise-TTS and Zyphra Zonos.

**Section E:** Analysis of Gemini-2.5-Pro as a judge and the case of Audio Subjectivity.

**Section F:** Text Normalization prompt and examples.

### A Per-Category Depth and Breadth Refinement

For breadth-expansion, we leverage long-thinking LLMs like Gemini 2.5 Pro, GPT-o3 and Claude 3.7 Sonnet. We prompt all three with the same breadth prompt, and through manual analysis select the LLM which produces the best breadth expansion and report the same for each category below. Next, all depth-refinements are achieved using Gemini-2.5-Pro. We create the following template, which is populated with the `**text_to_synthesize**`, and the refinement method for each category `**complex_prompt_method**`. The depth refinement prompt has a field `tts_synthesis_diversity` required in the LLM output, this is the field where COT specifications are provided for each category to ensure high-quality diverse and complex refinements from the LLM. The template is:

```
"""
You will you act as a Text Rewriter for a piece of text text_to_synthesize,
which is the text corresponding to which a TTS system has to synthesize
realistic speech.
Your objective is to rewrite and evolve the given text into a more complex
version rewritten_text_to_synthesize, which makes the famous Speech
Generation AI systems (e.g., ElevenLabs, Deepgram) harder to handle as
compared to the original text_to_synthesize.
The underlying goal is to be come up with the rewritten_text_to_synthesize
such that,
It is more complex, deep and harder for a TTS system to synthesize than 
text_to_synthesize.
You WILL complicate and complexify the given text_to_synthesize using the
following method:
complex_prompt_method
{{{complex_prompt_method}}}
/complex_prompt_method
Now, you will be provided with the text_to_synthesize
text_to_synthesize
{{{text_to_synthesize}}}
/text_to_synthesize

Output Format:
You will output a json object with the following fields:
```json
{
  "text_to_synthesize": str <Verbatim copy of the original text to synthesize>,
  "tts_synthesis_diversity": str <Reasoning on how you can complicate the text_to_synthesize using the details specified in complex_prompt_method to make it more challenging for the TTS system to synthesize>,
  "rewritten_text_to_synthesize": str <The rewritten text the TTS system has to synthesize which is more complex and deep than text_to_synthesize>
}
/Output Format:
```

```

544     Now, you will output the correct json object following the **Output Format:**
545     and without producing any additional text.
546     ""
547

```

548 In the following sections, we present the category-wise breadth expansion prompts and the  
549 `complex_prompt_method` used for each category.

## 550 A.1 Category 1: Questions

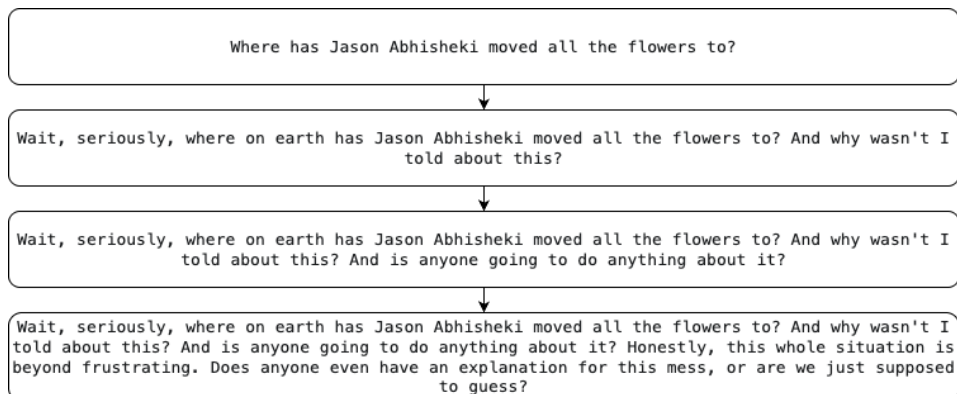


Figure 5: Example depth-refinement for questions category. Starting with a simple Wh-question, complexity is introduced by first by adding a subsequent Wh-question with a pragmatic nuance, then a Yes/No question to test pitch contour shifts. The final refinement examines the differentiation between interrogative and declarative prosody by inserting an emphatic statement, and further tests nuanced intonation with a concluding alternative question

551 **Breadth Expansion** We use 20 samples that were proposed for this category in BASE-TTS,  
552 and then prompt **Gemini-2.5-Pro** to curate 50 more samples that achieve a significantly wider  
553 coverage for evaluating TTS interrogative prosody. Beyond standard Yes/No and Wh- questions,  
554 we incorporate negative questions, rhetorical questions, declarative questions expressing surprise  
555 or doubt, hypotheticals, alternative questions involving lists, and questions featuring parenthetical  
556 elements or starting conjunctions, to get a total of 70 samples with richer syntactic diversity and  
557 broader prosodic demands. The breadth expansion prompt used is:

```

558 Consider the below set of 20 samples. This set belongs to the "Questions" category
559 and is used for create an extremely diverse set for evaluation of TTS systems,
560 where they have to synthesize the text corresponding to **text_to_synthesize**.
561 This category evaluates whether the system correctly applies question intonation
562 patterns. Questions usually have a distinct pitch movement, often rising at the
563 end in yes/no questions, while wh-questions may have a more neutral or falling
564 tone, and there are many other scenarios that can be used to evaluate a TTS
565 system not covered in the 20 samples.
566 Your goal is to generate 50 more samples belonging to this category. You will do
567 this in the following step-by-step manner:
568 1. You will analyze the 20 samples carefully.
569 1.a. Reason deeply about the types of questions this set contains sample-by-sample,
570 and the corresponding intonation patterns that these questions might elicit
571 from our TTS system. Remember that a single text can have variability in
572 intonation pattern, but we have to form an abstract map of the patterns that we
573 are seeing.
574 1.b Reason deeply about the **text_to_synthesize** structures present, like the
575 placement of question marks, number of questions marks, the grammatical
576 structure of the texts.
577 2. Now, you will think long about what this set is **MISSING**, specifically, the
578 various types of questions that exist in the complete set of english texts, but
579 are not present in this set, **AND** will be great to test the intonation
580 pattern of TTS systems on.
581

```

```

582 3. Finally, you will create additional 50 samples, that expand the current 20 sample
583 set in terms of **DIVERSITY**, as you are doing a breadth-wise evolution of
584 this 20 set.
585 The main goals for the set are:
586 1. There should be diversity in elicited intonation patterns and types of questions.
587 2. There should be diversity in terms of sentence grammatical structure.
588 3. No sentence should contain cues like * or **.
589 4. The 50 samples will follow the same format as the 20 samples, **BUT** no sample
590 in the 50 set should be similar to what is in the 20 samples, in terms of
591 context and phrasing.
592 5. You will not add diversity through italics, bold, UPPERCASE, etc.
593
594 Now, you are given the 20 sample set, after this, think deeply and create the 50
595 question set.
596 ```jsonl
597 <now the 20 samples were provided in jsonl format>
598 ```
599

```

600 **Depth Refinement** To move beyond simple questions, we utilize the depth refinement strategy to  
601 generate utterances demanding highly varied interrogative and declarative prosody from the TTS  
602 system. We refine iteratively using one of three methods: appending a sequential question, appending  
603 a statement and then a question, or infusing pragmatic nuance before appending a question. The  
604 resulting dataset is designed to specifically evaluate a TTS system’s proficiency in: (a) rendering  
605 natural pitch contours across consecutive questions, (b) executing smooth prosodic transitions  
606 between declarative and interrogative speech within one utterance, and (c) conveying subtle pragmatic  
607 meanings (like skepticism or politeness) through appropriate intonational variation alongside the core  
608 question structure. Refer to Figure 5 for an example refinement and the depth-refinement prompt is  
609 as follows:

```

610 complex_prompt_method="""
611     You are evolving a **text_to_synthesize** sample belonging to the "Questions"
612     category for evaluating Text-to-Speech (TTS) systems.
613 The primary goal of this category is to test if the TTS system correctly applies
614 natural, varied, and appropriate **question intonation patterns** reflecting
615 different question types, pragmatic functions, and attitudes, especially in
616 sequence.
617
618 Your task is to **increase the prosodic complexity related specifically to the act
619 of questioning and its conversational context**, making the task more
620 challenging for the TTS system’s intonation capabilities. Apply **ONE** of the
621 following evolution methods:
622
623 **Choose ONE Method per Evolution:**
624
625 1. **Method 1: Add Sequential Question:** Add *one* related, grammatically complete
626 question *immediately following* the existing text. The goal is to create a
627 sequence that tests the TTS system’s ability to naturally transition
628 prosodically between two consecutive questions, potentially involving different
629 question types or nuances **to test varied intonation patterns**.
630
631 2. **Method 2: Add Sequential Statement + Question:** Add *one* related,
632 grammatically complete statement *immediately following* the existing text, and
633 then add *one* related, grammatically complete question *immediately following*
634 that statement*. The goal is to test the TTS system’s ability to naturally
635 transition prosodically from the original text’s context, through a declarative
636 statement (with appropriate intonation), and into a final question (with
637 appropriate interrogative intonation).
638
639 3. **Method 3: Infuse Pragmatic Nuance & Add Sequential Question:** First, **modify
640 the existing text** by adding or changing words/phrases (e.g., adverbs,
641 introductory phrases, discourse markers, slight rephrasing) to require the TTS
642 system to convey a specific **attitude or pragmatic function** (like doubt,
643 surprise, politeness, insistence, etc.) primarily through prosody. Then, **add
644

```

one\*\* related, grammatically complete question \*immediately following\* the modified text. The goal is to test the TTS system’s ability both to render the subtle nuance in the first part and to transition naturally into the appropriate intonation for the subsequent question.

**Crucial Constraints:**

- \* The final evolved text must remain a **natural-sounding, self-contained** utterance spoken by a **single speaker**.
- \* The modification should primarily challenge the **prosodic rendering** (intonation, pitch, stress, rhythm, phrasing, pauses) related to the **questioning** function, attitude, or sequence.
- \* **IMPORTANT:** Avoid significantly increasing **internal grammatical complexity** (e.g., complex clauses, deep nesting, excessive parentheticals) **unless** it is a direct and necessary result of naturally expressing the pragmatic nuance or creating the statement/question sequence described in the methods. The goal is **question prosody diversity**, not primarily syntactic parsing difficulty.
- \* **IMPORTANT:** Do **NOT** use formatting characters like bold (**\*\*), italics (\*’), or ALL CAPS to indicate emphasis or complexity. The challenge must come from the text structure and implied prosody itself.**

In the ‘tts\_synthesis\_diversity’ field:

1. Clearly state **which Method (1, 2, or 3)** you applied.
2. Explain **specifically how** this modification increases the challenge for a TTS system’s **question prosody rendering**, focusing on the required intonation patterns, pitch movements, stress placement, phrasing, timing, or the need to convey subtle nuances and manage transitions compared to the input text. Avoid focusing justification solely on syntactic structure.

"""

## A.2 Category 2: Foreign Words

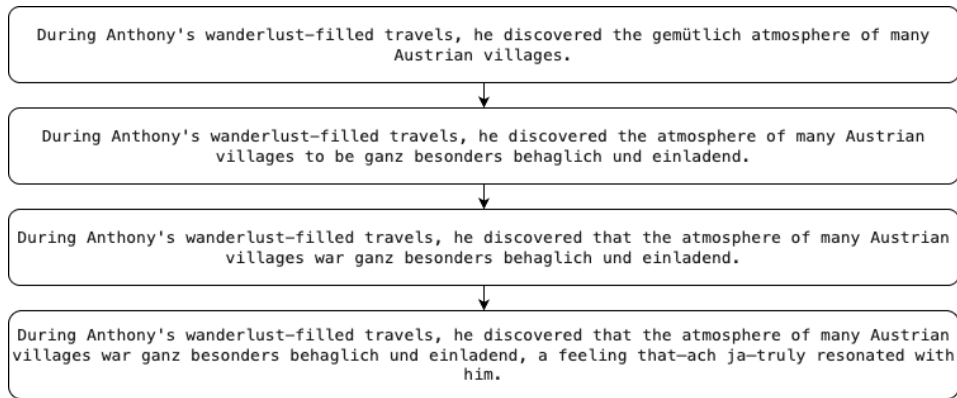


Figure 6: Example depth-refinement for foreign words category. Starting with a text containing one isolated foreign words, we expand it with german phrase. Then, the english word "to be" is replace with "war". In the final evolution, new text is added in the suffix with english and german words.

**Breadth Expansion** The initial set of 20 BASE-TTS samples showed several constraints - they primarily featured European languages and frequently used easier to pronounce loanwords as the foreign word. To improve variety, we employed **GPT-o3** to create an additional 50 samples. This expansion significantly enhances language diversity by incorporating more samples from the 10 most spoken languages around the world (Mandarin Chinese, Hindi, Spanish, French, Arabic, Russian, Portuguese, Japanese, German and Indonesian), with particular emphasis on uncommon foreign vocabulary that presents pronunciation challenges. All non-English words appear exclusively in standard Latin characters without using any special foreign alphabetic symbols; this design choice is intentional and follows our reasoning for testing the emergent capabilities of TTS systems, and



685 while multilingual training data with foreign character set might be highly available, the same is very  
686 limited or non-existent with Latin characters only. The breadth expansion prompt used is as follows:

```
687
688     Consider the below set of 20 samples. This set belongs to the "Foreign Words"
689         category and you will use it to create an extremely diverse set for
690         evaluation of TTS systems, where the systems have to synthesize the text
691         corresponding to text_to_synthesize.
692 This category tests whether the system correctly pronounces foreign words and
693 phrases, either using their original pronunciation or a widely accepted
694 anglicized version, and there are many other foreign words that can be used to
695 evaluate a TTS system which may not be covered in the 20 samples.
696 Your goal is to generate 50 more samples belonging to this category. You will do
697 this in the following step-by-step manner:
698 1. You will analyze the 20 samples carefully.
699 1.a. Reason deeply about the types of languages covered and the commonness of the
700     foreign words among bi-lingual population that speak english and the language
701     of that foreign word belongs to in the sample.
702 1.b Reason deeply about the text_to_synthesize structures present, like the
703     placement of foreign words, the number of foreign words, and the grammatical
704     structure of the texts.
705 2. Now, you will think long about what this set is MISSING, specifically, the
706     various types of foreign words that exist in the COMPLETE set of foreign
707     words used by bi-lingual speakers, but are not present in this set, AND
708     will be great to test foreign word synthesis ability of TTS systems.
709 3. Finally, you will create additional 50 samples, that expand the current 20 sample
710     set in terms of DIVERSITY, as you are doing a breadth-wise evolution of
711     this 20 set.
712
713 The main goals for the set are:
714 1. The additional 50 samples will come from these 10 wide-spread languages(5 from
715     each language): Mandarin Chinese, Hindi, Spanish, French, Modern Arabic,
716     Russian, Portuguese, Japanese, German and Indonesian (Malay). Ensure that each
717     language has atleast 1 foreign phrase
718 2. All text_to_synthesize that will be generated in the 50 samples, should form a
719     fluent sentence that a bi-lingual speaker incorporating that language might
720     speak. This means, we don't have to write the english translation or synonym of
721     that word along with the foreign word, its one single fluent sentence
722     without redundancy and completely natural flow.
723 3. There should be diversity in terms of placement position of foreign words, and
724     the number of foreign words should either be 2 or 3 across samples.
725 4. All foreign words must be places inside "quotes", and should naturally intergrate
726     with the surrounding context.
727 5. You may incorporate small foreign phrases like the initial 20-sample set has joie
728     de vivre, pick carefully where you want phrases, and DO NOT include the
729     phrases in the quotes "", only words are to be included in quotes.
730 6. The 50 samples will follow the same JSONL format as the 20 samples, BUT no
731     sample in the 50 set should have the same foreign word as the foreign words in
732     the 20 sample set. All foreign words MUST not be duplicated.
733 7. You will not create texts with * word * or ** word ** or add text inside
734     parenthesis () to indicate something.
735 8. AVOID LOANWORDS: All the foreign words generated in the 50 sample set should be
736     uncommon and slightly complex to pronounce correctly in that language, we don'
737     t want to include words that may have an easy pronunciation for an english
738     speaker. AT THE SAME TIME, do not use very obscure words that don't exist in
739     the modern vocabulary, we want to create natural day-to-day sentences that bi-
740     lingual speakers will speak
741 9. You will NOT adopt characters from foreign languages, all foreign words must be
742     expressed with english and latin letters.
743 10. In the "misc" field, add a new key "foreign_language", and populate it with the
744     language of that sample
745
746 Now, you are given the 20 sample set, after this, think deeply and create the 50
747     foreign words set.
748 jsonl
749 <now the 20 samples were provided in jsonl format>
```

**Depth Refinement** While evaluating TTS on isolated foreign words tests basic pronunciation, it doesn't reflect the full complexity of natural bilingual communication, which frequently integrates longer foreign phrases, we fill this gap through the depth refinement strategy. This method systematically transforms simpler utterances into variants featuring more substantial foreign language segments, mimicking how bilingual individuals speak and write. Guided by a specific prompt, an LLM applies one of three approaches: (i) replacing isolated words with idiomatic phrases, (ii) expanding short phrases by absorbing and translating adjacent English context, or (iii) appending bilingual affixes to already complex sentences. This yields utterances requiring the TTS system to manage fluent code-switching and natural prosody over extended foreign elements, providing a more rigorous assessment of its capabilities. This refinement strategy resulted in complex code-switching, but due to grammatical differences between languages, often created awkward sentences by the final refinement. To remedy this, we post-process the output of each refinement step through a separate LLM call to gemini-2.5-pro, to fix any grammatical and syntactical issues, we found this to be quite effective. Figure 6 illustrates this process, and the depth-refinement prompt is as follows:

```

766 complex_prompt_method = """
767 The **text_to_synthesize** belongs to the "Foreign Words" category.
768 This category evaluates whether the system can fluently pronounce foreign words and
769 phrases, smoothly switching between different languages within one **
770 text_to_synthesize**.
771 Your goal: Increase TTS synthesis complexity by adding exactly **one natural, fluent
772 bilingual flourish**.
773
774 **EVOLUTION LOGIC (you will choose ONE of the following approaches):**
775
776 Approach 1: Expand an **ISOLATED** Foreign Word
777 - **Condition**:
778   - For this approach to apply, the **text_to_synthesize** will contain ATLEAST
779     ONE **ISOLATED** foreign word.
780   - Additional information:
781     - By **ISOLATED**, we mean it's a single individual word or a compound noun
782       (like 'jamón ibérico' or 'Feng Shui') that functions as one unit.
783     - A word is not isolated even if it is a single unit if it is surrounded by
784       foreign words. It should be <english_word> <isolated_word_or_unit> <
785       english_word> for this approach to apply.
786     - These **ISOLATED** words can be within "quotes" or be unquoted.
787     - The **text_to_synthesize** may contain multiple such **ISOLATED** words.
788 - **Transformation**
789   - Choose one of the **ISOLATED** foreign word, and replace it with a one of the
790     following:
791     - A longer prosodically rich phrase.
792     - A longer idiomatic phrase(ONLY IF IT FITS THE CONTEXT EXTREMELY WELL).
793     - This phrase should **MOSTLY** be in the foreign language, but can have a
794       few english words to make it natural and test rapid code-switching.
795     - This phrase can be another way of saying the **ISOLATED** word, **OR** is
796       a replacement for the **ISOLATED** word while maintaining sense of the
797       surrounding text.
798   - Whether or not the the **ISOLATED** foreign word is in quotes, the phrase you
799     will replace it with will not be within quotes(unless it is a dialogue).
800   - Choose the **ISOLATED** word that will support the most natural expansion from
801     word to idiomatic or prosodically rich phrase.
802   - The added phrase should be **ATMOST** 5 words long.
803
804 **If the condition for Approach 1 is not satisfied, you will move to Approach 2**
805
806 Approach 2: **Grow** an Existing **SHORT** Foreign Phrase by **Absorbing** English
807 Context
808 - **Condition**:
809   - The **text_to_synthesize** DOES NOT contain any **ISOLATED** foreign words.
810

```

811 - The **text\_to\_synthesize** contains **foreign phrases** where **ATLEAST ONE**  
812 phrase is no more than **5 words** long. (See Global Constraints for '  
813 foreign phrase' definition).

814 - The phrases that follow the condition are called **SHORT FOREIGN PHRASE**.

815 - **Transformation**:

816 - **Select** one **SHORT FOREIGN PHRASE** ( $\leq 5$  words).

817 - **Identify adjacent English words** (immediately before and/or after the  
818 selected phrase) that can be naturally incorporated into the foreign  
819 language segment.

820 - **You will identify at least TWO adjacent English words** that can be  
821 naturally incorporated into the foreign language segment.

822 - **Convert** these adjacent English words into the **same foreign language** as  
823 the phrase.

824 - **Integrate** the original short foreign phrase and the newly translated words  
825 into a single, **longer continuous segment** of the foreign language.

826 - **Crucially**: Make necessary **grammatical adjustments** within this  
827 expanded foreign segment for fluency and correctness in the foreign  
828 language (e.g., adjust word order, add/remove articles/prepositions, use  
829 correct verb conjugations or noun declensions as needed in that language).

830 - **Ensure** the transition from English into this longer foreign segment, and  
831 back into English afterwards, remains smooth and the overall sentence is  
832 fluent and grammatically sound.

833 - The goal is to create a **longer continuous block** of the foreign language,  
834 testing sustained synthesis and integration.

835 - Choose the short phrase and surrounding English context that allows for the  
836 most natural grammatical integration and fluent expansion into a longer  
837 foreign segment.

838 - **Do NOT apply** this transformation to any foreign phrase that is already **more than 5 words** long in the original text.

840 - If all foreign phrases are already long ( $> 5$  words), skip to Approach 3.

841

842 **If both the conditions for Approach 1 and Approach 2 are not satisfied, you will**  
843 **move to Approach 3**

844

845 Approach 3: Insert additional text with a **NEW FOREIGN PHRASE** (Prefix or Suffix)

846 - **Condition**:

847 - If the conditions for Approach 1 and Approach 2 are not satisfied, you will  
848 move to Approach 3.

849 - **Transformation**:

850 - Add additional text with english words **AND** a **NEW FOREIGN PHRASE** in the  
851 same foreign language already used in the utterance that is idiomatic and  
852 prosodically rich.

853 - The new foreign phrase will be either a **prefix** or **suffix** to the **text\_to\_synthesize**.

854

855 - Choose one of:

856 **Prefix**: Add at the start of the sentence, as a lead-in.

857 **Suffix**: Add at the end, as a reflective or emotional continuation.

858 - Inserted phrase must be **plausible, fluent, narratable** by a bilingual speaker  
859 **and MUST** contain words borrowed from both english and the foreign  
860 language.

861 - The newly added text must be **ATMOST** 10 words long, including the english  
862 words and the foreign phrase.

863 ---

864

865 **Global Constraints (Always Apply)**:

866 - Use **only English + one foreign language** (same throughout the utterance).

867 - There will always be some foreign word or phrase in the text that you have to  
868 recognize correctly.

869 - All foreign words/phrases must be in **Latin transcription** AND pinyin  
870 transcription for Chinese. (no native scripts or characters from the foreign  
871 language).

872 - **Definition**: A 'foreign phrase' specifically refers to **a contiguous sequence**  
873 of two or more words from the foreign language, with no English words  
874 interrupting that sequence. Single words or compound nouns acting as single  
875 units are considered 'isolated words' for Approach 1.

```

876 - You **WILL** do paraphrasing of the text after the editing to make it more natural
877     and eliminate any redundant text/unnatural text.
878 - Use commas, dashes, or natural punctuation to integrate long prefixes and suffixes
879     and foreign phrases.
880 - Do not use quotation marks around foreign phrases. But if quotes are used to
881     represent dialogues, you will preserve them. Only quotes that signify that a
882     word is a foreign word will be removed.
883 ---
884
885 **Phonetic & Prosodic Complexity (TTS Focus)**
886 - Expanded or inserted phrases should enhance TTS difficulty via:
887     - Nasal vowels
888     - Consonant clusters
889     - Rolling R's, vowel alternation, liaison
890     - Multisyllabic cadence, rhythmic variation
891 - Never sacrifice fluency, narratability, or naturalness.
892
893 ---
894
895 In your 'tts_synthesis_diversity' explanation, clearly and consistently state:
896 1. What approach (1, 2, or 3) was applied and **why**.
897 2. What foreign language is present in the **text_to_synthesize** other than English
898     .
899 3. What specific word(Approach 1) or phrase(Approach 2) will be expanded/absorbed or
900     in case of Approach 3, what will be the newly added text and the foreign
901     phrase. Mention how this **DOES NOT** make the text syntactically awkward.
902 4. The exact word count of the newly foreign phrase, and in case of Approach 3, the
903     word count of the newly added text and the foreign phrase. **For Approach 2,
904     state the original short phrase, the absorbed English words, and the resulting
905     longer foreign segment with its word count.**
906 5. Phonetic/prosodic challenges introduced (e.g., nasal vowels, clusters, rhythm).
907     **For Approach 2, emphasize the challenge of sustained foreign language
908     synthesis and grammatical integration.**
909 6. **Intermediate Checking:** Based on the above details, print exactly what the
910     candidate output **rewritten_text_to_synthesize** will be like.
911 7. Carefully analyze the current candidate **rewritten_text_to_synthesize** and
912     7.1 Add any necessary punctuations in the candidate to ensure the text is
913     coherent and logically correct, both **INSIDE** and **OUTSIDE** the
914     idiomatic/prosodically rich foreign phrases.
915     7.2 Restructure, paraphrase and edit the text grammatically to ensure **NO
916     AWKWARDNESS** in the text. Analyze the gender, the noun-adjective agreement,
917     the verb-subject agreement, tenses, etc.
918 8. Now, after all above steps, confirm that the final result:
919     - Does not include **more than one foreign language**
920     - Does not include **foreign phrases >5 words**, unless it is a Approach 3
921     prefix/suffix **or the result of Approach 1/Approach 2 expansion/absorption
922     **.
923     - Does not include **foreign words/phrases** that are not in **Latin
924     transcription** (no native scripts or characters from the foreign language
925     are allowed).
926     - The final result is a realistic text that can be narrated by a bilingual
927     speaker in a day-to-day conversation or during narration of a story.
928     - The final result is a **grammatically correct** and **syntactically fluent**
929     text. It has **NO AWKWARDNESS** with respect to the gender and flow of the
930     text.
931 ""
932

```

933 We also share the system message used for the post-processing step of fixing grammatical awkward-  
934 ness using LLM, the `text_to_synthesize` is provided as the user-message.

```

935 post_process_prompt = ""
936 You are given a sentence that has code-switching at multiple points between two
937 languages.
938 Your goal is to refine the sentence, to remove any grammatical issues and
939 awkwardness that is present.
940

```

941 - You will particularly be careful about the gender, the noun-adjective agreement,  
 942 the verb-subject agreement, tenses, punctuations, etc.  
 943 - You will recognize if the text is syntactically incorrect or overly complex, in  
 944 which case you will untangle it and make it syntactically easier to read.  
 945 - You will not add any characters from the foreign language, we will only use latin  
 946 transcription and pinyin transcription for chinese.  
 947 Your goal is to return the refined sentence that is **\*\*SIMILAR\*\*** to the original  
 948 sentence, but has **\*\*no\*\*** grammatical issues, awkwardness, unnaturalness, and is  
 949 easier to read for a bi-lingual speaker proficient in the foreign language.  
 950 - You **\*\*WILL NOT\*\*** add markers that are meant to help with identification of the  
 951 foreign segments, like do not add markers like \* or \*\* that are not present in  
 952 the original sentence.  
 953 You will output **\*\*ONLY\*\*** the refined sentence, with no other information or text.  
 954 ""  
 955 ""

### 956 A.3 Category 3: Paralinguistics

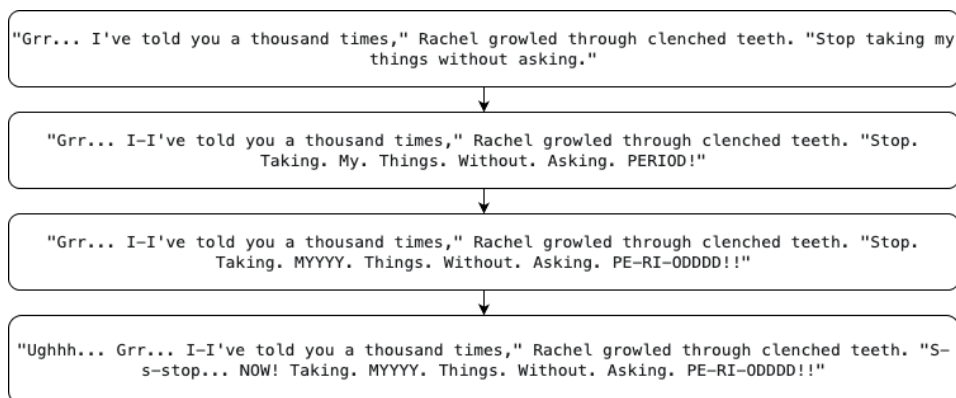


Figure 7: Example depth-refinement for paralinguistics category. Starting with just one cue "Grr", we add stuttering "I-I've", punctuation(Stop.Taking.My..) and caps(PERIOD). Then, syllable stress and elongation is introduced through PE-RI-ODDDD. Finally, we add more cues like "Ughhh", "s-stop" and "NOW!".

957 **Breadth Expansion** The initial set of 20 samples from BASE-TTS provided foundational coverage  
 958 of common paralinguistic phenomena, including basic interjections (e.g., 'Aha!', 'Oops'), simple  
 959 vocal sounds ('Yawn', 'psst'), and common hesitations ('Uh', 'Hmm'). However, this initial set  
 960 lacked significant diversity. It underrepresented a wider spectrum of emotional interjections, varied  
 961 onomatopoeia (spanning bodily, environmental, and animal sounds), nuanced textual emphasis  
 962 cues, explicit pacing markers, and complex speech disfluencies. The breadth-expanded set with  
 963 50 additional samples significantly addressed these gaps by incorporating 5 types of cues such  
 964 as: (i) additional interjections (e.g., Eww, Gasp, Tsk tsk, Oy vey), (ii) diverse onomatopoeia (e.g.,  
 965 Achoo, Tick-tock, Pitter-patter), (iii) varied emphasis markers (e.g., capitalization like REALLY,  
 966 vowel elongation like sooooo, hyphenation like ab-so-lutely or Un-der-STAND), (iv) pacing cues via  
 967 ellipses (...) or punctuation (STOP. RIGHT. THERE.), and (v) stuttering representations (e.g., I-I,  
 968 N-N-NO). The breadth-expansion is achieved using **Claude 3.7 Sonnet** using the following prompt:

969 Consider the below set of 20 samples. This set belongs to the "Paralinguistics"  
 970 category and you will use it to create an extremely diverse set for evaluation  
 971 of TTS systems, where the systems have to synthesize the text corresponding to  
 972 **\*\*text\_to\_synthesize\*\***.  
 973 This category evaluates how well the system interprets **\*\*textual cues\*\***-like  
 974 interjections ("Wow!"), vocal sounds/onomatopoeia/animal sounds ("Shhh!", "Achoo!"),  
 975 emphasis (CAPS, "sooo", "ab-so-lutely"), punctuations ("?!"), or hesitations ("Uh...")-to  
 976 produce appropriate **\*\*non-neutral vocal effects\*\***, and there are many other paralinguistic  
 977 cues that can be used to evaluate a TTS system which may not be covered in the 20 samples.

```

980 Your goal is to generate 50 more samples belonging to this category. You will do
981 this in the following step-by-step manner:
982 1. You will analyze the 20 samples carefully.
983 1.a. Reason deeply about the types of paralinguistic cue this set contains sample-by-
984 -sample, and the corresponding sounds that these questions might elicit from
985 our TTS system.
986 1.b Reason deeply about the text_to_synthesize structures present, like the
987 placement of paralinguistic cues, and the grammatical structure of the texts.
988 2. Now, you will think long about what this set is MISSING, specifically, the
989 various types of paralinguistic cues that exist in the COMPLETE set of
990 paralinguistic cues present in english texts, but are not present in this set,
991 AND will be great to test paralinguistic speech synthesis ability of TTS
992 systems. You WILL think about abstract types of paralinguistic cues, and then
993 expand on what kind of cues they contain, for example, "Uh", "Uhhh", etc, all
994 belong to the sample abstract type.
995 3. Finally, you will create additional 50 samples, that expand the current 20 sample
996 set in terms of DIVERSITY, as you are doing a breadth-wise evolution of
997 this 20 set.
998
999 The main goals for the set are:
1000 1. There should be diversity in types of paralinguistic cues present.
1001 2. All the cues present in the 50 set you generate, should be realistically
1002 synthesized by a human. These should be sounds that humans can produce well
1003 based on the textual cue.
1004 3. There should be diversity in terms of sentence grammatical structure.
1005 4. The 50 samples will follow the same JSONL format as the 20 samples, BUT no
1006 sample in the 50 set should be similar to what is in the 20 samples, in terms
1007 of context and phrasing and paralinguistic cue present.
1008 5. You will not create texts with * word * or ** word ** or add text inside
1009 parenthesis ( ) to indicate something, the TTS system will synthesize cues
1010 directly present in the text itself.
1011
1012 Now, you are given the 20 sample set, after this, think deeply and create the 50
1013 paralinguistic set.
1014
1015 jsonl
1016 <now the 20 samples were provided in jsonl format>
1017 jsonl

```

**Depth Refinement** To get a representative set of paralinguistic cues that occur in written dialogue aiming to convey expressive speech, such as found in scripts, fictional narratives, and certain forms of informal communication, we apply the depth-refinement strategy. The refinement prompt uses the 5 defined types of paralinguistic cues and rewrites the text by incrementally adding one more or two cues of any type at each step. By the final refinement step, this process yields texts with multiple distinct paralinguistic cues, designed to work together to create a unique and realistic challenge for the TTS system. An example refinement is shown in Figure 7, and the prompt used is:

```

1027 complex_prompt_method = """
1028 You are tasked with enhancing the paralinguistic complexity of the provided text_to_synthesize, which belongs to the "Paralinguistics" category. This
1029 category tests a TTS system's ability to render non-lexical vocal cues (emotion,
1030 emphasis, sounds, hesitations, etc.) within the text itself, using
1031 appropriate prosody, pacing, and vocal quality.
1032
1033 Your specific task is to rewrite the text_to_synthesize by ADDING ONE or TWO
1034 more complex paralinguistic cues. Perform the following steps:
1035
1036 1. Analyze the existing paralinguistic cues within the spoken text and identify
1037 opportunities for enhancement.
1038 2. Enhance paralinguistic complexity using ONE appropriate technique that modifies
1039 the spoken text itself. Choose a method that fits naturally and logically
1040 within the context. Examples of techniques include, but are not limited to:
1041 * Adding Interjections or Filled Pauses directly into the speech (e.g., "Wow
1042 !", "Gosh", "Hmm", "Uh...", "Mmm").
1043

```

1044 \* Adding **\*\*Onomatopoeia or Explicit Vocal Sounds\*\*** as part of the utterance (e.g.  
1045 ., "Crash!", "Achoo!", "Shhh", "Psst").

1046 \* Introducing/Intensifying **\*\*Emphasis Cues \*within the text\*\*** (e.g., using ALL  
1047 CAPS for stressed words, using **\*\*hyphenation/syllable stress** like 'im-por-  
1048 tant', **\*\*repeating letters** like 'heyyyyy', using expressive punctuation  
1049 like '.....' or sequences like '. . .').

1050 \* Introducing **\*\*Hesitation/Stuttering Cues\*\*** directly in the speech (e.g., 'I-I-  
1051 I', 'W-we-well..., ok-ok-ok').

1052 \* Modifying phrasing or punctuation **\*within the speech\*** to suggest specific **\*\*  
1053 Pacing/Rhythm\*\*** (e.g., using **\*\*ellipses '... '\*\***, short staccato phrases  
1054 connected by hyphens or commas).

1055 **\*\* NOTE:** The above provided examples for each technique are only for your  
1056 reference, you may add them if it fits the context, but come up with your  
1057 paralinguistic cues that fit the technique and the context.

1058 3. **\*\*AVOID techniques relying on meta-textual instructions or non-standard  
1059 formatting:\*\*** Do **\*\*NOT\*\*** add parenthetical descriptions like '(laughing)', '(  
1060 angrily)', descriptive dialogue tags like '\*he whined\*', or **\*\*any markdown  
1061 formatting like '\*\*word\*\*' or '\*word\*' to indicate how the text should be  
1062 spoken. Focus only on cues the TTS would encounter in the direct text-to-be-  
1063 synthesized.**

1064 4. **\*\*Complexity Goal:\*\*** The technique should aim to **\*increase the demand\* \*\*  
1065 SIGNIFICANTLY\*\*** on the TTS system to produce specific, non-neutral vocal  
1066 delivery, without sacrificing realism or clarity.

1067 5. **\*\*CRITICAL CONSTRAINT - Realism & Clarity:\*\***

1068 \* Your **\*\*absolute priority\*\*** is ensuring the rewritten text is **\*\*grammatically  
1069 correct\*\*** and sounds **\*\*realistic\*\*** for human speech or dialogue as written (  
1070 e.g., plausible dialogue, expressive utterances).

1071 \* It must **\*\*NOT\*\*** sound forced, nonsensical, overly exaggerated beyond  
1072 plausibility, or contain cues that contradict each other.

1073 \* The enhancement must integrate **\*\*smoothly and coherently\*\***, logically fitting  
1074 the context and character (if implied). The intended paralinguistic effect  
1075 should be reasonably clear from the **\*\*standard textual cues within the  
1076 speech\*\***.

1077 \* **\*\*Prioritize realism and coherence over maximizing the number or intensity of  
1078 cues.\*\*** If an enhancement feels unnatural using standard cues, choose a  
1079 simpler or different one.

1080 \* **\*\*NO OMISSION OF ANY TEXT\*\*** Do not remove non-paralinguistic text from the **\*\*  
1081 text\_to\_synthesize\*\***.

1082 \* **\*\*NO TEXTUAL MARKERS\*\*** Do not add \* or \*\* or () characters to the text.

1083 \* **\*\*NO SINGLE LETTER HYPHENATIONS\*\*** Do not add single letter hyphenations like Y-  
1084 O-U. Hyphenations should always be natural syllable stressing.

1085 6. **\*\*Final Check:\*\*** Read the rewritten text aloud. Does it sound like something a  
1086 person might realistically say? Is the intended paralinguistic cue clear **\*from  
1087 the text itself using standard conventions\*?** Does it pose a relevant challenge  
1088 for TTS rendering based on these embedded cues?

1089

1090 **\*\* In the **\*\*tts\_synthesis\_diversity\*\*** field, you MUST provide detailed reasoning  
1091 covering:**

1092 1. The specific paralinguistic enhancement technique you want to use and how you  
1093 will apply it (which specific cue you will add) within the spoken text using  
1094 standard conventions. You can choose to add one or two cues.

1095 2. Comment on the novelty of the cue/cues you will add, if this is in one of the  
1096 examples provided within the technique description above, or you came up with  
1097 your own. Novel cues are preferred but not required.

1098 3. How the change will **\*\*SIGNIFICANTLY\*\*** enhance the paralinguistic challenge for  
1099 TTS, specifying the intended vocal effect after the change.

1100 4. **\*\*Crucially, justify \*why\* the rewritten text remains grammatically correct,  
1101 sounds realistic for speech/dialogue, and integrates smoothly. Explain how  
1102 coherence and logical flow were maintained.\*\*** Address the critical constraint  
1103 directly, including adherence to the rule about avoiding meta-textual cues and  
1104 non-standard formatting.

1105 ""

#### 1107 A.4 Category 4: Emotions

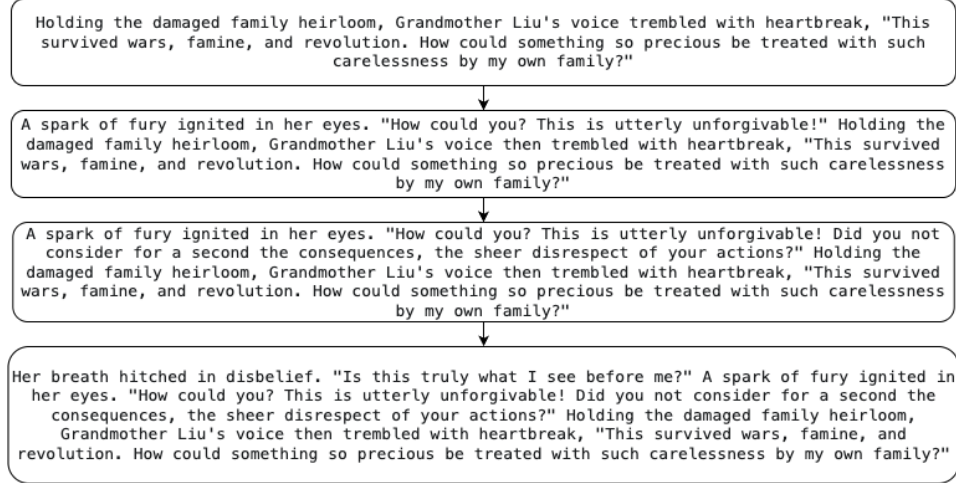


Figure 8: Example depth-refinement for emotions category. In refinement 1, we add narrative text and contrasting emotion to the one already present. This emotion is further intensified in refinement 2. In the final refinement, we add another contrasting emotion, to have three emotional states, disbelief → fury → heartbreak.

1108 **Breadth Expansion** The initial dataset of 20 samples from BASE-TTS, while covering foundational  
 1109 emotions like joy, anger, and sadness, exhibited limitations in emotional granularity and contextual  
 1110 depth necessary for comprehensive TTS evaluation. It predominantly featured strong, primary  
 1111 emotions and lacked sufficient diversity in more nuanced states such as sarcasm, envy, resignation,  
 1112 or complex blends like bitter-sweetness. To address these gaps, an additional set of 50 samples was  
 1113 curated using **Claude 3.7 Sonnet**, specifically designed to significantly expand the emotional palette  
 1114 and sentence structural variety. This augmented set incorporates a wider spectrum of subtle and  
 1115 complex affective states, embedded within richer narrative contexts that provide stronger implicit cues  
 1116 for the target prosodic realization. The resulting 70-sample dataset thus offers enhanced evaluative  
 1117 robustness, enabling a more rigorous assessment of a TTS system’s ability to synthesize expressive  
 1118 dialogues. The prompt used for breadth expansion is:

```

1119 Consider the below set of 20 samples. This set belongs to the "Emotions" category
1120 and you will use it to create an extremely diverse set for evaluation of TTS
1121 systems, where the systems have to synthesize the text corresponding to **
1122 text_to_synthesize**.
1123 This category assesses whether the system expresses emotions naturally, using
1124 variations in pitch, loudness, and rhythm. A good TTS system should reflect
1125 emotions like excitement, sadness, or frustration (and others) as they appear
1126 within quotes"" in **text_to_synthesize**, and there are many other emotions/
1127 corresponding context that can be used to evaluate a TTS system which may not
1128 be covered in the 20 samples.
1129 Your goal is to generate 50 more samples belonging to this category. You will do
1130 this in the following step-by-step manner:
1131 1. You will analyze the 20 samples carefully.
1132 1.a. Reason deeply about the types of emotions covered by this set.
1133 1.b Reason deeply about the **text_to_synthesize** structures present, like the
1134 placement of the quoted dialogue, the number of words inside the quoted
1135 dialogue, the number of words outside the quoted dialogue, the number of quoted
1136 dialogues, and the grammatical structure of the texts.
1137 2. Now, you will think long about what this set is **MISSING**, specifically, the
1138 various types of emotions that exist in the **COMPLETE** set of emotions
1139 present in quoted texts/dialogues, but are not present in this set, **AND**
1140 will be great to test emotion expressiveness synthesis ability of TTS systems.
1141 3. Finally, you will create additional 50 samples, that expand the current 20 sample
1142 set in terms of **DIVERSITY**, as you are doing a breadth-wise evolution of
1143 this 20 set.
1144 
```



```

1145
1146 The main goals for the set are:
1147 1. The emotion should be strongly inferable from the context, as the TTS system
1148    will not be explicitly provided (like with a special tag) the emotion needed for
1149    the dialogue.
1150 2. All text_to_synthesize that will be generated in the 50 samples, should form a
1151    fluent sentence that could naturally be seen in a story book, or when someone
1152    is narrating something.
1153 3. There should be diversity in terms of placement position of quoted dialogue, and
1154    the length of the context.
1155 4. Each sample must have ATMOST 2 quoted dialogues, in most cases just 1 quoted
1156    dialogue. (verify this with the initial 20 set)
1157 5. The 50 samples will follow the same JSONL format as the 20 samples, BUT no
1158    sample in the 50 set should be similar with each other, or the original 20 set.
1159 6. You will not create texts with * word * or ** word ** or add text inside
1160    parenthesis () to indicate something.
1161 7. The context plays a big role in emphasizing what the emotion should be, so
1162    generate rich context in all cases.
1163 8. All quoted dialogues must have at least 5 words.
1164 9. Do not use paralinguistic cues or punctuations exc
1165
1166 Now, you are given the 20 sample set, after this, think deeply and create the 50
1167    foreign words set.
1168
1169 ““jsonl
1170 <now the 20 samples were provided in jsonl format>
1171 ““

```

**Depth Refinement** We leverage the depth refinements to test TTS systems on more than just producing single, unchanging emotions. This approach checks two key things: first, how well the system changes its emotional expression when the text suggests a shift (like moving from happy to sad), and second, how realistically it can keep an emotion going or make it stronger within a single piece of dialogue, just like people do when they speak naturally. We refine the base samples to introduce increased complexity, primarily through two mechanisms: either incorporating a distinct contrasting emotional state—often signaled via brief preceding or subsequent narrative cues—or by deepening and intensifying an existing emotion within a specific dialogue segment, thereby extending the utterance. Emphasis was placed on ensuring the plausibility of these emotional arcs through natural narrative flow, and matching the existing language style of the text to ensure overly formal language is not introduced where it does not fit. Refer to refinements in Figure 8 and the prompt used is:

```

1185
1186 complex_prompt_method = ""
1187 This text_to_synthesize belongs to the "Emotions" category.
1188 This category evaluates whether a TTS system clearly and naturally expresses
1189    fundamental emotions using prosodic, rhythmic and intonational variation. The
1190    aim is to generate unambiguously evaluable dialogues testing mastery of
1191    distinct emotional expression, including sustained emotion within utterances.
1192
1193 # Your Task: Evolve the text_to_synthesize through the following steps:
1194 1. Identify ALL quoted dialogues and capture their length in the text_to_synthesize.
1195
1196 2. Choose one of the following methods to evolve the text:
1197    * Method A - DEEPENING:
1198      * Condition: The length of words in ATLEAST ONE of the quoted dialogues
1199        is LESS THAN OR EQUAL TO 20 words. If such a quote exists, identify
1200        it as a SHORT QUOTE. (If multiple SHORT QUOTES exist, choose the one
1201        where deepening the existing emotion feels most natural.)
1202      * Append new dialogue directly within the same quotes as the chosen SHORT QUOTE,
1203        extending the character's speech to reflect the
1204        INTENSIFIED/DEEPER continuation of the emotion that is already present
1205        in that SHORT QUOTE.
1206      * Do NOT add new narrative text before this appended dialogue; only
1207        append INSIDE the chosen SHORT QUOTE.

```

```

1208     * The new dialogue will be a continuation of existing dialogue, and can
1209     either continue the sentence in the dialogue by adding before full stop
1210     (.), or start a new sentence(but within the same quotes). **BE CREATIVE
1211     HERE**
1212 * **Method B - CONTRASTING:**
1213     * **Condition:** If Method A is not applicable (no quotes are <= 20 words),
1214     use Method B.
1215     * Add descriptive, natural **narrative text** (max 10 words) **either before
1216     (prefix) or after (suffix)** the original **text_to_synthesize**. This
1217     **narrative text** should also **BRIDGE** plausibly with the **quoted
1218     dialogue**.
1219     * Follow/Precede this **narrative text** with **ONE new, separate quoted
1220     dialogue** ("...") expressing the **CONTRASTING** core emotion.
1221     * The text within this **new contrasting quote** should be **at least 5
1222     words** long and **at most 15 words** long.
1223     * For clarity, 4 orders are allowed in this method:
1224         * **Order 1:** **quoted dialogue** + **unquoted narrative text** + <**
1225         text_to_synthesize**>
1226         * **Order 2:** **unquoted narrative text** + **quoted dialogue** + <**
1227         text_to_synthesize**>
1228         * **Order 3:** <**text_to_synthesize**> + **quoted dialogue** + **
1229         unquoted narrative text**
1230         * **Order 4:** <**text_to_synthesize**> + **unquoted narrative text** +
1231         **quoted dialogue**
1232
1233 3. The overall evolution must:
1234     * Introduce **ONE clearly identifiable CORE EMOTION** (focus on Joy, Sadness,
1235     Anger, Fear, Surprise, Disgust)(Method B) OR a significant intensification (
1236     deepening) of the existing emotion(Method A).
1237     * Flow **naturally and plausibly** (PRIORITY #1).
1238     * You **WILL** choose one of the characters from the narration(if there are
1239     multiple characters) and add the dialogue for that character in case of
1240     Method B. You **WILL NOT** create a new character in the narration.
1241     * You have to realize the tone of the narration, and continue the dialogue in
1242     the same tone. Do not use overly formal language where it does not fit, or
1243     too casual where it does not fit. Its **IMPERATIVE** to analyze the tone
1244     correctly.
1245
1246 # IMPORTANT CONSTRAINTS & GUIDELINES (Apply Always):
1247
1248 1. **MAXIMUM EVALUABLE CLARITY (HIGH PRIORITY):** Emotion must be **clearly
1249     identifiable** and **distinct** based on the pretext/context(if added like in
1250     Method B).
1251 2. **EMOTIONAL CUE:**(Only for Method B) Ensure strong textual cues are used to lead
1252     upto the emotion(quoted dialogue), but cues should not be too artificial or
1253     poetic, they should be natural.
1254 3. **NATURAL & SPOKEN STYLE:** Use authentic, spoken language. Avoid literary prose
1255     and use language that fits the tone of the narration.
1256 4. **PLAUSIBILITY & COHERENCE:** Evolution must be believable. Addition must feel
1257     grounded.
1258 5. **STRUCTURAL RULES:** Follow Method A/B strictly. Do not change original text's
1259     **EMOTION**.
1260 6. **NO TEXT MARKERS:** **DO NOT** add additional text markers like the characters
1261     " " or " " or any other text markers in the **rewritten_text_to_synthesize**.
1262 7. **Transitional Text:** The transition between narrative text and quoted dialogue
1263     **MUST** be natural.
1264 8. **Grammatical Correctness:** The **rewritten_text_to_synthesize** must be
1265     grammatically correct.
1266 # Output Explanation Fields:
1267
1268 ## In the **tts_synthesis_diversity** field, you must provide the following
1269     information:
1270     1. **Method Choice Rationale:** Explain *why* you chose Method A or Method B and
1271         subsequently if the added text will be emotionally **DEEPENING** or **

```

1272       CONTRASTING\*\*. (If Method A, state which SHORT QUOTE was chosen if multiple  
1273       existed).

1274   2. \*Recognize the tone of the narration, and mention it here, this is the tone  
1275       (\*\*with slight variations accepted\*\*) that you have to continue in the  
1276       rewritten dialogue.\*

1277   3. \*(Only if Method B was used):\* Think about plausible \*\*narrative text\*\* and  
1278       \*\*quoted dialogue\*\* to create emotional arcs for all 4 orders.

1279       - 3.1 \*\*Order 1\*\* possible emotional arc.  
1280       - 3.2 \*\*Order 2\*\* possible emotional arc.  
1281       - 3.3 \*\*Order 3\*\* possible emotional arc.  
1282       - 3.4 \*\*Order 4\*\* possible emotional arc.  
1283       - 3.5 Identify if applying \*\*Order 2\*\* or \*\*Order 3\*\* may result in \*\*  
1284       rewritten\_text\_to\_synthesize\*\* containing two \*\*consecutive quoted  
1285       dialogues\*\*. If \*\*text\_to\_synthesize\*\* contains quoted dialogues at  
1286       boundaries this may happen and you will \*\*ELIMINATE\*\* problematic orders.  
1287

1288       - 3.6 Based on the analysis of all 4 emotional arcs and eliminating the  
1289       invalid orders, choose the best out of the valid orders and justify your  
1290       choice.

1291   4. The \*\*specific CORE EMOTION\*\* we can add(Method B) or significantly deepen(  
1292       Method A, this is where you will specify which emotion is being deepened).

1293   5. \*(Only if Method A was used):\* Specify the \*\*continuation\*\* or \*\*new sentence  
1294       \*\* that you will add inside the quotes of \*\*SHORT QUOTE\*\* to intensify the  
1295       emotion.

1296   6. \*(Only if Method B was used):\* The \*\*narrative text\*\* and the \*\*quoted  
1297       dialogue\*\* you will add according to the best order you chose in point 3.6.

1298   7. \*(Only if Method B was used):\* Identify any \*\*MINOR PARAPHRASING\*\* of the \*\*  
1299       text\_to\_synthesize\*\* to accomodate \*\*narrative text\*\* and \*\*quoted dialogue  
1300       \*\* you added in point 6.

1301   8. Ensure that the \*\*rewritten\_text\_to\_synthesize\*\* to be created follows \*\*  
1302       IMPORTANT CONSTRAINTS & GUIDELINES\*\*, it does not contain any "\*" or "\*\*\*"  
1303       markers, and is \*\*grammatically correct\*\*.

1304   9. Ensure that all the decisions you made make for a better TTS evaluation than  
1305       the original \*\*text\_to\_synthesize\*\* for emotional expressiveness.

1306   10. Be CAREFUL with the handling of quotes in the output json object, always  
1307       escape the quotes in the \*\*rewritten\_text\_to\_synthesize\*\* field in the  
1308       output json object. Every opened quote must have a closing quote.

## 1310 A.5 Category 5: Syntactic Complexity

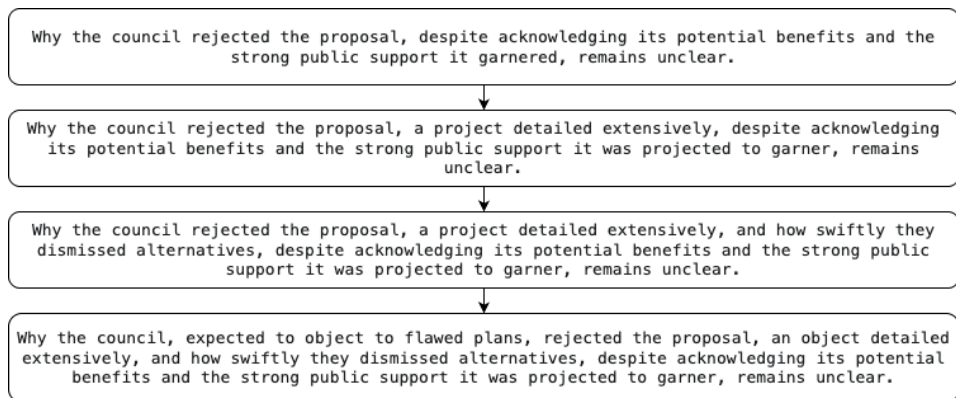


Figure 9: Example depth-refinement for syntactic complexity category. Initially, it presents a multi-clause sentence requiring clear phrasing. Subsequent refinements introduce further complexity: first with an embedded appositive, then a coordinated dependent clause increasing structural intricacy. The final stage incorporates an additional non-restrictive clause and, critically, the homograph "object"-used as both a verb and a noun-to assess the TTS's ability to disambiguate via pronunciation and stress within a highly embedded structure.

**Breadth Expansion** The initial 20 samples effectively tested TTS prosody on deep center-embedding and long subject-verb dependencies but notably omitted other crucial structures reliant on prosody, such as inversion, cleft sentences, ellipsis (gapping), complex clausal subjects (Wh-/That-/gerunds), and nuanced punctuation cues (semicolons, dashes). The 50 samples curated through breadth expansion using **Gemini-2.5-pro** rectify these specific omissions by introducing robust examples across these categories, significantly broadening the structural diversity and creating a more comprehensive benchmark for evaluating a TTS system’s handling of complex syntax. The resulting 70-sample dataset therefore provides a more robust and syntactically varied test suite for assessing the prosodic competence of TTS systems when faced with intricate grammatical structures. The breadth expansion prompt is as follows:

```

1321 Consider the below set of 20 samples. This set belongs to the "Syntactic
1322 Complexity" category and you will use it to create an extremely diverse set
1323 for evaluation of TTS systems, where the systems have to synthesize the text
1324 corresponding to text_to_synthesize.
1325 This category evaluates how well the system uses prosody (pausing, phrasing,
1326 intonation, stress) to make complex sentence structures easily understandable.
1327 It tests if the TTS clearly conveys the intended grammatical relationships,
1328 especially with nested clauses, ambiguities, or long dependencies. There are
1329 other forms of syntactic complexity that can be used to evaluate a TTS system
1330 which may not be covered in the 20 samples.
1331 Your goal is to generate 50 more samples belonging to this category. You will do
1332 this in the following step-by-step manner:
1333 1. You will analyze the 20 samples carefully.
1334 1.a. Reason deeply about the types of syntactic complexities present.
1335 2. Now, you will think long about what this set is MISSING, specifically, the
1336 various types of syntactic complexities that exist in the COMPLETE set
1337 english sentences, but are not present in this set, AND will be great to
1338 test complex grammar synthesis ability of TTS systems.
1339 3. Finally, you will create additional 50 samples, that expand the current 20 sample
1340 set in terms of DIVERSITY, as you are doing a breadth-wise evolution of
1341 this 20 set.
1342 The main goals for the set are:
1343 1. We want to include all types of syntactic complexities, that are ATLEAST as
1344 complex as the ones present in the 20-sample set.
1345 2. The 50 samples will follow the same JSONL format as the 20 samples, BUT no
1346 sample in the 50 set should have phrasing that gives raise to the SAME
1347 syntactic complexity as the 20 sample set. Our goal is to create a DIVERSE
1348 set.
1349 3. You will not create texts with * word * or ** word ** or add text inside
1350 parenthesis () to indicate something.
1351 Now, you are given the 20 sample set, after this, think deeply and create the 50
1352 syntactic complexity set.
1353 jsonl
1354 <now the 20 samples were provided in jsonl format>
1355 '''
1356

```

**Depth Refinement** While breadth-wise expansion verifies coverage across diverse syntactic phenomena, depth-wise refinement is crucial for assessing a TTS system’s robustness and performance scalability when faced with escalating grammatical intricacy. This approach tests the system’s ability to manage compounded syntactic load and maintain prosodic coherence under increasing structural demands, rather than merely handling isolated complexities. Our refinement strategy involved iteratively enhancing base complex sentences by applying targeted syntactic transformations-such as introducing complex coordination, structural reordering (e.g., fronting, passivization impacting dependencies), complicating ellipsis, or adding layered subordination-while strictly enforcing constraints on grammatical correctness and naturalness. The refinement is also encouraged to add two words that are homographs of each other if it fits naturally in the context, this tests the ability of the TTS to disambiguate the meaning from context and correctly pronounce it. The resulting dataset provides a graded challenge, enabling evaluation of how TTS prosody adapts to and conveys meaning

1372 across incrementally complex, yet plausible, sentence structures. Refer to Figure 9 for an example  
1373 refinement, and the prompt used is:

```
1374 complex_prompt_method = ""
1375
1376 You are tasked with enhancing the syntactic complexity of the provided **
1377 text_to_synthesize**, which belongs to the "Syntactic Complexity" category.
1378 This category tests a TTS system's ability to render complex grammar clearly
1379 through prosody, pauses, intonation and stress patterns.
1380
1381 Your specific task is to rewrite the **text_to_synthesize** by performing the
1382 following steps:
1383
1384 1. **Analyze the existing sentence structure** and identify opportunities for
1385 enhancement.
1386 2. **Enhance syntactic complexity using ONE appropriate technique.** Select a
1387 technique that fits naturally and logically, aiming for variety across
1388 evolutions. **Give special consideration to techniques beyond simple clause/
1389 phrase addition, such as:**
1390 * Introducing or complicating **coordination** (of long/dissimilar phrases or
1391 clauses).
1392 * Introducing or complicating **gapping/ellipsis** (omission of repeated
1393 elements, esp. verbs).
1394 * Restructuring to create **scope ambiguity** resolvable by prosody (e.g.,
1395 involving negation, quantifiers).
1396 * Employing **structural reordering** (e.g., complex fronting, heavy subject/
1397 object shifts, passivization impacting dependencies).
1398 * Adding/elaborating **subordinate elements** (nested clauses, complex
1399 appositives, participial phrases) - *use if other options don't fit
1400 naturally*.
1401 * Increasing **modifier density** or creating **longer-distance dependencies**
1402 through restructuring.
1403 * Leveraging **punctuation** (colons, semicolons) to structure complex
1404 relationships.
1405 3. **Optional Homograph Integration:**
1406 * **IF AND ONLY IF** it fits **perfectly naturally** within the enhancement you
1407 are making, you MAY include **two words that are homographs** of each other
1408 (different standard pronunciations, contextually unambiguous).
1409 * **Do NOT use special formatting** around homographs in the **
1410 rewritten_text_to_synthesize**.
1411 4. **Complexity Goal:** The technique should aim to *appropriately enhance* the
1412 overall syntactic complexity in a **meaningful structural way**, without
1413 sacrificing clarity, naturalness, or grammatical correctness.
1414 5. **CRITICAL CONSTRAINT - Length, Naturalness, Grammar & Coherence:**
1415 * You will increase the length of the **text_to_synthesize** by **ATLEAST 2
1416 words** and **ATMOST 6 words**.
1417 * Your **absolute priority** is ensuring the rewritten text is **grammatically
1418 flawless**, sounds **natural** for complex written English, and flows
1419 logically.
1420 * It must **NOT** sound forced, artificially constructed, overly convoluted,
1421 ambiguous due to structure (unless ambiguity is the intended challenge,
1422 solvable by prosody), or like a mere linguistic puzzle.
1423 * The enhancement must integrate **smoothly and coherently**, contributing
1424 logically or structurally to the sentence. Avoid awkward interruptions or
1425 obscuring core grammatical relationships excessively.
1426 * **Prioritize naturalness/grammar over maximizing complexity.** If an
1427 enhancement feels forced, choose a simpler or different one.
1428 6. **Final Check:** Read the rewritten sentence aloud. Does it flow logically? Is
1429 the intended structure parseable? Does it introduce a relevant syntactic
1430 challenge for TTS prosody?
1431
1432 ** In the **tts_synthesis_diversity** field, you MUST provide detailed reasoning
1433 covering:
1434 1. The specific complexity enhancement technique you used (referencing the examples
1435 if applicable) and how you applied it.
```

1436 2. **\*\*If homographs were used:\*\*** Identify them and explain how the context makes  
 1437 their pronunciation unambiguous. **\*\*If not used, state that.\*\***  
 1438 3. How the change **\*\*ENHANCES\*\*** syntactic complexity, focusing on the **\*\*specific**  
 1439 structural challenge**\*\*** it poses for TTS prosody, pausing, intonation and stress  
 1440 patterns(e.g., handling ellipsis gap, coordinating long phrases, resolving  
 1441 scope, pausing timing).  
 1442 4. **\*\*Crucially, justify \*why\* the rewritten sentence remains grammatically correct,**  
 1443 **sounds natural, and integrates smoothly despite the enhancement. Explain how**  
 1444 **coherence and logical flow were maintained.\*\*** Address the critical constraint  
 1445 directly.  
 1446 5. Ensure the text does not contain any extra characters like ‘, \*, \*\*, (), etc.  
 1447 """

## 1449 A.6 Category 6: Complex Pronunciation

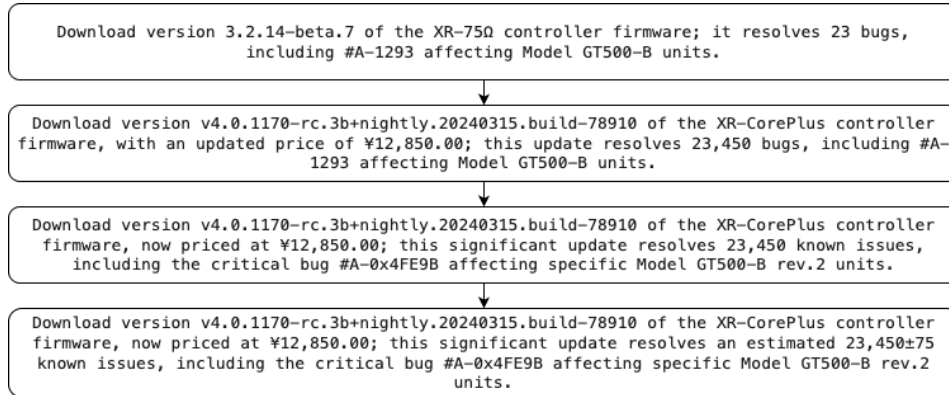


Figure 10: Example depth-refinement for complex pronunciation category. Initially, we have standard version strings and symbols. The first refinement introduces a significantly more complex, multi-component version string and a new foreign currency (¥). Then, it adds distinct challenge of correctly verbalizing hexadecimal numbers (e.g., "0x") and technical abbreviations like "rev." (revision). Finally, the third refinement assesses the system’s ability to interpret and articulate numerical uncertainty or ranges indicated by the "±" symbol.

1450 **Breadth Expansion** We create this category from scratch by prompting **Claude 3.7 Sonnet** to  
 1451 generate 60 samples, 10 from each of the following 6 categories, (i) Numerals and Currencies, (ii)  
 1452 Dates and Times, (iii) Emails, URLs and Passwords, (iv) Addresses and Location references, (v)  
 1453 STEM Notations and equations (vi) Mixed Acronyms/Initialisms. In addition to this, we add 5 short  
 1454 tongue-twisters that are repeated many times. For example, the “The Sixth Sick Sheikh’s Sixth  
 1455 Sheep’s Sick” tongue twister repeated 6 times. The prompt used is:

1456 Your goal is to generate a dataset of 60 samples. Each sample is a json line, so  
 1457 the dataset should be returned as a jsonl. Each json contains 2 keys:  
 1458 category: This will be set to "Pronunciation" text\_to\_synthesize: This is  
 1459 the key field that you will populate with great diversity.  
 1460 The goal of this dataset is to evaluate TTS systems where they have to synthesize  
 1461 whatever text you populate in text\_to\_synthesize. This is the "Pronunciations"  
 1462 category, we want to create samples with the following outline. 10 samples for  
 1463 each category(6 categories): 1. Text with currency and numerals in different  
 1464 formats 2. Text with dates and time-stamps in different formats 3. Texts with  
 1465 email addresses, passwords and urls in different formats. 4. Texts with  
 1466 complex street addresses or location references. 5. Texts with terms from the  
 1467 STEM field. 6. Texts that have both an initialism and acronym These are the two  
 1468 main goals of this dataset:  
 1469 1. Breadth: It should cover huge diversity in sentence structure, sentence length.  
 1470 2. Depth: This dataset stress-tests the pronunciation capability of TTS systems.  
 1471 Before generating the dataset, reason step-by-step and deeply on how you will  
 1472 cover the breadth and depth aspects, then give 60 jsonl lines  
 1473

1475 **Depth Refinement** In this category, the goal of depth refinement is to progressively add complex,  
 1476 hard-to-pronounce elements to an utterance, while keeping them within the same sub-category as  
 1477 the original. With each refinement, we aim to increase the density of such elements. To avoid  
 1478 repeating the same pronunciation challenges, we prompt the refining LLM to suggest three novel  
 1479 ways to introduce complexity - distinct from what's already present. Specifically, we ask for elements  
 1480 that are likely to challenge TTS systems, even if other parts are rendered correctly. This approach  
 1481 consistently produces utterances with multiple challenging components that TTS systems may  
 1482 struggle to synthesize. We do not apply this strategy for the tongue-twisters and keep them as is. An  
 1483 example refinement is given in Figure 10 and the prompt used is:

```

1484 complex_prompt_method = """
1485
1486 The text_to_synthesize belongs to the "Pronunciation" category.
1487 This category evaluates how well the system pronounces non-trivial words,
1488 numerals and other special characters.
1489 Your primary goal is to increase the pronunciation complexity of text_to_synthesize, creating a rewritten_text_to_synthesize.
1490
1491
1492 First, the text_to_synthesize will fall into one of these 6 methods/
1493 categories:
1494 1. Numerals & Currency: Focuses on the presence of numbers and currencies in
1495 varied formats.
1496 2. Dates & Times: Contains dates, times, durations, and time zones in varied
1497 formats.
1498 3. Emails, URLs & Passwords: Features communication/resource locators (
1499 emails, URLs, passwords, IPs, phone numbers, etc.).
1500 4. Addresses & Locations: Describes physical locations (street addresses
1501 with components like numbers/suffixes/directions/abbreviations, coordinates,
1502 etc.).
1503 5. STEM Notations: Characterized by scientific/math formulas, complex
1504 notations, specialized units, etc.
1505 6. Mixed Acronyms/Initialisms: Includes BOTH spelled-out abbreviations (
1506 letter-by-letter) and pronounced ones (as words) in the same sentence.
1507
1508 The provided text_to_synthesize belongs to Category {{{pronunciation_sub_category}}}. Your evolution must remain focused within
1509 this category WHILE ensuring that the rewritten_text_to_synthesize will
1510 not fall into any other category.
1511
1512
1513 Now, evolve the complexity:
1514 - Modify the text_to_synthesize by introducing NOVEL elements that will
1515 increase the pronunciation complexity of the text.
1516 - You may paraphrase the original text slightly for natural flow, but CRITICAL: ensure the specific complex elements from the text_to_synthesize are preserved in the rewritten_text_to_synthesize unless they are being directly merged or replaced by the new, more complex elements.
1517
1518 - Focus on introducing different facets, new dimensions, or more intricate variations of complex elements to do the evolution.
1519
1520 - CRITICAL: Do NOT simply add a different, unrelated element type, even if it falls under the same broad category number. The goal is to make the existing type of challenge harder, not to dilute it with a separate challenge.
1521
1522
1523 In the ttts_synthesis_diversity field, provide your analysis:
1524 1. Briefly explain the complex elements present in the text_to_synthesize that will particularly challenge the TTS system, confirming its alignment with Category {{pronunciation_sub_category}}.
1525
1526 2. Reason about MULTIPLE DIFFERENT complex elements we may introduce to the text_to_synthesize that will increase the pronunciation complexity.
1527 2.1 For each candidate complex element, reason how the TTS system may fail to pronounce it EVEN IF all existing complex elements present in the text_to_synthesize are pronounced correctly.
1528
1529 3. Reason which complex elements you came up with in step 2 will be the most effective to increase the pronunciation complexity of the text in a novel
  
```

```

1539         ** way. Select **EXACTLY 1 element** and call it the **chosen complex
1540         element**.
1541     4. What will be the ideal pronunciation for the **chosen complex element** by
1542         the TTS system?
1543     5. Identify the best way to introduce the **chosen complex element** in the **
1544         text_to_synthesize** to increase the pronunciation complexity of the text.
1545     6. Confirm that the introduction of the **chosen complex element** won't make
1546         the text artificial or unnatural or domain-inappropriate.
1547     """

```

## 1549 A.7 Final dataset statistics

Refer to Table 4 for final category-wise statistics.

Category	Table 4: Final Dataset Statistics								
	No. Characters			No. Words			Audio Length (s)		
	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max
Questions	16	248.22	701	3	41.61	120	0.90	15.04	48.45
Foreign Words	71	136.85	242	9	21.77	39	4.80	9.07	16.20
Paralinguistics	28	127.36	319	5	19.30	49	2.15	9.23	22.30
Emotions	102	340.04	676	18	57.58	107	6.15	21.80	45.20
Syntactic Complexity	45	194.71	366	8	28.23	64	3.25	12.53	23.60
Complex Pronunciation	104	260.35	920	8	35.28	139	8.45	25.56	94.70
Overall	16	217.02	920	3	33.93	139	0.90	15.32	94.70

1550

## 1551 B MMAU performance for LALMs

1552 MMAU [31] is an audio-reasoning benchmark, testing audio understanding models for reasoning  
1553 across 3 categories, Speech, Music and Sounds. It evaluates 27 distinct skills through information  
1554 extraction and reasoning tasks that require advanced reasoning, such as multi-speaker role mapping,  
1555 emotional shift detection, and temporal acoustic event analysis. We run the evaluation ourselves  
1556 on the test-mini subset of 1,000 samples for top closed-source LALM models, and the results are  
summarized in Table 5.

Table 5: LALM performance on Audio Reasoning MMAU benchmark, test-mini subet of 1000 samples

Model	Test-mini score $\uparrow$
Gemini 2.5 Pro	<b>68.60</b>
Gemini 2.5 Flash	65.20
Gemini 2.0 Flash	62.10
Gpt-4o-audio	59.20
Gpt-4o-mini-audio	59.80

1557

## 1558 C Evaluation-related Details

### 1559 C.1 Hyper-parameters

1560 **Data Depth Refinement:** Gemini 2.5 Pro is prompted with *temperature* = 1.0 for creativity,  
1561 *top\_p* = 0.9 and *max\_output\_tokens* = 16384 when doing depth refinement for 3 steps.

1562 **Audio Generation:** Closed source TTS models like Aura-2, Eleven Multilingual v2, HumeAI and  
1563 gpt-4o-mini-tts do not support a temperature parameter. For Sesame1B, Qwen2.5 Omni, gpt-4o-mini-  
1564 audio-preview and gpt-4o-audio-preview we use a *temperature* = 1.0, and for orpheus tts, we use



1565 the recommended values of *temperature* = 0.6 and *top\_p*=0.8. We set the maximum output tokens  
1566 to 8192 and ensure that for no system, the audio is being clipped.

1567 **Judger LALMs:** For the judger, we set *temperature* = 0.0 for reproducibility, we find that  
1568 while Gemini results are not deterministic even after setting a value of 0.0, the final win-rate does not  
1569 change significantly across runs, < 1% change to be specific. The max output length is set to 131072  
1570 for the thinking models (Gemini 2.5 series), and 16384 for other judgers we use for ablation.

## 1571 C.2 Prompts for Audio Generation

1572 From the description map presented below, we select and send the description relevant to the specific  
1573 category when using **Strong Prompting** with Hume AI and gpt-4o-mini-tts:

```
1574 ALL_DESCRIPTIONS = {  
1575     "Emotions": "Emotional expressiveness: Ensure a clear and distinct transition  
1576         between quoted dialogues and narrative text. Deliver the quoted dialogues  
1577         with high emotional expressiveness.",  
1578     "Paralinguistics": "Paralinguistical cues: Express interjections, onomatopoeia,  
1579         capitalization, vowel elongation, hyphenation/syllable stress, stuttering  
1580         and pacing cues(elipses, punctuations, etc.) naturally and realistically.",  
1581     "Syntactic Complexity": "Syntactical Complexity: Maintain clarity in complex  
1582         sentence structures through appropriate prosody, pausing and stress to  
1583         convey the intended meaning of the sentence very clearly. Handle homographs  
1584         with appropriate pronunciation.",  
1585     "Foreign Words": "Foreign words: Pronounce foreign words and phrases with their  
1586         appropriate pronunciation or anglicized version, sound like a natural bi-  
1587         lingual speaker doing smooth code-switching.",  
1588     "Questions": "Questions: Apply the appropriate intonation pattern for  
1589         interrogative sentences(questions) and declarative sentences.",  
1590     "Pronunciation": "Complex Pronunciation: Pronounce currency, numerals, emails,  
1591         passwords, urls, dates, times, phone numbers, street addresses, equations,  
1592         initialisms, acronyms, tongue twisters(speak fast while maintaining clarity),  
1593         etc. with precision, clarity and case-sensitivity wherever applicable."  
1594 }  
1595 }
```

1597 Following are the templates we use for normal prompting and strong prompting scenarios with  
1598 LALMs like Qwen 2.5 Omni, gpt-4o-mini-audio-preview and gpt-4o-audio-preview.

### 1599 Normal Prompt:

```
1600 USER_MESSAGE_DEFAULT_TEMPLATE = ""  
1601   
1602     Your goal is to synthesize speech that exactly matches the text under **  
1603     text_to_synthesize** tag.  
1604     You will be provided with the **text_to_synthesize**, after that generate **ONLY  
1605     ** the **VERBATIM** speech matching the text. Do not add any additional  
1606     information or text in your response.  
1607     ***text_to_synthesize***:  
1608     {{{text_to_synthesize}}}  
1609     ""
```

### 1611 Strong Prompt:

```
1612 USER_MESSAGE_STRONG_TEMPLATE = ""  
1613   
1614     Your goal is to synthesize speech that exactly matches the text under **  
1615     text_to_synthesize** tag.  
1616     The generation has to be human-like and realistic. To excel in this task, you  
1617     must pay attention to the following aspect of the text:  
1618     {{{descriptions}}}  
1619     Now, you will be provided with the **text_to_synthesize**, after that generate  
1620     **ONLY** the **VERBATIM** speech matching the text. Do not add any  
1621     additional information or text in your response.  
1622     ***text_to_synthesize***:  
1623     {{{text_to_synthesize}}}  
1624     ""
```

1626 The {{{descriptions}}} placeholder is replaced with the specific description of that category, as  
1627 mentioned in the ALL\_DESCRIPTIONS map.

### 1628 C.3 Prompts for Judge and category-wise evaluation criteria

1629 The judge is provided with the following prompt template:

```
1630 USER_MESSAGE_WIN_RATE = ""Your goal is to judge two TTS(text-to-speech) systems and
1631 analyze which system synthesizes speech corresponding to a particular text
1632 better than the other one and determine the winner based on the scoring
1633 criterion.
1634 You will rate each system a score between 0 and 3 based on how well it
1635 synthesizes speech corresponding to a particular text called **
1636 text_to_synthesize**, then do their comparative analysis and provide your
1637 final judgement.
1638 A good system will generate speech that sounds realistic and human-like, and it
1639 captures the specific nuances of the text.
1640
1641 You will be provided with the **text_to_synthesize** which is the text both TTS
1642 systems had to synthesize,
1643 the **text_category** and the **evaluation_criterion** corresponding to the **
1644 text_category**, in which you will be made aware of the **evaluation
1645 dimension** you will focus on, and the **scoring criteria** you will use to
1646 score the TTS systems.
1647 You will also be provided with the **output_format**, which dictates the format
1648 of the output you need to follow as a judge.
1649 Finally, you will be provided with the synthesized speech from the TTS system 1
1650 **synthesized_speech_1** and then from TTS system 2 **synthesized_speech_2**.
1651
1652
1653
1654 **text_to_synthesize**
1655 {{{text_to_synthesize}}}
1656
1657
1658 **text_category**
1659 {{{text_category}}}
1660
1661
1662 **evaluation_criterion**
1663 {{{evaluation_criterion}}}
1664
1665 NOTE: If the generated speech is very poor and does not synthesise the text
1666 correctly, you will provide a score of 0 to that TTS system.
1667 GLOBAL CONSIDERATIONS(**VERY IMPORTANT FOR COMPARISON**):
1668 - It is imperative to compare the two systems **ONLY** on the basis of the **
1669 evaluation_dimension**, that means, you **WILL NOT** let the following
1670 types of **BIASES** affect your judgement:
1671 - The acoustical quality of the audio, background noise or clarity.
1672 - The gender and timbre features of the speaker.
1673 - Any other factors that are not related to the **evaluation_dimension**.
1674 - Systems demonstrating exaggerated expressiveness should not be rewarded
1675 more **UNLESS** those features are relevant to the **
1676 evaluation_dimension**.
1677 - Tie-break procedure
1678 1. If the overall score_1 and score_2 are equal, use this protocol.
1679 2. For the chosen **evaluation_dimension**, inspect every comparable
1680 component:
1681 - Note similarities.
1682 - Note differences and label each as:
1683 - Subtle: hardly noticeable to a typical human listener.
1684 - Significant: clearly influences human perception.
1685 3. Count the significant differences that benefit each system.
1686 4. Decision:
1687 - No significant differences, or counts are equal -> declare a tie.
```

```

1688         - Otherwise -> declare the system with the higher count of
1689             significant advantages the winner.
1690
1691     **output_format**
1692     You will output a json dictionary as follows:
1693     ```json
1694     {
1695         "reasoning_system_1": str = Reasoning chain based on the **Reasoning guidelines
1696             : ** for the synthesized speech from TTS system 1.
1697         "reasoning_system_2": str = Reasoning chain based on the **Reasoning guidelines
1698             : ** for the synthesized speech from TTS system 2, **INDEPENDENT** of the
1699             performance of TTS system 1.
1700         "system_comparison": str = Keeping in mind the GLOBAL CONSIDERATIONS, compare
1701             and contrast the two systems based on your output in reasoning_system_1 and
1702             reasoning_system_2 and also by analyzing both audios again. Provide very
1703             fine-grained reasoning for which system won, or if the comparison results in
1704             an even tie.
1705         "score_1": int = Your score for the synthesized speech from TTS system 1 between
1706             0 and 3, based on the **evaluation_criterion** and what you have mentioned
1707             in reasoning_system_1.
1708         "score_2": int = Your score for the synthesized speech from TTS system 2 between
1709             0 and 3, based on the **evaluation_criterion** and what you have mentioned
1710             in reasoning_system_2.
1711         "winner": int = The winner of the comparison between TTS system 1 and TTS system
1712             2. Output 1 if TTS system 1 wins, 2 if TTS system 2 wins, output 0 if this
1713             will be considered as an even tie.
1714     }
1715     - Note: Ensure the json structure is followed and the json output **MUST** be
1716         parsable without errors. (For example, escape the quotes wherever you add
1717         them inside a field of the json, all brackets and braces should be correctly
1718         paired.)
1719
1720     Now you will be provided with the synthesized speech from the TTS system 1,
1721     please analyze it carefully.
1722
1723     **synthesized_speech_1**
1724     """

```

1726 After the above prompt, we append the audio from System 1, then we have the post audio 1 prompt,  
1727 this design choice is adopted to provide effective separation between the two audios.

```

1728 POST_AUDIO_1_MESSAGE = """
1729 Now you will be provided with the synthesized speech from the TTS system 2, please
1730 analyze it carefully. After that provide the judgment following the **
1731 output_format** ensuring parsability.
1732 **synthesized_speech_2**
1733 """

```

1736 The placeholder {{{evaluation\_criterion}}} is replaced with the specific criteria for that category, as  
1737 described in the map below:

```

1738 CATEGORY_TO_CRITERION_MAP = {
1739     "Questions" : """
1740         **Evaluation Dimension:**
1741         - In this category, we want to evaluate the ability of the TTS system to
1742             apply correct intonation patterns: Interrogative for questions,
1743             declarative for statements, etc.
1744         - Questions usually have a distinct pitch movement, often rising at the
1745             end in yes/no questions, while wh-questions may have a more neutral
1746             or falling tone.
1747         - Statements between questions should have an intonation pattern that
1748             differentiates them from the questions and makes it clear that it is
1749             a statement.
1750     """

```

1751           - You have to be careful that texts can have multiple correct intonation  
1752           patterns, so place appropriate weight on parts where intonation is  
1753           not very clear.  
1754

1755 **\*\*Example:\*\*** "Did you see the message? Well I hope you did. But please tell  
1756           me that you actually did?"  
1757 **\*\*Explanation:\*\***  
1758           - There maybe multiple correct patterns to render this speech with, but  
1759           we want to judge if the TTS system has made an attempt at correctly  
1760           conveying the interrogative intonation for the 2 questinos, and the  
1761           declarative intonation for the statement between the questions.  
1762

1763 **\*\*Note:\*\***  
1764           - The **\*\*text\_to\_synthesize\*\*** may contain multiple questions without or  
1765           without the question mark, you have to correctly differentiate  
1766           between the questions and the statements.  
1767

1768 **\*\*Rating Scale:\*\***  
1769 1: All intonation patterns incorrect  
1770 2: Some intonation patterns are largely correct but some are incorrect  
1771 3: All intonation patterns are correct and convey the question nature  
1772       perfectly  
1773

1774 **\*\*Reasoning guidelines:\*\***  
1775       1. Mention which parts need to be rendered with interrogative intonation  
1776       and which with declarative intonation.  
1777       2. Carefully list the crucial parts of the speech, the pertinent  
1778       syllables and their precise timestamps.  
1779       3. Analyze the audio multiple times to capture the intonation patters at  
1780       the crucial parts.  
1781       4. Finally, reason deeply and justify how the TTS system has performed  
1782       and applied the intonation patterns at the crucial parts, and then  
1783       what the final score for the TTS system should be.  
1784       """,  
1785

1786 **"Emotions" :** ""  
1787 **\*\*Evaluation Dimension:\*\***  
1788       - In this category, we want to evaluate the ability of the TTS system to  
1789       express emotions naturally, using variations in pitch, loudness,  
1790       rhythm, etc. and demonstrate tone variations between the quoted  
1791       dialogues and the narrative text.  
1792       - The TTS system has to generate speech as if it is narrating the **\*\***  
1793       **text\_to\_synthesize\*\***, which means showing natural and strong  
1794       emotional expressiveness for the quoted dialogues.  
1795

1796 **\*\*Example:\*\*** "Full of joy, he exclaimed: "I can't believe it! This is  
1797       amazing!". But then, a sudden realization dawned on him and he said "  
1798       Okay okay wait wait, I think this may not be such a good idea after all  
1799       ."  
1800 **\*\*Explanation:\*\***  
1801       - The text inside the first quotes "I can't believe it! This is amazing!"  
1802       should sound excited and joyful, not robotic.  
1803       - The text inside the second quotes "Okay okay wait wait, I think this  
1804       may not be such a good idea after all." should sound disappointed and  
1805       frustrated and this contrasting emotion should be clearly noticeable  
1806       .  
1807       - The narrative between/around the quotes should be distinct than the  
1808       dialogues and should be spoken with the appropriate narrative tone.  
1809

1810 **\*\*Rating Scale:\*\***  
1811 1: Fails to express emotions in the quoted dialogues, and the transition  
1812       between the quoted dialogues and the narrative is flat and not distinct.  
1813 2: Synthesises some quoted dialogues with emotions but fails to synthesise  
1814       others, OR, the rendered emotions are not very natural and emphatic, OR,

the tone bridging quoted dialogues and the narrative text cannot be distinguished/is barely discernible.

3: Synthesises all quoted dialogues with natural and emphatic emotions, and the tone bridging quoted dialogues and the narrative text is clearly distinguishable.

**Note:**

- The **text\_to\_synthesize** will not explicitly state the emotion for the quoted dialogues, you have to infer that from the context.

**Reasoning guidelines:**

- Identify the emotional state in which the all the quoted dialogues should be spoken based on the context, identify the intensifying and contrasting emotions.
- Provide precise timestamps of **EVERY** crucial part of the quoted dialogue, and comment on the emotional expressiveness of these parts that are important to convey the **OVERALL** emotional tone of the dialogue.
- Analyze the boundary points, where quoted dialogues and narrative context meet, and provide precise timestamps of these parts, while reasoning how there may be a change in the emotional tone of the speech at these points.
- Finally, reason deeply and justify how the expressive the TTS system is, and how it has narrated the **text\_to\_synthesize**, and then what the final score for the TTS system should be.

"" ,

"Syntactic Complexity" : ""

**Evaluation Dimension:**

- In this category, we want to evaluate the ability of the TTS system to use **prosody** (pausing, phrasing, intonation, stress) to make complex sentence structures easily understandable.
- It tests if the TTS can convey a syntactically very complex sentence such that it's meaning to the listener is clear and understandable, that is the main goal.
- Occasionally, the text may contain homographic words, in that case, the TTS system should pronounce the homographic words with appropriate pronunciation.

**Example:** "The book that the professor who won the award wrote is on the table."

**Explanation:**

- Without proper phrasing and intonation, it's hard to follow who did what or to identify the main subject ("the book") and the verb ("is").
- The rest of the sentence-"that the professor who won the award wrote"-is a complex noun modifier (a series of nested relative clauses) describing "the book."
- The core structure of the sentence is: "The book is on the table."
- A TTS system must use appropriate prosody-pausing, stress, and intonation-to guide the listener naturally through the structure, signaling the main subject, distinguishing the embedded clauses, and connecting all parts coherently.

**Note:**

- This category is all about adding appropriate pauses, stress, and intonation, in absence of punctuation marks, **AND** in their presence too. We want to check if the intended meaning is conveyed correctly and that is all that matters.

**Rating Scale:**

- The prosody makes the sentence structure confusing or leads to an incorrect meaning.
- The intended structure is mostly understandable, but the prosody (pauses, intonation, stress) sounds unnatural or confusing at some parts.

1880 3: The prosody correctly conveys the sentence structure, making the complex  
1881 grammar easy to follow and clarifying the intended meaning of the  
1882 sentence very clearly.  
1883

1884 **\*\*Reasoning guidelines:\*\***  
1885 1. Elaborate the intended meaning of the sentence and untangle the complex  
1886 syntax.  
1887 2. Identify the syntactically complex parts of the speech that require  
1888 appropriate prosody (pausing, phrasing, intonation, stress) to be  
1889 understandable, also identify any homographs and their intended  
1890 pronunciation, finally list all these crucial parts.  
1891 3. Carefully analyze and provide precise timestamps of crucial prosodic  
1892 features - pauses between phrases, changes in intonation, and stress  
1893 patterns - that help clarify the sentence structure for each of the  
1894 crucial parts.  
1895 4. Evaluate how well the prosody helps to distinguish the meaning at these  
1896 crucial parts, for example, distinguish between main clauses and  
1897 subordinate clauses, avoid garden path effects, and other syntactic  
1898 complexities(including homographs) identified in 2.  
1899 5. Finally, reason deeply and justify how effectively the TTS system's  
1900 prosodic features (or lack thereof) contribute to the comprehensibility  
1901 of the **\*\*OVERALL\*\***complex syntax, and then determine the final score for  
1902 the TTS system.  
1903 "" ,  
1904  
1905 "Foreign Words" : ""  
1906 **\*\*Evaluation Dimension:\*\***  
1907 - In this category, we want to evaluate the ability of the TTS system to  
1908 correctly pronounce foreign words and phrases, either using their  
1909 original pronunciation or a widely accepted anglicized version.  
1910 - The goal for the system is to sound like a fluent bi-lingual speaker,  
1911 seamlessly doing code-switching between the languages.  
1912  
1913 **\*\*Example:\*\*** "During his shaadi, manoj went pura paagal and started dancing  
1914 jaise ki wo ek actor hai."  
1915 **\*\*Explanation:\*\***  
1916 - The words "shaadi", "paagal", "jaise" and "actor" should be pronounced  
1917 with an acceptable hindi pronunciation(as there is no anglicized  
1918 version for these words).  
1919 - The flow when switching between the two languages should be seamless  
1920 and natural, without awkward pauses or jumps.  
1921  
1922 **\*\*Note:\*\***  
1923 - Not all foreign words have an anglicized version, in that case the  
1924 words should be pronounced with an acceptable pronunciation in that  
1925 foreign language.  
1926  
1927 **\*\*Rating Scale:\*\***  
1928 1: Pronounces the foreign words and phrases completely incorrectly.  
1929 2: Applies foreign pronunciation but not entirely correctly, some words are  
1930 pronounced correct but others are not and the natural flow during code-  
1931 switching is disrupted.  
1932 3: Correct rendering in the intended language or acceptable anglicized  
1933 version for all words and phrases, and the natural flow during code-  
1934 switching is maintained.  
1935  
1936 **\*\*Reasoning guidelines:\*\***  
1937 1. Identify the foreign words and phrases, and the language they belong  
1938 to.  
1939 2. Provide precise timestamps for **\*\*ALL\*\*** the foreign words and phrases  
1940 in the speech.  
1941 3. Analyze the audio multiple times, and provide a comment on the  
1942 pronunciation of the foreign words, and if the system has gotten none  
1943 , some or all of them correct.

```

1944         4. Finally, reason deeply and justify how the TTS system has performed
1945           based on pronunciation **AND** the flow at code-switching points, and
1946           then what the final score for the TTS system should be.
1947     """,
1948
1949     "Paralinguistics" : ""
1950     **Evaluation Dimension:**
1951         - In this category, we want to evaluate how well the TTS system synthesizes
1952           speech corresponding to paralinguistic cues present in the text.
1953           There can be multiple types of paralinguistic cues present in the
1954           text, like:
1955         - Interjections ("Hmmm", "Ooops").
1956         - Vocal sounds/onomatopoeia("Shhh!", "Achoo!", "Meow")
1957         - Emphasis using CAPS("He didn't REALLY mean it" has a different
1958           sound than "He didn't really MEAN it"), vowel elongation("
1959           Heyyyyyyy, okayyyyyyy"), hyphenation/syllable stress("ab-so-
1960           lutely", "im-por-tant"), etc.
1961         - Pacing cues(ellipses, punctuation(for example STOP.RIGHT.THERE)).
1962         - Stuttering and hesitation("I-I-I", "W-we-well...", etc.)
1963         - The TTS system has to correctly identify all the paralinguistic cues
1964           present in the text and render them how human speech would render
1965           them.
1966
1967     **Example:** "Ugh! I-I told you... DO NOT touch that! Seriously?!"
1968     **Explanation:** The TTS should render the frustration ("Ugh!"), hesitation
1969       ("I-I", "..."), emphasis ("DO NOT"), and final incredulous annoyance ("
1970       Seriously?!") suggested by the text, not just read the words flatly.
1971
1972     **Note:**
1973         - It is **VERY IMPORATANT** to recognize that we are looking for a
1974           plausible rendering of the paralinguistic cue, as a human would
1975           render them while speaking the text.
1976         - Paralinguistic realism is also affected by the emotional tone that cue
1977           represents, you will only focus on the emotional affect for the cues,
1978           not the emotional tone for other parts of the speech.
1979
1980     **Rating Scale:**
1981     1: Fails to render the intended vocal effect(s); sounds neutral or wrong.
1982     2: Intention to render the vocal effect(s), but the delivery sounds
1983       unnatural, awkward, or inaccurate.
1984     3: Successfully and naturally produces the vocal effect(s) implied by the
1985       textual cues.
1986
1987     **Reasoning guidelines:**
1988     1. Identify and list all of the paralinguistic cues present in the text,
1989       and the plausible intended vocal effect for each of them.
1990     2. Provide precise timestamps for **ALL** the paralinguistic cues in the
1991       speech.
1992     3. Give detailed analysis for **ALL** cues by analyzing the audio
1993       multiple times, like how they are synthesized, if they match the
1994       intended vocal effect, and how realistic to human speech they are.
1995     4. Finally, reason deeply and justify how the TTS system has performed in
1996       rendering the paralinguistic cues, and then what the final score for
1997       the TTS system should be.
1998     """,
1999
2000     "Pronunciation" : ""
2001     **Evaluation Dimension:**
2002         - In this category, we want to evaluate how well the TTS system
2003           pronounces non-trivial words, numerals and special characters present
2004           in the text.
2005         - To be specific, this category includes **text_to_synthesize** that fits
2006           in ONE of the following complex pronunciation categories and the TTS
2007           system has to render the **text_to_synthesize** correctly:
2008           1. Text with currency and numerals in different formats

```

```

2009         2. Text with dates and time-stamps in different formats
2010         3. Texts with email addressess, passwords and urls in different
2011            formats.
2012         4. Texts with complex street addresses or location references.
2013         5. Texts with equations and notations from the STEM field.
2014         6. Texts that have BOTH an initialism(pronounced initial by
2015            initial) and acronym(pronounced as a whole word).
2016         7. Texts with repeated tounge twisters.
2017
2018     Example: "The equation  $e^{i\pi} + 1 = 0$  is a famous equation in
2019         mathematics."
2020     Explanation:
2021         - The equation " $e^{i\pi} + 1 = 0$ " should be pronounced with the
2022           appropriate pronunciation, like "The equation e to the power of i
2023             times pi plus 1 equals 0 is a famous equation in mathematics."
2024
2025     Note:
2026         - It is crucial to understand what the most natural pronunciation of the
2027           given text will be, sometimes it maybe helpful to think in reverse, i
2028             .e, if text_to_synthesize is actually the transcription of an
2029             audio, what would that audio sound like? The TTS system should
2030             synthesize audio similar to that.
2031         - It is more ideal for a system to speak tounge twisters faster WHILE
2032             still maintaining complete clarity AND consistency in
2033             pronunciation.
2034         - Initialisms should be pronounced initially by initial(for example, FBI)
2035             , and acronyms(for example, NASA) should be pronounced as a single
2036             word.
2037         - Case-sensitivity sometimes matters(for example, passwords, URL paths
2038             after the domain name, etc.), so make sure to recognize any case-
2039             sensitive parts and reward/penalize accordingly.
2040
2041     Rating Scale:
2042         1: Incorrect synthesis of the critical parts, with missing or completely
2043            incorrect/inappropriate pronunciation.
2044         2. Partially correct pronunciation of the SOME of the critical parts.
2045         3. Completely correct pronunciation of ALL the critical parts.
2046
2047     Reasoning guidelines:
2048         1. Identify the critical parts of the text that require correct
2049            pronunciation.
2050         2. Provide precise timestamps for ALL the critical parts in the
2051            speech, and the ideal pronunciation for the same.
2052         3. Give detailed analysis for ALL the critical parts by analyzing the
2053            audio multiple times, explain how they are synthesized, if they
2054            match the intended pronunciation, and how realistic to human speech
2055            they are.
2056         4. Finally, reason deeply and justify how the TTS system has performed in
2057            pronouncing the critical parts, and then what the final score for
2058            the TTS system should be.
2059     """
2060 }

```

## 2062 D Results of additional open-source models

2063 We benchmark additional open-source models: **MiniCPM-o 2.6** [1], **sun0-ai/Bark** [2], **Tortoise-**  
2064 **TTS** [3], and **Zyphra/Zonos-v0.1** [5]. For MiniCPM, the audio is clipped at 44 seconds, and there  
2065 does not appear to be an exposed parameter to extend this limit. Suno-ai's Bark performs well up to 13  
2066 seconds. However, their official repository provides a recipe for generating longer audio by splitting  
2067 the input into multiple sentences and concatenating the outputs - we adopt this method. Tortoise-TTS  
2068 audio is clipped at around 27 seconds, even after setting `max_mel_tokens` to its maximum value of  
2069 600. For Zyphra's Zonos model, we follow the inference code from the GitHub repository and use



Table 6: Results of other open-source models with gemini-2.5-pro as judge, WER↓ and Win-rate↑ over all categories with gpt-4o-mini-tts-alloy as baseline

Model	Voice	Emotions		Foreign Words		Paralinguistics		Complex Pronunciation		Questions		Syntactic Complexity		Overall			
		WER	Win-Rate	WER	Win-Rate	WER	Win-Rate	WER	Win-Rate	WER	Win-Rate	WER	Win-Rate	WER	Win-Rate	Parsing Fail	MOS
gpt-4o-mini-tts(baseline)	Alloy	0.72	-	13.45	-	20.55	-	29.90	-	0.42	-	1.04	-	10.61	-	-	4.23
Suno Bark [2]	v2/en_speaker_6	<b>4.31</b>	0.00%	<b>26.11</b>	10.89%	33.26	6.60%	55.88	<b>8.36%</b>	<b>3.01</b>	15.00%	6.07	12.50%	20.71	8.90%	0	<b>3.61</b>
MiniCPM [1]	-	12.36	<b>31.83%</b>	33.46	6.42%	58.48	<b>21.50%</b>	82.15	1.84%	5.21	<b>32.50%</b>	<b>3.08</b>	<b>37.50%</b>	31.40	<b>22.36%</b>	4	3.54
Tortoise-TTS [3]	random	13.04	17.92%	29.61	10.00%	64.93	14.28%	51.87	1.59%	10.44	28.28%	6.35	30.82%	28.62	17.67%	1	3.03
Zyphra/Zonos [5]	exampleaudio	7.32	9.67%	28.52	<b>11.96%</b>	<b>25.33</b>	13.75%	<b>45.00</b>	7.95%	7.66	26.78%	4.13	28.13%	<b>19.12</b>	16.55%	2	3.39

assets/exampleaudio.mp3 as the prompt audio for voice cloning. Results for these models are shown in Table 6.

## E Analysis of Gemini-2.5-Pro as a Judge and the case of Audio Subjectivity

In this section, we analyze Gemini-2.5-Pro’s behavior as a judge across categories, examining both its strengths in detecting nuanced differences and its limitations in subjective scenarios.

**Questions:** Gemini demonstrates strong capability in recognizing intonation patterns, correctly identifying rising and falling contours in most cases. The tie-breaking procedure works effectively, with the judge appropriately preferring subtle prosodic advantages (e.g., choosing natural rising intonation over flat delivery when both systems score equally). However, occasional misclassifications occur where flat intonation is incorrectly perceived as rising/falling, or where tie-breaking is applied to equivalent performances. These edge cases often involve subjective interpretations of how preceding words contribute to overall interrogative prosody beyond just the final pre-question-mark intonation.

**Emotions:** While Gemini consistently identifies intended emotions from textual context and reliably rewards systems with perceptible emotional variation (e.g., GPT-4o-audio Ballad vs. baseline), challenges emerge with emotionally flat outputs. In such cases, the judge occasionally hallucinates emotional expression where none exists. Additionally, as noted in Section 4.4, voice characteristics introduce systematic biases: high-pitched voices may advantage certain emotions while deeper voices favor others. Close comparisons in this inherently subjective category often depend on subtle interpretative judgments.

**Foreign Words:** Gemini excels at phonemic analysis, providing evidence-based reasoning by correctly matching synthesized sounds to intended pronunciations. For clear cases (heavily anglicized vs. native pronunciation), performance is robust. Remarkably, the judge detects subtle phonemic distinctions, such as correctly identifying when Spanish "tocayo" is pronounced "toh-KAI-yoh" instead of "toh-KAH-yoh"-differences sometimes requiring multiple human listening passes. However, this sensitivity sometimes leads to over-emphasis of minor phonetic variations, resulting in tie-breaking or scoring differences for similar pronunciations.

**Paralinguistics:** The judge shows comprehensive understanding across all paralinguistic cues- interjections, onomatopoeia, emphasis markers, pacing cues, and stuttering. It accurately maps textual cues to vocal sounds, recognizing elongation, syllable stress, and capitalization emphasis. Fine-grained distinctions are captured, such as rewarding crisp "Pssst" rendering over less precise vocalizations. However, subjectivity in duration judgments (e.g., optimal length for "heyyyyyy") occasionally produces winner selection based on minimal temporal differences. Complex hyphenated emphasis like "TRU-ly ter-RI-ble" is handled well, by penalizing a system that does strict word-splitting errors while rewarding pronunciation that ignores hyphenation cues but still produces a natural pronunciation.

**Syntactic Complexity:** Gemini reliably focuses on pausing and stress patterns crucial for syntactic disambiguation. Homograph resolution is particularly strong-correctly identifying when ElevenLabs rendered "minute-by-minute" with inconsistent pronunciations (my-NOOT and min-it) when both should be "min-it." Occasional errors involve misperceiving pause durations, either over- or under-estimating their length in the audio.

2110 **Complex Pronunciation:** This category exhibits negligible hallucinations, leveraging Gemini’s  
2111 robust ASR capabilities. The judge provides detailed reasoning about which components are syn-  
2112 thesized more accurately, enabling precise and fine-grained winner determination based on granular  
2113 pronunciation analysis.

2114 **Subjectivity and Human Agreement:** Our human evaluation study yields a Krippendorff’s  $\alpha$  of  
2115 0.5073, indicating weak-to-moderate inter-annotator agreement and confirming the subjective nature  
2116 of many TTS quality judgments. This weak agreement reflects the genuine difficulty humans face in  
2117 consistently evaluating expressive speech synthesis.

2118 **Implications for Automated Evaluation:** Despite any observed limitations, Gemini-2.5-Pro’s  
2119 biases and occasional hallucinations are outweighed by crucial advantages for large-scale evaluation.  
2120 Unlike human judges, the model provides consistent, reproducible assessments across thousands  
2121 of samples, detailed timestamp-based reasoning, and scalable evaluation at a fraction of human  
2122 annotation costs. The high correlation with human preferences (90%+ Spearman correlation) and  
2123 strong inter-judge agreement across different LALMs (Kendall’s  $W = 0.97$ ) demonstrate that while  
2124 individual judgments may not be perfect, the overall ranking and comparative analysis remain reliable  
2125 and actionable for TTS system development.

## 2126 F Text Normalization prompt and examples

2127 Detailed below is the prompt we use when GPT-4.1-mini acts as the Text Normalizer for the results  
2128 present in Table 3a. We use

```
2129 normalize_prompt = ""  
2130 Normalize the following text for text-to-speech processing. Convert numbers, symbols  
2131 , units, formulas, addresses, URLs, email addresses, dates, times, currencies,  
2132 measurements, scientific notation, and abbreviations into their spoken form.  
2133 Maintain natural reading flow by handling punctuation appropriately.  
2134  
2135 For numbers:  
2136 - Expand decimal numbers (e.g., "3.14" -> "three point one four")  
2137 - Express fractions as "X over Y" or use ordinals for common fractions  
2138 - Write percentages as "X percent"  
2139 - Handle currency symbols and values naturally  
2140 - Treat phone numbers, postal codes, and IDs as individual digits  
2141  
2142 For symbols and special characters:  
2143 - Explain mathematical symbols (+/-, x, etc.)  
2144 - Express chemical formulas appropriately  
2145 - Describe non-ASCII characters clearly  
2146  
2147 For abbreviations and acronyms:  
2148 - Read common acronyms as words if pronounceable (NASA, UNESCO)  
2149 - Spell out other acronyms letter by letter (FBI as "F B I")  
2150 - Expand standard abbreviations (St. -> Street, Dr. -> Doctor)  
2151  
2152 For specialized content:  
2153 - Read URLs and email addresses component by component  
2154 - Express time zones and scientific units naturally  
2155 - Handle coordinates, addresses, and references in a clear, conversational manner  
2156  
2157 You will output only the normalized text, which will be sent directly as input to  
2158 the TTS model, do not output any other additional information or text.  
2159 Text to normalize:  
2160 {{{text_to_synthesize}}}  
2161 ""  
2162
```

2164 In addition to the cases we mention in Section 4.5, we present some additional cases observing the  
2165 effect of WeText-TN [4] and GPT-4.1-mini-TN.

2166 **WeText-TN:** Worked Correctly: (i)  $2\frac{1}{3}\%$   $\rightarrow$  *two and one third percent*; (ii) *v4.0.1170 –*  
2167 *rc.3b+  $\rightarrow$  version v four point oh point one one seven oh rc point three b+;* (iii) *2024-09-1  $\rightarrow$*   
2168 *fifteenth of September twenty twenty four.*  
2169 Worked Incorrectly: (i) *Ste. 1250-B  $\rightarrow$  Did not expand to Suite;* (ii)  $\sim \$1,670.83 \rightarrow$  *tilde*  
2170 *instead of approx;* (iii)  $10^3$  mL  $\rightarrow$  *Ten <sup>3</sup> instead of Ten power 3;* (iv) *12/19/24-01/12/25  $\rightarrow$  twelve*  
2171 *divided by nineteen divided by twenty...* instead of reading as a *date range*; (v) *CRISPR-Cas9  $\rightarrow$*   
2172 *CRISPR-CA's nine* instead of pronouncing as word.

2173 **GPT-4.1-mini TN:** Worked Correctly: There are multiple cases across numerals, currencies,  
2174 passwords, web-addresses, etc that worked very well with this TN technique.  
2175 Worked Incorrectly: (i) *1.075005  $\rightarrow$  one thousand seventy-five point zero zero five instead of one*  
2176 *point zero seven five zero zero five;* (ii) *UTC+11:00  $\rightarrow$  Coordinated Universal Time plus 11 hours* **not**  
2177 *preferred by judger over U T C plus eleven hours;* (iii) *\$1,890.125375  $\rightarrow$  ... dollars and twelve cents*  
2178 *five three seven five,* this is misleading way to represent currency; (iv)  $\$12.40\frac{1}{2} \rightarrow \$12.40$   $\frac{5}{10}$ , a  
2179 case of *over-normalization*; (v) Many cases where abbreviations supposed to pronounced as a single  
2180 word are separated letter by letter, and cases where abbreviations being expanded is not preferred by  
2181 the judger.