Putting HUMANS first: Efficient LAM Evaluation with Human Preference Alignment

Woody Haosheng Gan[1*], **William Held**[2,3], **Diyi Yang**[2]

[1]University of Southern California, [2]Stanford University, [3]OpenAthena

woodygan@usc.edu, held@stanford.edu, diyiy@stanford.edu

Abstract

The rapid proliferation of large audio models (LAMs) demands efficient approaches for model comparison, yet comprehensive benchmarks are costly. To fill this gap, we investigate whether minimal subsets can reliably evaluate LAMs while reducing costs and data redundancy. Analyzing 10 subset selection methods with 18 audio models across 40 tasks covering major LAM evaluation dimensions, we show that subsets of just 50 examples (0.3% of data) can achieve over 0.93 Pearson correlation with full benchmark scores. To understand how well these scores align with what practitioners ultimately care about–user satisfaction–we collect 776 human preference ratings from realistic voice assistant conversations, finding that both subsets and full benchmark achieve only 0.85 correlation with human. To better predict preferences, we trained regression models on these selected subsets, achieving 0.98 correlation—outperforming regression models trained on both random subsets and the full benchmark. This demonstrates that in regression modeling, well-curated subsets outpredict the full benchmark, showing quality over quantity. We open-source these regression-weighted subsets as the HUMANS benchmark, an efficient proxy for LAM evaluation that captures both benchmark performance and user preferences.

1 Introduction

The landscape of large audio models (LAMs) has expanded rapidly, with families like Gemini (Gemini Team et al., 2023), GPT-audio (OpenAI, 2024a), Qwen-Omni (Xu et al., 2025a), and Ultravox (Fixie AI, 2024) demonstrating diverse capabilities. This proliferation creates a practical challenge: how to quickly compare and select models without exhaustive evaluation. Existing LAM benchmarks containing thousands of examples create substantial computational barriers—audio evaluation requires 10–100× more tokens than text, making single-model evaluation cost hundreds of GPU-hours and dollars. This makes it impractical to quickly compare candidate models, evaluate checkpoints, or A/B test configurations. More critically, static benchmarks may poorly align with human preferences (Li et al., 2025; Schaeffer et al., 2025), failing to reflect what practitioners care about when conversational quality and user experience are paramount. This raises two critical questions: *Can we reliably rank LAMs using small benchmark subsets? How to capture what users actually care about?*

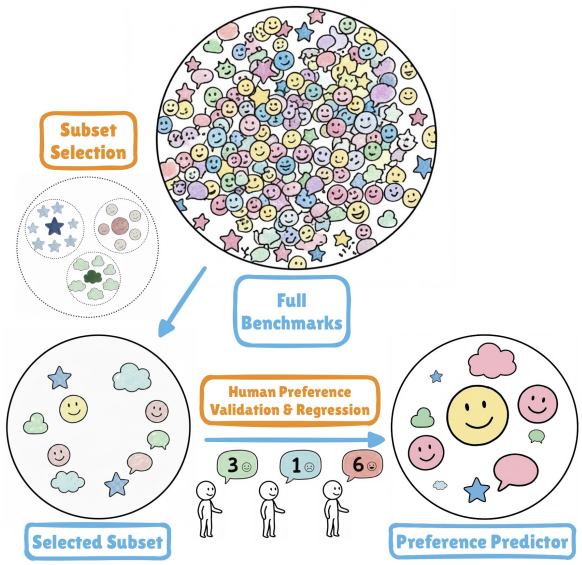

Figure 1: **Overview.** We select minimal subsets from full benchmark pools, validate alignment with human preferences through interactive evaluations, and train regression models to efficiently predict user satisfaction.

To answer these questions, we first conduct a comprehensive analysis of benchmark subset selection for LAMs. Evaluating 18 audio models across 40 tasks covering major LAM conversation scenarios (∼16,000 datapoints from 5 benchmarks), we systematically compare 10 subset selection methods, showing that carefully selected subsets of just

[*]Work done while visiting Stanford University.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 39132–39174

50 examples (0.3% of data) achieve 0.934 Pearson correlation with full benchmark scores.

To understand whether benchmark scores reflect user satisfaction in real-world deployment, we collect 776 human preference ratings from 10-minute interactive conversations with 7 representative models across realistic scenarios spanning tool calling, task-oriented dialogue, and open chat. Our analysis reveals that both selected subsets and the full benchmark plateau at 0.85 correlation, indicating a substantial gap between static evaluation metrics and real-world user experience. Beyond quantitative ratings, we analyze qualitative feedback, revealing failure modes such as excessive verbosity and robotic speech not emphasized by benchmarks.

To further improve human preference prediction, we train regression models on selected subsets, creating HUMANS (HUman-aligned Minimal Audio evaluatioN Subsets) benchmark, which achieves 0.978 correlation with user satisfaction, outperforming regression models on random sampling and full benchmark. Our contributions are:

1. Systematic evaluation of 10 subset selection methods for audio benchmarks, demonstrating that small subsets enable reliable model ranking while dramatically reducing costs
2. A human preference dataset of 776 ratings from realistic voice assistant interactions, understanding benchmark-human alignment, providing qualitative analysis on user feedback, and enabling meta-evaluation of future benchmarks
3. Demonstration that regression models trained on benchmark subsets predict human preferences well, providing an efficient proxy that captures benchmark performance and user preference

2 Related Work

2.1 Large Audio Models

The landscape of large audio models (LAMs) has rapidly evolved from specialized architectures like Whisper (Radford et al., 2023) for speech recognition and VALL-E (Wang et al., 2023) for synthesis into versatile, general-purpose models. Contemporary LAMs include audio-in text-out models that process speech for text generation (e.g., Gemini (Gemini Team et al., 2023), Ultravox (Fixie AI, 2024), Voxtral (Liu et al., 2025), Gemma 3n (Gemma Team, 2025), Phi-4-multimodal (Abouelenin et al., 2025)) and end-to-end omni-modal systems that natively handle audio input and output (e.g., GPT-realtime (OpenAI, 2025d), Qwen-Omni (Xu et al., 2025a), GLM-4-Voice (Zeng et al., 2024), and MiniCPM-o (Yao et al., 2024)). This diversity makes systematically comparing models complex.

2.2 LAM Evaluation Benchmarks

The LAM evaluation landscape includes specialized benchmarks targeting specific aspects (SpeakBench (Manakul et al., 2025) for paralinguistics, MMAU (Sakshi et al., 2024) for reasoning, ADU-Bench (Gao et al., 2024) for dialogue), application-oriented benchmarks focusing on voice assistant scenarios (WildSpeech-Bench (Zhang et al., 2025b), VoiceBench (Chen et al., 2024)), general audio understanding benchmarks (AudioBench (Wang et al., 2024), AIR-Bench (Yang et al., 2024)), and comprehensive frameworks (Dynamic-SUPERB (Huang et al., 2024b), UltraEval-Audio (He et al., 2024), CAVA (Held et al., 2025)). This fragmentation and extensive scope creates substantial computational burden, motivating efficient subset selection.

2.3 Benchmark Subset Selection Methods

Sample-efficient benchmarking has roots in psychometrics, particularly Item Response Theory (IRT) for selecting discriminative items (Lalor et al., 2016; Martínez-Plumed et al., 2019), with foundational extensions to diversity-based clustering (Misir, 2021), training dynamics (Swayamdipta et al., 2020), and gradient-based active learning (Coleman et al., 2020). Modern adaptations on LLMs include Anchor Points (Vivek et al., 2023), Efficient Benchmarking (Perlitz et al., 2024), TinyBenchmarks (Polo et al., 2024), and SUBLIME (Saranathan et al., 2025) that achieve high correlation with full rankings using minimal subsets. However, these techniques remain largely unexplored for LAM evaluation.

2.4 Human Preference and Meta-Analysis

Traditional human evaluation assessed perceptual quality using metrics like MOS (ITU-T, 1996). In the LLM era, Chatbot Arena (Chiang et al., 2024) introduced large-scale pairwise preference collection, with the LMSYS dataset (Zheng et al., 2023) becoming a gold standard for meta-evaluating benchmarks. Human preferences are leveraged to predict satisfaction on unseen models (Schaeffer et al., 2025; Ryan et al., 2025). Recent work extended this to audio models: TalkArena (Li et al., 2025) collected preferences on audio-in text-out

systems in single-turn interactions, revealing significant misalignment between benchmark scores and human preferences. In our human evaluation, we further captures more realistic deployment scenarios of LAMs: real-time voice assistants handling multi-turn conversations and tool interactions.

3 Subset Selection

In this section, we systematically evaluate methods for selecting minimal yet informative benchmark subsets that preserve model rankings, across 40 tasks covering major LAM conversation evaluation dimensions (from 5 benchmarks, ∼16,000 datapoints). Through cross-validation on 18 diverse audio models, we identify the most effective subset selection methods and construct final minimal benchmark subsets for practical use.

3.1 Experimental Setup

3.1.1 Audio Models

We evaluate 18 publicly available audio models with diverse characteristics to ensure our findings generalize across the LAM landscape. Our selection spans multiple architectural paradigms: end-to-end omni-modal systems that natively process and generate speech (e.g., GPT-4o-audio, Qwen2.5-Omni), speech-to-text models that encode audio for text-based reasoning (e.g., Gemini 2.5, Ultravox, Voxtral), and pipeline systems combining separate components (e.g., Llama-3.2 with external STT/TTS). Models range from 1B parameters to large proprietary systems. For consistency in evaluation, all models without native audio output use GPT-4o-mini-tts (OpenAI, 2025a) for speech synthesis, while text-only models in pipeline configurations use GPT-4o-transcribe (OpenAI, 2025b) for audio input transcription. Complete model specifications, architectural details, and processing configurations are provided in Appendix A.

3.1.2 Benchmarks

We construct our evaluation suite from 5 established audio benchmarks: Dynamic-SUPERB Phase 2 (Huang et al., 2024a), CAVA (Held et al., 2025), UltraEval-Audio (He et al., 2024), Speak-Bench (Manakul et al., 2025), and WildSpeech-Bench (Zhang et al., 2025b), selecting tasks focused on LAM capabilities on human conversation and speech. These benchmarks are selected to be complementary, collectively providing tasks that represent the majority of evaluation scenarios in recent LAM literature. This yields 40 distinct evaluation tasks evaluating different dimensions of audio model capabilities, including speech recognition, dialogue understanding, instruction following, multi-turn function calling, and more. To ensure comparability across diverse metrics, we unify scales of all metrics to [0,1] where 1 represents best performance. Complete task descriptions and metric unification procedures are detailed in Appendix B.

Our full evaluation across 18 models and ∼16,000 examples required approximately 1520 GPU-hours on NVIDIA A6000 GPUs for open-source models and $2,400 in API costs for proprietary models, motivating our investigation of selecting minimal and informative subsets.

3.1.3 Full Benchmark Reference Scores

To establish reference scores, we compute task-averaged scores where each task contributes equally regardless of item count:

$$\text{Score}(m) = \frac{1}{T}\sum_{t=1}^{T} \bar{s}_{m,t} \tag{1}$$

where T is the number of tasks and $\bar{s}_{m,t}$ is model m's average score across all items in task t. Since our 40 tasks are selected to cover major LAM conversation evaluation dimensions, equal task weighting ensures each dimension contributes proportionally, preventing tasks with more examples from dominating rankings. These reference scores serve as our gold standard for evaluating whether selected subsets preserve model rankings.

3.2 Subset Selection Methods

3.2.1 Random Sampling Methods

Task-Balanced Random Sampling. As our baseline, we employ task-balanced random sampling where each datapoint in task t (containing x_t items) has sampling probability $p_i = 1/(T \cdot x_t)$, where T is the number of tasks, ensuring each task contributes equally in expectation: $\mathbb{E}[\text{samples from task } t] = n/T$.

Random-Sampling-Learn. Building on the baseline subset, it uses Ridge regression to predict full benchmark scores from subset scores (Zhang et al., 2025a). we learn g by minimizing regularized loss over source models $\mathcal{M}$, then predict target scores as $h(f) = g[s(f, C)]$ (see Appendix C.1).

Random-Search-Learn. This extends Random-Sampling-Learn by performing $N = 1000$ random sampling iterations, training Ridge regression on

each using 75% of source models for training and 25% for validation, selecting the subset with lowest validation error, then retraining on all models.

3.2.2 Intrinsic Item Property Methods

Variance-Based Selection. We select items with highest discriminative power. For each item i, we compute variance $\sigma_i^2 = \frac{1}{K-1}\sum_{k=1}^{K}(s_{i,k} - \bar{s}_i)^2$ across model performances where $s_{i,k}$ is model k's score on item i, and select the top n highest-variance items globally.

Difficulty-Based Selection. We employ stratified sampling to span the full difficulty spectrum (Saranathan et al., 2025). We define difficulty as $D_i = 1 - \frac{1}{K}\sum_{k=1}^{K} s_{i,k}$, partition items into $B = 10$ bins, and allocate equal samples per bin using task-balanced probabilities.

3.2.3 Embedding-Based Clustering Methods

IRT-Based Performance Prediction. Inspired by tinyBenchmarks (Polo et al., 2024), we train a 5-dimensional two-parameter IRT model on source model responses to estimate latent item parameters (discrimination $\alpha_i \in \mathbb{R}^5$ and difficulty $\beta_i \in \mathbb{R}$). We construct 6-dimensional item embeddings $E_i = [\alpha_i; \beta_i]$ that encode each item's latent characteristics, which are more robust to distribution shift than raw correctness patterns. We use these embeddings for task-aware weighted K-Means clustering to select n anchor points. For target model prediction, we estimate ability parameters $\hat{\theta}_m$ from its anchor responses, then compute task-averaged scores using actual responses for observed items and IRT-predicted probabilities $\hat{p}_{im} = \sigma(\hat{\alpha}_i^\top \hat{\theta}_m - \hat{\beta}_i)$ for unseen items. Details are in Appendix C.5.

Anchor-Based Selection. We adapt the anchor points framework (Vivek et al., 2023) with task-aware weighting. We apply weighted K-Means clustering on item embeddings using Euclidean distance, with each item weighted by $1/(T \cdot |T_t|)$ where T is the number of tasks and $|T_t|$ is the task size, resulting in n clusters. We select the datapoint nearest to each centroid as an anchor point.

For target model m, the Anchor Point Weighted (APW) score is:

$$\text{APW}(m) = \sum_{i=1}^{n} w_i \cdot s_{m,a_i} \tag{2}$$

where a_i is anchor point i, s_{m,a_i} is model m's normalized score on a_i, and $w_i = \sum_{j \in C_i} b_j$ is the cluster weight (sum of task-normalized weights b_j for items in cluster C_i), ensuring equal task contribution. Implementation details are in Appendix C.6.

Embedding Choices for Clustering. We explore four embedding spaces for the anchor-based clustering step, each producing a method variant:

Anchor Points (APW): Clusters directly on source model score vectors (the original method).

Semantic Embedding: Prompts encoded using OpenAI's text-embedding-3-large (OpenAI, 2024d), PCA-reduced from 3072 to 50 dimensions.

Acoustic Embedding: 1024-dim acoustic embeddings extracted using WavLM-Large (Chen et al., 2022), reduced to 50 dimensions via PCA.

Combined Embedding: Combined representations concatenating: (1) acoustic embeddings using WavLM-Large (Chen et al., 2022), (2) semantic embeddings using OpenAI's text-embedding-3-large (OpenAI, 2024d), (3) source model performance scores, and (4) binary metadata indicating whether audio input/output is required. Acoustic and semantic embeddings are PCA-reduced to match source model count (e.g., 18 dims if 18 models in (3)) and MinMax-scaled to [0,1] to match the range of performance scores and metadata. Equal dimensionality for the first three components ensures balanced contribution to clustering.

3.3 Subset Selection Performance

We evaluate each subset selection method across varying subset sizes from $n = 10$ to $n = 1000$ using 3-fold cross-validation with 100 random repeats (300 total evaluations per size). In each fold, we use 12 models for subset selection and evaluate alignment with full benchmark scores on the remaining 6 held-out models via Pearson correlation. Table 1 reports correlation at key subset sizes ($n \in \{10, 20, 30, 50, 100, 200\}$) and Area Under the Correlation Curve (AUCC) over $n \in [10, 200]$, representing the average correlation achieved across this range. We also report N_{90} and N_{95}—the minimum subset sizes achieving Pearson correlation $r \geq 0.90$ and $r \geq 0.95$ with full benchmark scores, respectively. Figure 2 shows the correlation curves for the two top-performing methods and random sampling baseline across subset sizes. Complete correlation curves for all methods and results for alternative correlation metrics are provided in Appendix D.

Key findings from our evaluation:

- **Combined Embedding achieves best overall performance**: Highest AUCC (0.943) and

Method	Pearson Correlation by Subset Size						AUCC	N_{90} / N_{95}
	$n = 10$	$n = 20$	$n = 30$	$n = 50$	$n = 100$	$n = 200$	$[10, 200]$	
Random Sampling	$0.559_{\pm 0.021}$	$0.718_{\pm 0.014}$	$0.784_{\pm 0.011}$	$0.856_{\pm 0.008}$	$0.916_{\pm 0.005}$	$0.959_{\pm 0.003}$	0.891	83 / 164
Random-Sampling-Learn	$0.544_{\pm 0.022}$	$0.656_{\pm 0.017}$	$0.719_{\pm 0.014}$	$0.791_{\pm 0.012}$	$0.887_{\pm 0.006}$	$0.940_{\pm 0.003}$	0.854	119 / 300
Random-Search-Learn	$0.619_{\pm 0.018}$	$0.676_{\pm 0.017}$	$0.734_{\pm 0.015}$	$0.803_{\pm 0.010}$	$0.894_{\pm 0.005}$	$0.937_{\pm 0.004}$	0.866	99 / 300
Variance-based	$0.525_{\pm 0.020}$	$0.628_{\pm 0.016}$	$0.676_{\pm 0.014}$	$0.716_{\pm 0.013}$	$0.756_{\pm 0.012}$	$0.804_{\pm 0.011}$	0.742	– / –
Difficulty-based	$0.608_{\pm 0.020}$	$0.761_{\pm 0.012}$	$0.811_{\pm 0.009}$	$0.863_{\pm 0.007}$	$0.924_{\pm 0.004}$	$0.964_{\pm 0.002}$	0.902	71 / 157
IRT-based	$0.486_{\pm 0.020}$	$0.698_{\pm 0.014}$	$0.778_{\pm 0.013}$	$0.864_{\pm 0.007}$	$0.919_{\pm 0.004}$	$\underline{0.960}_{\pm 0.002}$	0.892	81 / 156
Anchor Points	$\mathbf{0.797}_{\pm \mathbf{0.011}}$	$\mathbf{0.856}_{\pm \mathbf{0.007}}$	$\mathbf{0.884}_{\pm \mathbf{0.006}}$	$\underline{0.907}_{\pm 0.005}$	$\underline{0.940}_{\pm 0.004}$	$0.952_{\pm 0.003}$	$\underline{0.927}$	$\underline{40}$ / 155
Semantic Embedding	$0.466_{\pm 0.023}$	$0.627_{\pm 0.016}$	$0.781_{\pm 0.012}$	$0.877_{\pm 0.007}$	$0.921_{\pm 0.005}$	$0.936_{\pm 0.003}$	0.856	60 / 350
Acoustic Embedding†	$\underline{0.736}_{\pm 0.013}$	$0.445_{\pm 0.019}$	$0.672_{\pm 0.016}$	$0.870_{\pm 0.008}$	$0.904_{\pm 0.006}$	$0.943_{\pm 0.003}$	0.850	92 / 250
Combined Embedding†	$0.651_{\pm 0.019}$	$\underline{0.831}_{\pm 0.010}$	$\underline{0.878}_{\pm 0.007}$	$\mathbf{0.934}_{\pm \mathbf{0.004}}$	$\mathbf{0.963}_{\pm \mathbf{0.002}}$	$\mathbf{0.977}_{\pm \mathbf{0.001}}$	**0.943**	**32 / 67**

Table 1: **Subset selection performance across methods and sizes.** Pearson correlation between subset and full benchmark scores (mean ± SEM over 300 evaluations). AUCC computed over $n \in [10, 200]$. N_{90}/N_{95} show minimum sizes achieving $r \geq 0.90/0.95$. "–" indicates threshold not achieved within $n = 1000$. **Bold** indicates best, underline second-best. †Audio-specific methods unique to this work (leverage acoustic features).

superior correlation for $n \geq 50$, reaching $r = 0.977$ at $n = 200$.

- **Anchor Points excel at small subset sizes**: Best performance for $n \leq 30$ (e.g., $r = 0.797$ at $n = 10$), demonstrating superior sample efficiency in minimal-evaluation scenarios.
- **Random Sampling provides surprisingly strong baseline**: With 0.891 AUCC and $r = 0.959$ at $n = 200$, random sampling outperforms variance-based, learning-based, and single-modality embedding approaches.
- **Learning-based methods substantially underperform**: Random-Sampling-Learn and IRT-based approaches achieve relatively lower AUCC (0.854, 0.892), potentially due to overfitting when generalizing learned patterns from limited source models to unseen models even with regularization.
- **Small subsets strongly correlate with full benchmarks**: Combined Embedding reaches $r = 0.934$ with only 50 samples (∼0.3% of full benchmark), enabling reliable model ranking with minimal evaluation.

Based on these results, we select Anchor Points for $n \leq 30$ and Combined Embedding for $n \geq 50$ as our best-performing methods. For each subset size, we apply the corresponding method using all 18 models as source models to construct the final benchmark subsets. These selected subsets provide practitioners with reliable, minimal evaluation sets that align with full benchmark scores while dramatically reducing costs. We use them as our "best" subsets for the analysis in Sections 4.3 and 4.4. Further analysis of task composition in these subsets reveals that these clustering-based methods naturally prioritize foundational capabilities like ASR and speaker diarization in smaller subsets, progressively incorporating more refined tasks as subset size increases (detailed in Appendix E).

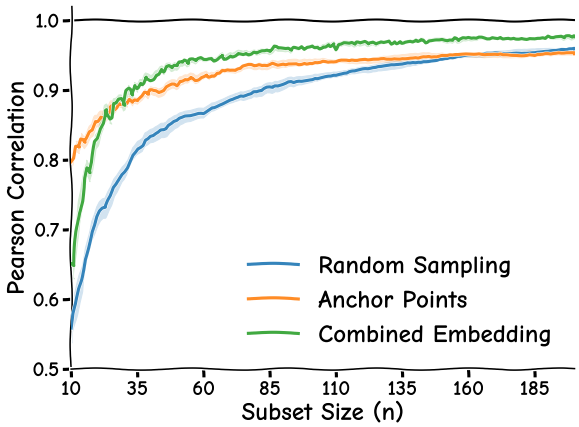

Figure 2: **Subset selection performance.** Pearson correlation with full benchmark scores. Combined Embedding achieves best overall performance (AUCC=0.943), while Anchor Points excel at small sizes ($n \leq 30$). For clarity, only the top-performing and baseline methods are shown here; complete correlation curves for all 10 methods are provided in Appendix D.

4 Human Evaluation

Beyond benchmark performance, practitioners ultimately care about real-world user experience. To obtain this gold standard for model performance, we collect human preference ratings in realistic voice assistant scenarios. This enables us to examine whether selected subsets and full benchmark scores align with user satisfaction.

4.1 Experimental Design

4.1.1 Conversational Agent Framework

Recognizing that large audio models are predominantly deployed as real-time conversational agents, we develop a model-agnostic voice agent

framework adapted from LiveKit (LiveKit, Inc., 2024) to enable real-time conversational evaluation. The framework supports end-to-end audio models, audio-in text-out models, and text-only models. For models without native audio input or output capabilities, we use the same STT and TTS components as described in Section 3.1.1 to ensure consistent audio processing across all evaluations. To further ensure fair comparison, we implement consistent system prompts, conversation management, and interaction protocols across all models. Technical details are provided in Appendix F.

4.1.2 Model Selection

We evaluate 7 representative models from our pool of 18, spanning diverse architectures and sizes: GPT-4o-audio-preview, GPT-4o-mini-audio-preview, Gemini-2.5-Flash, Qwen3-Omni-30B-A3B-Instruct-thinker, Ultravox-v0.4-ToolACE-8B, Voxtral-Small-24B-2507, and GPT-4o-mini.

4.1.3 Participant Recruitment

We recruited native English speakers from the United States via Prolific. This research was approved by the Institutional Review Board (IRB) at the authors' institution. Full human evaluation details are in Appendix G.

4.1.4 Conversation Protocol

Each participant engaged in a single 10-minute conversation[*] with a randomly assigned model and scenario. To capture realistic deployment conditions and ensure human evaluations broadly cover audio models' capabilities, we designed three scenario categories based on common voice assistant use cases (Bentley et al., 2018), emphasizing structured, evaluable tasks while maintaining representation of free-form conversation. Scenario generation details in Appendix G.8:

- **Open Chat (20%)**: Free-form conversations without specific goals or instructions, allowing natural interaction patterns to emerge.
- **Goal-Oriented Dialogue (40%)**: Structured conversations with defined objectives, using real interaction patterns sampled from LMSYS (Zheng et al., 2023) and WildChat (Zhao et al., 2024) datasets.
- **Tool Calling Tasks (40%)**: Objective-driven interactions requiring specific actions (shopping, messaging, calendar management, flight booking) where task completion can be measured and displayed to the participant.

[*]We chose 10 minutes based on pilot testing - it's long enough to capture multi-turn dialogue patterns and task completion (15-25 turns typical) while avoiding participant fatigue.

4.1.5 Rating Collection

Following each conversation, participants provided ratings on a 6-point Likert scale across five dimensions as well as open-ended feedback justifying their ratings:

- **Overall Satisfaction**: Holistic assessment of the interaction experience
- **Speech Understanding**: How well the assistant understood speech, intent, and paralinguistic cues
- **Naturalness**: How natural, conversational, and appropriately concise the interaction felt
- **Response Quality**: Accuracy, safety, relevance, and helpfulness of responses
- **Task Effectiveness**: Success and efficiency in helping achieve goals

4.2 Human Evaluation Results

We collected 776 total evaluations across the 7 models (approximately 111 conversations per model on average). Table 2 presents the average human ratings on each dimension.

- **Dimension-Specific Insights**: Understanding consistently exceeds overall satisfaction across models, indicating speech comprehension is not a limiting factor. In contrast, Naturalness scores fall below overall satisfaction, revealing conversational flow as the primary bottleneck. Dimension correlation analysis in Appendix H.2 reveals Response Quality (r=0.773) and Task Effectiveness (r=0.781) drive satisfaction most strongly, while Naturalness shows the weakest correlation (r=0.626)—suggesting users prioritize functional capabilities in their evaluation.

- **Qualitative Failure Mode Analysis**: Open-ended feedback reveals conversational quality issues dominate complaints: robotic speech style (42.8%), stilted flow (18.8%), and excessive verbosity (17.2%) appear in 56.7% of dissatisfied cases. Poor speech recognition accounts for only 8.7%, confirming ASR is largely solved. This reveals a potential mismatch: static benchmarks focus on correctness while users prioritize conversational experience.

- **Model-Specific Patterns**: Pipeline systems show elevated robotic complaints (GPT-4o-mini+STT+TTS: 50.6%), while open-source

Model	Human Evaluation						Benchmark
	Overall	Understanding	Naturalness	Quality	Effectiveness	N	Score
GPT-4o-audio-preview	**4.982** ± 0.091	**5.368** ± 0.080	**4.368** ± 0.117	**5.123** ± 0.086	**4.947** ± 0.105	114	0.575
Gemini-2.5-Flash+TTS	4.664 ± 0.111	5.191 ± 0.100	4.218 ± 0.123	4.936 ± 0.090	4.673 ± 0.113	110	**0.589**
GPT-4o-mini+STT+TTS	4.509 ± 0.122	5.158 ± 0.093	4.132 ± 0.109	4.772 ± 0.100	4.754 ± 0.110	114	0.498
Qwen3-Omni-30B+TTS	4.211 ± 0.140	4.872 ± 0.122	4.000 ± 0.129	4.578 ± 0.122	4.385 ± 0.143	109	0.575
Voxtral-Small-24B+TTS	3.982 ± 0.135	4.618 ± 0.128	3.845 ± 0.135	4.264 ± 0.128	4.036 ± 0.140	110	0.507
GPT-4o-mini-audio-preview	3.685 ± 0.147	4.741 ± 0.116	3.546 ± 0.129	4.296 ± 0.134	4.176 ± 0.143	108	0.466
Ultravox-v0.4-ToolACE-8B+TTS	3.342 ± 0.156	4.036 ± 0.159	3.063 ± 0.139	3.721 ± 0.155	3.550 ± 0.161	111	0.384

Table 2: **Human preference evaluation results.** Mean ratings ± standard error of the mean across five dimensions on a 6-point Likert scale (higher is better). Benchmark shows task-averaged scores from the full benchmark suite. N indicates the number of 10-minute conversations collected per model. Models with +TTS use GPT-4o-mini-tts for speech synthesis, while +STT+TTS indicates a full pipeline system. Models are ordered by Overall Satisfaction.

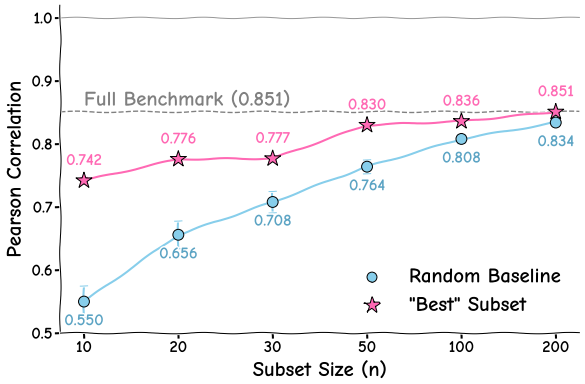

Figure 3: **Benchmark alignment with human preferences.** Pearson correlation between subset scores (averaged over 100 random initializations) and human overall ratings. "Best" Subset: Anchor Points for $n \leq 30$ and Combined Embedding for $n \geq 50$.

models struggle with verbosity (Qwen3-Omni: 27.1%, Voxtral: 23.3% vs. 17.2% average). GPT-4o-audio achieves highest satisfaction (4.98) despite highest latency complaints (38.6%), indicating users tolerate delays for quality.

More comprehensive qualitative analysis of human evaluations is provided in Appendix H. These human preference rankings provide our gold standard for validating benchmark subset selection in the following sections.

4.3 Alignment with Human Preferences

To examine alignment between benchmark performance and human preferences, we evaluate the full benchmark, task-balanced random sampling baseline, and the "best" subsets of sizes $n \in \{10, 20, 30, 50, 100, 200\}$ constructed in Section 3 (Anchor Points for $n \leq 30$, Combined Embedding for $n \geq 50$) on the 7 models with human preference data, computing Pearson correlations between each method's model scores and human overall satisfaction ratings.

Figure 3 presents the results. The full benchmark achieves moderate correlation with human preferences ($r = 0.851$). Our "best" subsets approach this ceiling efficiently—the 200-item subset (∼1.3% of data) matches full correlation—and consistently outperform random sampling, confirming that principled selection which excelled at benchmark score prediction also preserves human preference alignment. Yet this 0.85 ceiling possibly reflects the mismatch from our qualitative analysis of user feedback in Section 4.2: benchmarks and users may prioritize different quality dimensions.

4.4 Predicting Human Preferences

Given the gap between static benchmark scores and human preferences, can we improve human preference prediction with benchmark items so that practitioners could estimate models' likely human reception without costly user studies.

Motivation Our hypothesis is that human preferences emerge as a composite function of model performance across diverse benchmark dimensions (Schaeffer et al., 2025). If the full benchmark contains these key dimensions, and our selected subsets also capture them—as evidenced by their correlations with human preferences (Section 4.3)—then we can learn to weight benchmark items to better predict overall human satisfaction.

4.4.1 Prediction Framework

We employ Ridge regression on both the full benchmark and selected subsets from Section 3 to learn the relationship between benchmark performance and human preferences. For each model m, let $\mathbf{x}_m \in \mathbb{R}^n$ denote its item-level score vector on the n items in the benchmark or subset, where each score $s_{m,i} \in [0, 1]$ is normalized. Let $y_m \in [0, 1]$

denote the model's human overall satisfaction rating (linearly rescaled from the original 6-point Likert scale). We learn a linear predictor:

$$\hat{y}_m = \mathbf{w}^\top \mathbf{x}_m + b \tag{3}$$

where weights $\mathbf{w} \in \mathbb{R}^n$ and bias $b \in \mathbb{R}$ are learned via Ridge regression with L_2 regularization.

4.4.2 Evaluation Protocol

We evaluate the effectiveness of regressions on different subsets with leave-one-model-out (LOMO):

1. For each held-out model m_{test}:
   - Train Ridge regression on the remaining 6 models' subset scores and human ratings: $\{(\mathbf{x}_{m_i}, y_{m_i})\}_{i \neq \text{test}}$
   - Select regularization strength $\alpha \in \{10^{-4}, 10^{-3}, \ldots, 10^{4}\}$ via nested leave-one-out CV with 5 training models, and retrain on all 6 models with the selected α
   - Compute **Pearson correlation** between predicted scores $\{\hat{y}_{m_i}\}_{i=1}^{7}$ and actual human ratings $\{y_{m_i}\}_{i=1}^{7}$ across all models
2. Average Pearson correlations across all 7 LOMO folds to obtain the final metric

Given our limited pool of 7 models, this protocol maximizes regression training informativeness while ensuring fair comparison: all regression models use identical LOMO evaluation, making relative comparisons fair despite inflated absolute values. While not directly comparable to original benchmark correlations (which involve no training data), this metric reliably ranks regression models' effectively. We further validate in Appendix I that regression on best subsets outperforms original unweighted scores on held-out models for $n > 10$.

4.4.3 Results

Figure 4 presents the results:

- **"Best" subsets perform better**: Principled selection methods consistently outperform random sampling at different subset sizes, demonstrating that they capture more discriminative and informative benchmark items, while random sampling includes redundant or low-signal items.
- **Quality over quantity**: Performance peaks at $n = 100$ ($r = 0.978$) for "best" subset before dropping to $r = 0.965$ at $n = 200$ and $r = 0.949$ for the full benchmark. This non-monotonic trend suggests additional items can introduce lower-informative examples that perturb regression weights learning and degrades generalization. Effective human preference prediction requires high-quality, diverse item selection rather than maximizing evaluation coverage—a well-curated subset of 100 items outpredicts the full 15,964-item benchmark.

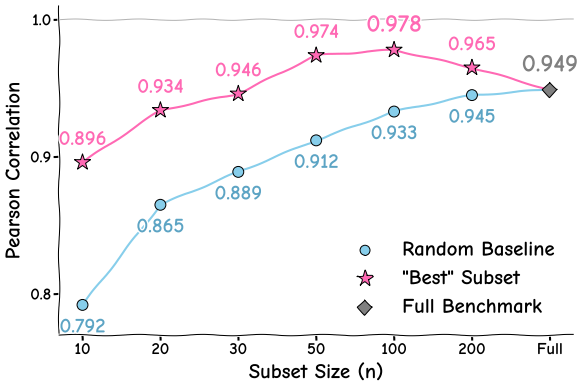

Figure 4: **Human preference prediction via Ridge regression.** Pearson correlation between predictions and actual satisfaction using LOMO CV. See Appendix I for fair comparison excluding in-sample predictions.

5 Open Benchmarks for Practitioners

To provide practitioners with efficient and ready-to-use evaluation tools, we present HUMANS (HUman-aligned Minimal Audio evaluatioN Subsets). For each subset size, we select the best-performing subset based on highest average Pearson correlation across 100-seed cross-validation, and train final Ridge regression models on all 7 human-evaluated models to predict human preferences. Each subset provides two evaluation modes: (1) **regression scores** using learned Ridge weights to predict human preference (overall satisfaction, as well as finer-grained dimensions including understanding, naturalness, response quality, and task effectiveness, each with its own regression model), and (2) **benchmark scores** using the subset selection method's original weights to efficiently approximate full benchmark.

HUMANS benchmark subsets of different sizes with selected items and weights are available at https://huggingface.co/datasets/woodygan/humans-benchmark.

Recalibration Guide. HUMANS supports incremental updates as the LAM landscape evolves. For new benchmarks or tasks, subsets can be reselected and regression weights retrained without additional human preference collection—existing ratings re-

main valid. For substantially new model architectures, the regression can be retrained with incremental human evaluations on the new models. In both cases, the framework supports targeted, low-cost updates rather than full re-evaluation.

6 Conclusion

This work addresses the computational challenge of evaluating large audio models through systematic subset selection. Our analysis demonstrates that principled selection methods identify minimal subsets that preserve both benchmark rankings and alignment with human preferences. Qualitative analysis of user feedback reveals critical gaps between what benchmarks measure and what users value, with conversational quality issues dominating complaints. Regression models trained on selected subsets outperform full benchmarks in predicting user satisfaction, showing that quality trumps quantity in evaluation. We release our subsets and human preference ratings to support efficient model comparison and meta-evaluation.

Limitations

Our work has several limitations: (1) Our human evaluation focuses on native English speakers from the United States, which may not represent the full spectrum of global users, particularly non-native speakers or speakers of other languages. The released HUMANS benchmark is therefore designed for practitioners evaluating LAMs deployed for English-speaking users. While the core framework—that curated subsets efficiently predict both benchmark scores and human preferences—should generalize across languages, the specific task design, subset composition, and regression weights would require recalibration for other languages. We recommend future work extending HUMANS to multilingual evaluation by incorporating language-specific tasks into the task pool. (2) Due to budget constraints and the need for statistical power, we evaluated only 7 models with human preferences, limiting the informativeness of correlation analysis and the robustness of our regression models when generalizing to new architectures. This small sample necessitated a LOMO evaluation protocol that may overestimate absolute generalization performance, though relative comparisons between methods remain valid. A larger pool of human-evaluated models would enable more rigorous held-out evaluation and stronger generalization claims. (3) Our benchmark subsets are optimized for conversational scenarios and may not generalize to other audio domains such as music understanding or generation. (4) Our subset selection methods are trained on current LAMs and may face extrapolation challenges when evaluating substantially more capable future models with different capability profiles. (5) While our methods aim at predicting model-level rankings for rapid comparison, they do not predict item-level scores for individual benchmark examples—though methods like IRT and anchor-based selection do support item-level analysis using selected items, as discussed in their original works. Future work could extend our approach to multilingual evaluation, more comprehensive human evaluations with larger model pools, adaptive subset selection for scenarios beyond conversational use cases such as creative audio generation, specialized domain applications, or emerging model capabilities, and item-level diagnostic evaluation for audio models. Additionally, human-aligned subsets and preference scores could also inform training data selection and model fine-tuning.

Ethical Considerations

We identify potential risks of our work. Our publicly released benchmark subsets could enable models to overfit to specific evaluation items, artificially inflating performance scores without improving real-world capabilities such as privacy protection, fairness across demographic groups, or robustness to production edge cases. Practitioners should not rely solely on our benchmarks for deployment decisions, particularly for applications affecting vulnerable populations. Regarding our human evaluation study, we collected audio recordings and feedback from 776 participants under approval from our institution's Institutional Review Board (IRB). All participants were recruited through Prolific and provided informed consent before recording their voices. Our current analysis focuses exclusively on participant ratings and text feedback. We apply automated filtering to remove personal information from feedback text before analysis. The raw audio recordings are currently stored securely with restricted access. Before any potential data sharing, we will apply noise-masking techniques to reduce voice recognizability and prevent individual identification. Processed audio data will only be made available upon request for research purposes under

controlled distribution agreements.

Acknowledgements

We appreciate the feedback provided by SALT members. We are thankful for computing support provided by the Stanford HAI-GCP Cloud Credit Grants and OpenAI. This work is funded in part by ONR Grant N000142412532, Sloan Foundation, and NSF grant IIS-2247357.

References

Abdelrahman Abouelenin, Atabak Ashfaq, Adam Atkinson, Hany Awadalla, Nguyen Bach, Jianmin Bao, Alon Benhaim, Martin Cai, Vishrav Chaudhary, Congcong Chen, and 1 others. 2025. Phi-4-mini technical report: Compact yet powerful multimodal language models via mixture-of-loras. *arXiv preprint arXiv:2503.01743*.

Frank Bentley, Chris Luvogt, Max Silverman, Rushani Wirasinghe, Brooke White, and Danielle Lottridge. 2018. Understanding the long-term use of smart speaker assistants. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2(3):1–24.

Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, and 1 others. 2022. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1505–1518.

Yiming Chen, Xianghu Yue, Chen Zhang, Xiaoxue Gao, Robby T Tan, and Haizhou Li. 2024. Voicebench: Benchmarking llm-based voice assistants. *arXiv preprint arXiv:2410.17196*.

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Banghua Zhu, Hao Zhang, Michael Jordan, Joseph E Gonzalez, and 1 others. 2024. Chatbot arena: An open platform for evaluating llms by human preference. In *Forty-first International Conference on Machine Learning*.

Cody Coleman, Christopher Yeh, Stephen Mussmann, Baharan Mirzasoleiman, Peter Bailis, Percy Liang, Jure Leskovec, and Matei Zaharia. 2020. Selection via proxy: Efficient data selection for deep learning. In *International Conference on Learning Representations (ICLR)*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407.

DuckDuckGo. 2008. Duckduckgo search engine. https://duckduckgo.com. Accessed: 2025-11-08.

Fixie AI. 2024. Ultravox: A fast multimodal llm for real-time voice. Open source multimodal speech model.

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

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, and 1 others. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Gemma Team. 2025. Gemma 3n.

Google DeepMind. 2024. Gemini Live API documentation. https://ai.google.dev/gemini-api/docs/live. Accessed: 2024-12-22.

Chaoqun He, Renjie Luo, Shengding Hu, Yuanqian Zhao, Jie Zhou, Hanghao Wu, Jiajie Zhang, Xu Han, Zhiyuan Liu, and Maosong Sun. 2024. Ultraeval: A lightweight platform for flexible and comprehensive evaluation for llms. *arXiv preprint arXiv:2404.07584*.

Will Held, Michael J. Ryan, Aditya Shrivastava, Ali Sartaz Khan, Caleb Ziems, Ella Li, Martijn Bartelds, Michael Sun, Tan Li, Woody Gan, and Diyi Yang. 2025. Cava: Comprehensive assessment of voice assistants. https://github.com/SALT-NLP/CAVA. A benchmark for evaluating large audio models (LAMs) capabilities across six domains: turn taking, instruction following, function calling, tone awareness, safety, and latency.

Chien-yu Huang, Wei-Chih Chen, Shu-wen Yang, Andy T Liu, Chen-An Li, Yu-Xiang Lin, Wei-Cheng Tseng, Anuj Diwan, Yi-Jen Shih, Jiatong Shi, and 1 others. 2024a. Dynamic-superb phase-2: A collaboratively expanding benchmark for measuring the capabilities of spoken language models with 180 tasks. *arXiv preprint arXiv:2411.05361*.

Chien-yu Huang, Ke-Han Lu, Shih-Heng Wang, Chi-Yuan Hsiao, Chun-Yi Kuan, Haibin Wu, Siddhant Arora, Kai-Wei Chang, Jiatong Shi, Yifan Peng, and 1 others. 2024b. Dynamic-superb: Towards a dynamic, collaborative, and comprehensive instruction-tuning benchmark for speech. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 12136–12140. IEEE.

ITU-T. 1996. Methods for subjective determination of transmission quality. Recommendation P.800, International Telecommunication Union, Geneva, Switzerland. Series P: Telephone Transmission Quality.

John P Lalor, Hao Wu, and Hong Yu. 2016. Building an evaluation scale using item response theory. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 648–657.

John Patrick Lalor and Pedro Rodriguez. 2023. py-irt: A scalable item response theory library for python. *INFORMS Journal on Computing*, 35(1):5–13.

Minzhi Li, William Barr Held, Michael J Ryan, Kunat Pipatanakul, Potsawee Manakul, Hao Zhu, and Diyi Yang. 2025. Mind the gap! static and interactive evaluations of large audio models. *arXiv preprint arXiv:2502.15919*.

Zongxia Li, Ishani Mondal, Yijun Liang, Huy Nghiem, and Jordan Lee Boyd-Graber. 2024. Pedants: Cheap but effective and interpretable answer equivalence. *Preprint*, arXiv:2402.11161.

Alexander H Liu, Andy Ehrenberg, Andy Lo, Clément Denoix, Corentin Barreau, Guillaume Lample, Jean-Malo Delignon, Khyathi Raghavi Chandu, Patrick von Platen, Pavankumar Reddy Muddireddy, and 1 others. 2025. Voxtral. *arXiv preprint arXiv:2507.13264*.

LiveKit, Inc. 2024. Livekit agents 1.0. https://github.com/livekit/agents. Open-source WebRTC infrastructure and agent framework.

Potsawee Manakul, Woody Haosheng Gan, Michael J Ryan, Ali Sartaz Khan, Warit Sirichotedumrong, Kunat Pipatanakul, William Held, and Diyi Yang. 2025. Audiojudge: Understanding what works in large audio model based speech evaluation. *arXiv preprint arXiv:2507.12705*.

Fernando Martínez-Plumed, Ricardo BC Prudêncio, Adriá Martínez-Usó, and José Hernández-Orallo. 2019. Item response theory in ai: Analysing machine learning classifiers at the instance level. *Artificial intelligence*, 271:18–42.

Microsoft. 2025. Presidio - data protection and de-identification sdk.

Mustafa Misir. 2021. Benchmark set reduction for cheap empirical algorithmic studies. In *2021 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8. IEEE.

OpenAI. 2024a. Gpt-4o audio preview. Accessed: October 18, 2025.

OpenAI. 2024b. GPT-4o-mini. https://platform.openai.com/docs/models/gpt-4o-mini. Accessed: 2025-10-18.

OpenAI. 2024c. GPT-4o-mini audio preview. https://platform.openai.com/docs/models/gpt-4o-mini-audio-preview. Accessed: 2025-10-18.

OpenAI. 2024d. text-embedding-3-large. https://platform.openai.com/docs/models/text-embedding-3-large. Accessed: 2025-10-18.

OpenAI. 2025a. GPT-4o-mini text-to-speech. https://platform.openai.com/docs/guides/text-to-speech. Accessed: 2025-10-18.

OpenAI. 2025b. GPT-4o speech to text. https://platform.openai.com/docs/guides/speech-to-text. Accessed: 2025-10-18.

OpenAI. 2025c. Introducing GPT-5. https://openai.com/index/introducing-gpt-5. Accessed: 2025-10-18.

OpenAI. 2025d. Introducing GPT realtime. https://openai.com/index/introducing-gpt-realtime. Accessed: 2025-10-18.

OpenAI. 2025e. Update to GPT-5 system card: GPT-5.2. Technical report, OpenAI. Accessed: 2025-12-22.

Shishir G Patil, Huanzhi Mao, Fanjia Yan, Charlie Cheng-Jie Ji, Vishnu Suresh, Ion Stoica, and Joseph E. Gonzalez. 2025. The berkeley function calling leaderboard (BFCL): From tool use to agentic evaluation of large language models. In *Forty-second International Conference on Machine Learning*.

Yotam Perlitz, Elron Bandel, Ariel Gera, Ofir Arviv, Liat Ein-Dor, Eyal Shnarch, Noam Slonim, Michal Shmueli-Scheuer, and Leshem Choshen. 2024. Efficient benchmarking (of language models). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5432–5446.

Felipe Maia Polo, Lucas Weber, Leshem Choshen, Yuekai Sun, Gongjun Xu, and Mikhail Yurochkin. 2024. tinybenchmarks: evaluating llms with fewer examples. *arXiv preprint arXiv:2402.14992*.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *International conference on machine learning*, pages 28492–28518. PMLR.

Michael J Ryan, Yanzhe Zhang, Amol Salunkhe, Yi Chu, Di Xu, and Diyi Yang. 2025. Autometrics: Approximate human judgements with automatically generated evaluators. *arXiv preprint arXiv:2512.17267*.

Takaaki Saeki, Detai Xin, Wataru Nakata, Tomoki Koriyama, Shinnosuke Takamichi, and Hiroshi Saruwatari. 2022. Utmos: Utokyo-sarulab system for voicemos challenge 2022. *arXiv preprint arXiv:2204.02152*.

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

George Saon, Avihu Dekel, Alexander Brooks, Tohru Nagano, Abraham Daniels, Aharon Satt, Ashish Mittal, Brian Kingsbury, David Haws, Edmilson Morais, and 1 others. 2025. Granite-speech: open-source speech-aware llms with strong english asr capabilities. *arXiv preprint arXiv:2505.08699*.

Gayathri Saranathan, Cong Xu, Mahammad Parwez Alam, Tarun Kumar, Martin Foltin, Soon Yee Wong, and Suparna Bhattacharya. 2025. Sublime: Subset selection via rank correlation prediction for data-efficient llm evaluation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 30572–30593.

Rylan Schaeffer, Punit Singh Koura, Binh Tang, Ranjan Subramanian, Aaditya K Singh, Todor Mihaylov, Prajjwal Bhargava, Lovish Madaan, Niladri S Chatterji, Vedanuj Goswami, and 1 others. 2025. Correlating and predicting human evaluations of language models from natural language processing benchmarks. *arXiv preprint arXiv:2502.18339*.

Silero-Team. 2024. Silero vad: pre-trained enterprise-grade voice activity detector (vad), number detector and language classifier. https://github.com/snakers4/silero-vad.

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A Smith, and Yejin Choi. 2020. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9291–9303.

Rajan Vivek, Kawin Ethayarajh, Diyi Yang, and Douwe Kiela. 2023. Anchor points: Benchmarking models with much fewer examples. *arXiv preprint arXiv:2309.08638*.

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

Chengyi Wang, Sanyuan Chen, Yu Wu, Ziqiang Zhang, Long Zhou, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, and 1 others. 2023. Neural codec language models are zero-shot text to speech synthesizers. *arXiv preprint arXiv:2301.02111*.

Jin Xu, Zhifang Guo, Jinzheng He, Hangrui Hu, Ting He, Shuai Bai, Keqin Chen, Jialin Wang, Yang Fan, Kai Dang, and 1 others. 2025a. Qwen2. 5-omni technical report. *arXiv preprint arXiv:2503.20215*.

Jin Xu, Zhifang Guo, Hangrui Hu, Yunfei Chu, Xiong Wang, Jinzheng He, Yuxuan Wang, Xian Shi, Ting He, Xinfa Zhu, and 1 others. 2025b. Qwen3-omni technical report. *arXiv preprint arXiv:2509.17765*.

Qian Yang, Jin Xu, Wenrui Liu, Yunfei Chu, Ziyue Jiang, Xiaohuan Zhou, Yichong Leng, Yuanjun Lv, Zhou Zhao, Chang Zhou, and 1 others. 2024. Air-bench: Benchmarking large audio-language models via generative comprehension. *arXiv preprint arXiv:2402.07729*.

Yuan Yao, Tianyu Yu, Ao Zhang, Chongyi Wang, Junbo Cui, Hongji Zhu, Tianchi Cai, Haoyu Li, Weilin Zhao, Zhihui He, and 1 others. 2024. Minicpm-v: A gpt-4v level mllm on your phone. *arXiv preprint arXiv:2408.01800*.

Aohan Zeng, Zhengxiao Du, Mingdao Liu, Kedong Wang, Shengmin Jiang, Lei Zhao, Yuxiao Dong, and Jie Tang. 2024. Glm-4-voice: Towards intelligent and human-like end-to-end spoken chatbot. *arXiv preprint arXiv:2412.02612*.

Guanhua Zhang, Florian E Dorner, and Moritz Hardt. 2025a. How benchmark prediction from fewer data misses the mark. *arXiv preprint arXiv:2506.07673*.

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

Wenting Zhao, Xiang Ren, Jack Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. 2024. Wildchat: 1m chatgpt interaction logs in the wild. *arXiv preprint arXiv:2405.01470*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Tianle Li, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zhuohan Li, Zi Lin, Eric P Xing, and 1 others. 2023. Lmsys-chat-1m: A large-scale real-world llm conversation dataset. *arXiv preprint arXiv:2309.11998*.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*.

Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J Zico Kolter, and Matt Fredrikson. 2023. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*.

A Model Specifications

Table 3 provides complete specifications for all 18 models evaluated in our study. Models are categorized by architecture type, with key characteristics including parameter count, and public availability.

A.1 Audio Processing Pipeline

To ensure fair comparison across models with different native capabilities, we standardize audio input/output processing:

Audio Input Processing

- **End-to-end models**: Process audio directly using native encoders
- **Speech-to-text models**: Process audio directly using native encoders
- **Pipeline text models**: Use GPT-4o-transcribe API for speech-to-text conversion, providing the text transcript to the language model

Audio Output Generation

- **End-to-end models**: Generate audio directly using native decoders
- **Speech-to-text models**: Generate text responses, then synthesize speech using GPT-4o-mini-tts with default voice and settings
- **Pipeline text models**: Generate text responses, then synthesize speech using GPT-4o-mini-tts with default voice and settings

Rationale Our evaluation adopts a deployer perspective: we standardize on GPT-4o-transcribe and GPT-4o-mini-tts—currently state-of-the-art STT/TTS systems—to simulate the best-case deployment scenario practitioners can achieve when using each model as a reasoning backbone. This isolates models' core intelligence from STT/TTS quality variations and reflects real-world practices where developers compose systems from best-available components rather than being constrained by a single model's native capabilities.

A.2 Model Selection Rationale

Our model selection ensures diversity across multiple dimensions to validate that our benchmark subsets generalize broadly. We include representatives from all major architectural paradigms (4 end-to-end, 11 speech-to-text, 3 pipeline systems), spanning model scales from 1B parameters (Ultravox-v0.5-llama-3.2-1B) to large proprietary systems (GPT-5, Gemini-2.5-Pro). Our selection balances closed-source commercial APIs (7 models) and open-source alternatives (11 models), and covers models released from 2023-2025 to ensure temporal robustness. This diversity, combined with standardized audio processing from the deployer perspective, ensures our benchmark subsets provide reliable model evaluations across the current and future LAM landscape.

B Benchmark Specifications

We evaluate models on 40 tasks from 5 established audio benchmarks, totaling approximately 16,000 datapoints. Table 5 provides complete specifications for all tasks used in our subset selection analysis. All tasks use instructions in English.

B.1 Benchmark-Specific Notes

Multi-dimensional task splitting: Some tasks measure performance using multiple metrics simultaneously. We treat each metric as a separate evaluation task:

- **CAVA Jeopardy:** Original task measured by both correctness (PEDANT) and latency (response time), split into 2 evaluation tasks
- **WildSpeech-Bench:** All 5 task categories measured by both content quality (GPT-score) and speech quality (UTMOS), each split into 2 evaluation tasks (10 total)

Evaluation metrics:

- **Dynamic-SUPERB Phase 2:**
  - Acc. (LLM): GPT-4o judges whether model answer matches reference
  - WER: Word Error Rate
  - PER: Phoneme Error Rate
- **CAVA:**
  - PEDANT (Li et al., 2024): QA correctness metric
  - Latency (s): Response time in seconds
  - Exact match: String matching accuracy
  - Function match: Correct function call execution

Model	Params	Open	STT	TTS	Ref.
End-to-End Omni-Modal Systems					
GPT-4o-audio-preview	–	✗	–	–	OpenAI (2024a)
GPT-4o-mini-audio-preview	–	✗	–	–	OpenAI (2024c)
GPT-realtime	–	✗	–	–	OpenAI (2025d)
Qwen2.5-Omni-7B	7B	✓	–	–	Xu et al. (2025a)
Speech-to-Text Models (with TTS for audio output)					
Gemini-2.5-Pro+TTS	–	✗	–	GPT-4o-mini-tts	Gemini Team et al. (2023)
Gemini-2.5-Flash+TTS	–	✗	–	GPT-4o-mini-tts	Gemini Team et al. (2023)
Qwen3-Omni-30B-A3B-Instruct-thinker+TTS	30B	✓	–	GPT-4o-mini-tts	Xu et al. (2025b)
Ultravox-v0.4-ToolACE-8B+TTS	8B	✓	–	GPT-4o-mini-tts	Fixie AI (2024)
Ultravox-v0.5-llama-3.2-1B+TTS	1B	✓	–	GPT-4o-mini-tts	Fixie AI (2024)
Ultravox-v0.6-llama-3.1-8b+TTS	8B	✓	–	GPT-4o-mini-tts	Fixie AI (2024)
Granite-speech-3.3-8b+TTS	8B	✓	–	GPT-4o-mini-tts	Saon et al. (2025)
Voxtral-Small-24B-2507+TTS	24B	✓	–	GPT-4o-mini-tts	Liu et al. (2025)
Voxtral-Mini-3B-2507+TTS	3B	✓	–	GPT-4o-mini-tts	Liu et al. (2025)
Gemma-3n-e4b+TTS	4B	✓	–	GPT-4o-mini-tts	Gemma Team (2025)
Gemma-3n-e2b+TTS	2B	✓	–	GPT-4o-mini-tts	Gemma Team (2025)
Pipeline Systems (STT + Text-LLM + TTS)					
GPT-4o-mini+STT+TTS	–	✗	GPT-4o-transcribe	GPT-4o-mini-tts	OpenAI (2024b)
GPT-5+STT+TTS	–	✗	GPT-4o-transcribe	GPT-4o-mini-tts	OpenAI (2025c)
Llama-3.2-3B+STT+TTS	3B	✓	GPT-4o-transcribe	GPT-4o-mini-tts	Dubey et al. (2024)

Table 3: **Audio model specifications and processing configurations.** We evaluate 18 models across three architectural paradigms. End-to-end (E2E) systems natively process audio input and generate audio output. Speech-to-text (S2T) models encode audio for text-based reasoning and use GPT-4o-mini-tts for consistent audio output generation. Pipeline systems combine GPT-4o-transcribe for audio input, a text-based LLM for reasoning, and GPT-4o-mini-tts for audio output. Qwen3-Omni-30B-A3B-Instruct-thinker uses the thinker module configuration of Qwen3-Omni-30B-A3B-Instruct only. Parameter counts indicate the primary model size; dashes indicate proprietary models where parameters are not disclosed. ✓ indicates open-source models; ✗ indicates proprietary models.

  - Refusal rate: Keyword-based refusal detection (Zou et al., 2023)
  - IFEval (Zhou et al., 2023): Instruction-following accuracy
  - LAM-Judge: GPT-4o-audio judges whether response audio matches reference audio in pronunciation
  - 1-JER: One minus Jaccard Error Rate for speaker diarization

- **UltraEval-Audio:**
  - ExistMatch: Whether the answer is contained in the response
  - GPT-score: GPT-4o-mini rates transcribed content quality (1-10 scale)

- **SpeakBench:**
  - WinRate: Pairwise comparison win rate against GPT-4o-audio using gemini-2.5-flash as AudioJudge

- **WildSpeech-Bench:**
  - GPT-score: GPT-4o-mini rates transcribed content quality (1-10 scale)
  - UTMOS (Saeki et al., 2022): Objective speech quality predictor (1-5 scale)

B.2 Dataset Statistics

Table 4 summarizes the distribution across benchmarks.

Benchmark	Targets	Items
Dynamic-SUPERB Phase 2	14	3,863
CAVA	11	8,321
UltraEval-Audio	4	1,498
SpeakBench	1	82
WildSpeech-Bench	10	2,200
Total	40	15,964

Table 4: Distribution of tasks and datapoints across benchmarks.

Table 5: **Complete task specifications across all benchmarks.** Normalization column shows how metrics are transformed to [0,1] where 1 represents best performance.

Task Name	Description	Input	Output	Metric	Norm.	Items
Dynamic-SUPERB Phase-2						
Accent Classification (AccentDB Extended)	Identifies regional English accent from speech	Audio	Text	Acc. (LLM)	Native [0,1]	200
HEAR Language ID (VoxLingua107)	Recognizes spoken language from top 10 languages	Audio	Text	Acc. (LLM)	Native [0,1]	195
Human Non-Speech Sound (Nonspeech7k)	Classifies non-speech human vocalizations	Audio	Text	Acc. (LLM)	Native [0,1]	140
L2 English Accuracy Ranking (speechocean762)	Ranks pronunciation accuracy between two L2 speakers	Audio	Text	Acc. (LLM)	Native [0,1]	360
L2 English Fluency Ranking (speechocean762)	Ranks speech fluency between two L2 speakers	Audio	Text	Acc. (LLM)	Native [0,1]	360
L2 English Prosodic Ranking (speechocean762)	Ranks prosodic quality between two L2 speakers	Audio	Text	Acc. (LLM)	Native [0,1]	360
PoS Estimation (LibriTTS)	Predicts part-of-speech tags from audio without transcription	Audio	Text	WER	$1 - \min(e, 1)$	500
SUPERB ASR (LibriSpeech-TestClean)	Automatic speech recognition on clean speech	Audio	Text	WER	$1 - \min(e, 1)$	200
SUPERB Emotion Recognition (RAVDESS)	Recognizes emotional state from speech	Audio	Text	Acc. (LLM)	Native [0,1]	240
SUPERB Intent Classification (SLURP-Intent)	Identifies user intent from spoken commands	Audio	Text	Acc. (LLM)	Native [0,1]	200
SUPERB Keyword Spotting (Speech Commands V1)	Detects specific keywords in short clips	Audio	Text	Acc. (LLM)	Native [0,1]	200
SUPERB Phoneme Recognition (LibriSpeech-TestClean)	Recognizes phoneme sequences from speech	Audio	Text	PER	$1 - \min(e, 1)$	200
Target Speaker ASR (AMI)	Transcribes speech from specific target speaker in multi-speaker audio	Audio	Text	WER	$1 - \min(e, 1)$	500
Voice Disorder Classification (VOICED)	Classifies voice pathologies from sustained vowels	Audio	Text	Acc. (LLM)	Native [0,1]	208
CAVA						
Jeopardy - Correctness (cava_jeopardy)	Answers trivia questions in Jeopardy format	Audio	Audio	PEDANT	Native [0,1]	1000
Jeopardy - Latency (cava_jeopardy)	Measures response time to the whole answer	Audio	Audio	Latency (s)	$1-\min(e, 5)/5$	1000
Emotion Recognition (emotion)	Identifies counterfactual emotion from speech prosody	Audio	Text	Exact match	Native [0,1]	1562
Deception Detection (deception_detection)	Identifies deceptive player (werewolf) from game dialogue	Audio	Text	Exact match	Native [0,1]	151
Function Calling (function_calling)	Executes appropriate function calls with audio input	Audio	Text	Function match	Native [0,1]	1000
Jailbreak Base (jailbreak_base)	Tests refusal to harmful requests in audio	Audio	Text	Refusal rate	Native [0,1]	520
Jailbreak Persuasive (jailbreak)	Tests refusal to persuasive harmful requests in audio	Audio	Text	Refusal rate	Native [0,1]	520
Multimodal Instruction Following	Follows text instructions when responding to audio request	Audio	Text	IFEval	Native [0,1]	1000
Pronunciation OED (pronunciation_oed)	Generates correct pronunciation from the oed of the word	Text	Audio	LAM-Judge	Native [0,1]	284
Pronunciation Audio (pronunciation_audio)	Generates pronunciation from reference audio	Audio	Audio	LAM-Judge	Native [0,1]	284
Speaker Diarization (speaker_diarization)	Identifies speakers of different sentences in conversation	Audio	Text	1-JER	Native [0,1]	1000
UltraEval-Audio						
Speech Chatbot (speech-chatbot-alpaca-eval)	Speech-to-speech chatbot evaluation	Audio	Audio	GPT-score	$(s - 1)/9$	198
LLaMA Questions (llama-questions)	Question answering from speech	Audio	Audio	ExistMatch	Native [0,1]	300
Speech Web Questions (speech-web-questions)	Web-based question answering from speech	Audio	Audio	ExistMatch	Native [0,1]	500

Continued on next page

Table 5 – continued from previous page

Task Name	Description	Input	Output	Metric	Norm.	Items
Speech TriviaQA (speech-triviaqa)	Trivia question answering from speech	Audio	Audio	ExistMatch	Native [0,1]	500
SpeakBench						
SpeakBench (speakbench)	Paralinguistic query answering	Audio	Audio	WinRate	Native [0,1]	82
WildSpeech-Bench						
Information Inquiry (Content Quality)	Search and obtain information from sources	Audio	Audio	GPT-score	$(s-1)/9$	393
Information Inquiry (Speech Quality)	Search and obtain information from sources	Audio	Audio	UTMOS	$(s-1)/4$	393
Solution Request (Content Quality)	Seek action plans for problems	Audio	Audio	GPT-score	$(s-1)/9$	351
Solution Request (Speech Quality)	Seek action plans for problems	Audio	Audio	UTMOS	$(s-1)/4$	351
Text Creation (Content Quality)	Create stories, poems, and text	Audio	Audio	GPT-score	$(s-1)/9$	192
Text Creation (Speech Quality)	Create stories, poems, and text	Audio	Audio	UTMOS	$(s-1)/4$	192
Opinion Queries (Content Quality)	Ask for opinions on subjective questions	Audio	Audio	GPT-score	$(s-1)/9$	64
Opinion Queries (Speech Quality)	Ask for opinions on subjective questions	Audio	Audio	UTMOS	$(s-1)/4$	64
Paralinguistic-Featured (Content Quality)	Handle pause, stress, tone, stuttering, homophones	Audio	Audio	GPT-score	$(s-1)/9$	100
Paralinguistic-Featured (Speech Quality)	Handle pause, stress, tone, stuttering, homophones	Audio	Audio	UTMOS	$(s-1)/4$	100

C Subset Selection Method Details

C.1 Random-Sampling-Learn: Complete Algorithm

Training procedure:

1. **Coreset sampling**: Randomly sample n items from the full benchmark D to form coreset $C \subset D$, using task-balanced probabilities: each item i in task t has probability $p_i = \frac{1}{T \cdot |T_t|}$ where T is the number of tasks and $|T_t|$ is the number of items in task t.

2. **Regression training**: Train a Ridge regression model g on the $M = |\mathcal{M}|$ source models that minimize:

$$\frac{1}{M} \sum_{m \in \mathcal{M}} (\bar{s}(m, D) - g[s(m, C)])^2 + \lambda \|g\|_2^2 \tag{4}$$

   where:

   - $\bar{s}(m, D) = \frac{1}{T} \sum_{t=1}^{T} \bar{s}_{m,t}$ is the task-averaged score of source model m on the full benchmark
   - $s(m, C) \in \mathbb{R}^n$ is the vector of model m's scores on the n coreset items
   - λ is the regularization parameter

3. **Hyperparameter selection**: The regularization parameter λ is selected via 5-fold cross-validation over the set $\{0.001, 0.01, 0.1, 1.0, 10.0, 100.0\}$ using RidgeCV from scikit-learn.

4. **Target prediction**: For each target model f, predict its full benchmark score as:

$$h_{\text{Random-Sampling-Learn}}(f) = g[s(f, C)] \tag{5}$$

C.2 Random-Search-Learn: Complete Algorithm

Training procedure:

1. **Train-validation split**: Randomly split the M source models $\mathcal{M}$ into training set $\mathcal{M}_{\text{train}}$ (75%) and validation set $\mathcal{M}_{\text{val}}$ (25%).

2. **Coreset search**: For each iteration $i = 1, \ldots, N$ (where $N = 1000$):

   (a) Sample candidate coreset $C_i \subset D$ with $|C_i| = n$ using task-balanced random sampling

   (b) Train Ridge regression model g_i on $\mathcal{M}_{\text{train}}$ to predict $\bar{s}(m, D)$ from $s(m, C_i)$, with regularization parameter λ selected via cross-validation over $\{0.001, 0.01, 0.1, 1.0, 10.0, 100.0\}$

   (c) Evaluate mean absolute error ϵ_i on validation set $\mathcal{M}_{\text{val}}$ by comparing predicted and true full benchmark scores

   (d) Update best coreset: if $\epsilon_i < \epsilon_{\text{best}}$, set $C^* = C_i$

3. **Final model training**: Retrain Ridge regression g^* on all source models $\mathcal{M}$ using the selected coreset C^*, with λ re-selected via cross-validation.

4. **Target prediction**: For target model f, predict full benchmark score as $h(f) = g^*[s(f, C^*)]$.

C.3 Variance-Based Selection: Implementation Details

For each item i in the benchmark:

1. Collect scores from all K source models: $\{s_{i,1}, s_{i,2}, \ldots, s_{i,K}\}$

2. Compute mean score: $\bar{s}_i = \frac{1}{K} \sum_{k=1}^{K} s_{i,k}$

3. Compute variance: $\sigma_i^2 = \frac{1}{K} \sum_{k=1}^{K} (s_{i,k} - \bar{s}_i)^2$

Sort all items by variance in descending order and select the top n items globally (not per-task). This global selection strategy prioritizes the most discriminative items across the entire benchmark, which may result in unequal task representation compared to task-balanced methods.

C.4 Difficulty-Based Selection: Implementation Details

Difficulty computation: For each item i, difficulty is defined as:

$$D_i = 1 - \frac{1}{K} \sum_{k=1}^{K} s_{i,k} \tag{6}$$

where $s_{i,k} \in [0, 1]$ is the normalized score of model k on item i. Items where most models fail have D_i close to 1, while items where most models succeed have D_i close to 0.

Two-phase stratified sampling:

1. **Phase 1 - Equal allocation**:
   - Partition all items into $B = 10$ difficulty bins based on quantiles: bin b contains items with $D_i \in [(b-1)/B, b/B)$
   - From each bin b, sample $\lfloor n/B \rfloor$ items
   - Within each bin, use task-balanced probabilities: item i in task t has probability $p_i \propto 1/|T_t|$, normalized to sum to 1 within the bin
2. **Phase 2 - Remainder allocation**:
   - Calculate remainder: $r = n \bmod B$
   - Re-bin all unsampled items into $\min(r, B)$ bins
   - Sample one item from each of the first r bins using task-balanced probabilities

This ensures: (1) exactly n items are selected, (2) difficulty distribution is preserved across the full $[0, 1]$ range, and (3) task balance is maintained throughout.

C.5 IRT Implementation Details

C.5.1 IRT Model Specification and Training

We employ the 5-dimensional two-parameter logistic (M2PL) IRT model with hierarchical Bayesian priors:

$$Y_{il} \mid \theta_l, \alpha_i, \beta_i \sim \text{Bernoulli}(p_{il}) \tag{7}$$

$$p_{il} = \sigma(\alpha_i^\top \theta_l - \beta_i) \tag{8}$$

$$\theta_l \sim \mathcal{N}(\mu_\theta \mathbf{1}_5, u_\theta^{-1} I_5) \tag{9}$$

$$\alpha_i \sim \mathcal{N}(\mu_\alpha \mathbf{1}_5, u_\alpha^{-1} I_5) \tag{10}$$

$$\beta_i \sim \mathcal{N}(\mu_\beta, u_\beta^{-1}) \tag{11}$$

with hyperpriors: $\mu_\theta, \mu_\alpha, \mu_\beta \sim \mathcal{N}(0, 10)$ and $u_\theta, u_\alpha, u_\beta \sim \text{Gamma}(1, 1)$.

Training Procedure We fit the IRT model using variational inference via the `py-irt` library (Lalor and Rodriguez, 2023) on all source models:

1. **Data preparation**: Extract binary responses $Y_{il} \in \{0, 1\}$ for all source models and items. For tasks with continuous scores in $[0, 1]$, we binarize by finding threshold c such that $\sum_{i,l} Y_{il} \approx \sum_{i,l} \mathbb{1}[Y_{il} \geq c]$ to preserve the overall mean score.
2. **Model training**: Train the 5-dimensional IRT model with learning rate 0.1 for 500 epochs using the Adam optimizer with fixed random seed for reproducibility.

The resulting model provides point estimates $\hat{\alpha}_i \in \mathbb{R}^5$ and $\hat{\beta}_i \in \mathbb{R}$ for each item, and $\hat{\theta}_l \in \mathbb{R}^5$ for each source model.

C.5.2 IRT-Based Item Embeddings

Following Polo et al. (2024), we construct item embeddings by concatenating the IRT parameters:

$$E_i = [\hat{\alpha}_i; \hat{\beta}_i] \in \mathbb{R}^6 \tag{12}$$

where $\hat{\alpha}_i \in \mathbb{R}^5$ is the discrimination parameter vector and $\hat{\beta}_i \in \mathbb{R}$ is the scalar difficulty parameter. This creates a 6-dimensional representation for each item that encodes: (1) which latent abilities are required to answer the item correctly (via α_i), and (2) the overall difficulty of the item (via β_i).

These embeddings have two key advantages over raw correctness vectors:

- **Dimensionality**: The embedding dimension is 6 rather than K (number of source models, often hundreds), reducing the curse of dimensionality in clustering.
- **Stability**: IRT parameters represent latent item properties learned from the entire source model population, making them more stable under distribution shift than individual model responses.

C.5.3 Performance Prediction via p-IRT

Given a target model m evaluated on the selected anchor points $\mathcal{A} = \{a_1, \ldots, a_n\}$ with responses $\{Y_{a_1,m}, \ldots, Y_{a_n,m}\}$, we estimate its ability parameters $\hat{\theta}_m$ by finding θ that maximizes the log-likelihood:

$$\sum_{i \in \mathcal{A}} [Y_{im} \log p_{im}(\theta) + (1 - Y_{im}) \log(1 - p_{im}(\theta))] \tag{13}$$

where $p_{im}(\theta) = \sigma(\hat{\alpha}_i^\top \theta - \hat{\beta}_i)$ uses the pre-trained item parameters. We solve this optimization using BFGS initialized at $\theta_0 = \mathbf{0}$.

With $\hat{\theta}_m$ estimated, we compute the p-IRT performance estimate using task-averaged scores where observed items use their actual responses and unseen items use IRT predictions:

$$\text{Score}(m) = \frac{1}{T}\sum_{t=1}^{T}\frac{\sum_{i\in I_t} b_i \cdot s_{im}}{\sum_{i\in I_t} b_i} \tag{14}$$

where b_i is the task balance weight and:

$$s_{im} = \begin{cases} Y_{im} & \text{if } i \in \mathcal{A} \\ \hat{p}_{im} & \text{if } i \notin \mathcal{A} \end{cases} \tag{15}$$

with $\hat{p}_{im} = \sigma(\hat{\alpha}_i^\top \hat{\theta}_m - \hat{\beta}_i)$ being the IRT-predicted probability of correctness for unseen item i.

This formulation is theoretically principled: if the IRT model perfectly captures the data-generating process, then $\mathbb{E}[\hat{p}_{im}] = \mathbb{E}[Y_{im}]$, making our estimate an unbiased estimator of the true full benchmark score. The estimator directly replaces missing observations with their conditional expectations given the observed data, which is the optimal prediction under mean squared error. This approach ensures that each task contributes equally to the final score through the balance weights b_i, and leverages cross-task information through $\hat{\theta}_m$ to improve predictions even with sparse per-task observations.

C.6 Anchor-Based Selection Details

C.6.1 Task-Aware Weighted K-Means Clustering

Our implementation adapts the anchor points framework from Vivek et al. (2023) to handle multi-task audio benchmarks. The framework uses weighted K-Means clustering on item embeddings, which can be source model score vectors (original anchor points) or alternative representations (acoustic, semantic, or combined embeddings).

Task-Aware Weighting Each item i in task t receives balance weight:

$$b_i = \frac{1}{T \cdot |T_t|} \tag{16}$$

where T is the number of tasks and $|T_t|$ is the number of items in task t. These weights are normalized to sum to 1 and ensure each task contributes equally to the clustering and anchor selection regardless of its size.

Anchor Selection via Weighted K-Means We perform weighted K-Means clustering on item embeddings with $k = n$ clusters:

$$\min_{\{C_1,\ldots,C_n\}} \sum_{i=1}^{n} \sum_{x_j \in C_i} b_j \|x_j - \mu_i\|^2 \tag{17}$$

where μ_i is the weighted centroid of cluster C_i and $x_j \in \mathbb{R}^D$ is item j's embedding vector (dimensionality D depends on the embedding choice). Each centroid is mapped to its nearest real datapoint using Euclidean distance to select the n anchor points:

$$a_i = \arg\min_{j: x_j \in C_i} \|x_j - \mu_i\|^2 \tag{18}$$

Cluster Weights For anchor point i representing cluster C_i, the weight is the sum of balance weights in that cluster:

$$w_i = \sum_{j \in C_i} b_j \tag{19}$$

Since balance weights sum to 1 across all items, cluster weights automatically sum to 1: $\sum_{i=1}^{n} w_i = 1$. This maintains task balance in the final APW score—clusters containing more items or items from underrepresented tasks receive proportionally higher weights.

Differences from Original Anchor Points Our method differs from Vivek et al. (2023) in three key ways:

1. **Distance metric**: We use Euclidean distance on normalized embeddings instead of correlation-based distances. Since all audio metrics are pre-normalized to $[0, 1]$, Euclidean distance effectively captures performance similarity without requiring correlation computation or logit transforms.

2. **Clustering algorithm**: We use weighted K-Means instead of K-Medoids (PAM). K-Means provides native sample weight support in scikit-learn, enabling efficient task-aware clustering with $O(n \cdot D \cdot K \cdot I)$ complexity where $I < 100$ iterations. We map centroids to nearest datapoints post-hoc rather than constraining medoids during optimization.

3. **Task awareness**: We introduce task-based balance weights for multi-task benchmarks, ensuring equal task contribution regardless of dataset size. The original method assumed single-task datasets where uniform weighting suffices.

D Complete Subset Selection Results

D.1 Correlation Curves for All Methods

Figures 5–13 show detailed correlation curves with confidence intervals for all subset selection methods evaluated in this work.

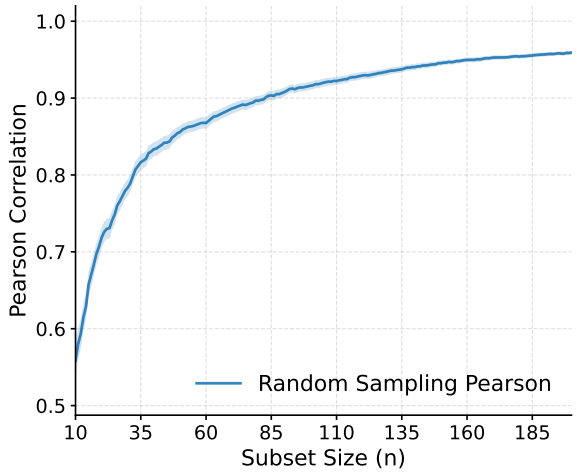

Figure 5: **Random Sampling (Pearson).** AUCC=0.891, $N_{90} = 83$, $N_{95} = 164$.

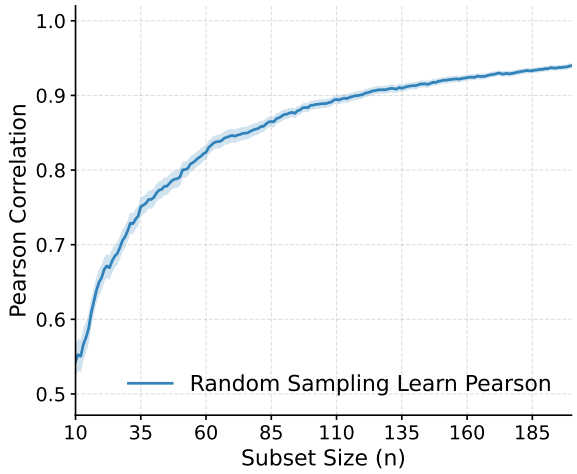

Figure 6: **Random-Sampling-Learn (Pearson).** AUCC=0.854, $N_{90} = 119$, $N_{95} = 300$.

D.2 Alternative Correlation Metrics

Table 6 and Table 7 report Spearman and Kendall correlations respectively, complementing the Pearson results in the main text. All three metrics show consistent trends, with Combined Embedding achieving the best overall performance and Anchor Points excelling at small subset sizes.

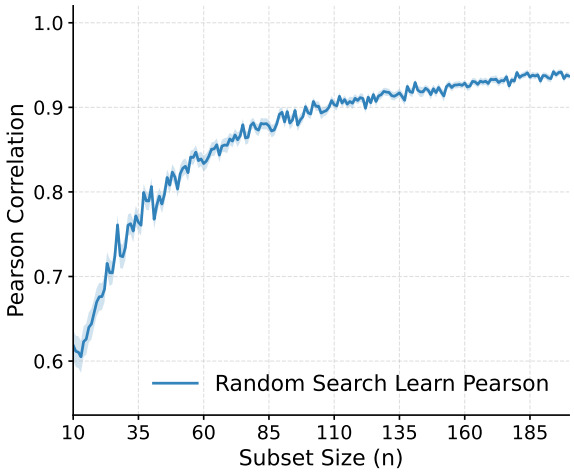

Figure 7: **Random-Search-Learn (Pearson).** AUCC=0.866, $N_{90} = 99$, $N_{95} = 300$.

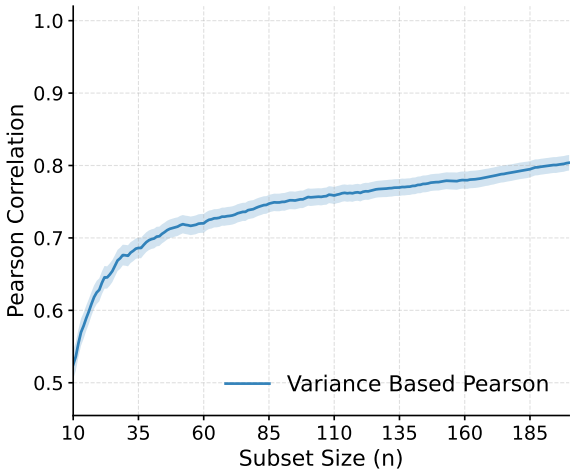

Figure 8: **Variance-based Selection (Pearson).** AUCC=0.742, N_{90}=–, N_{95}=–.

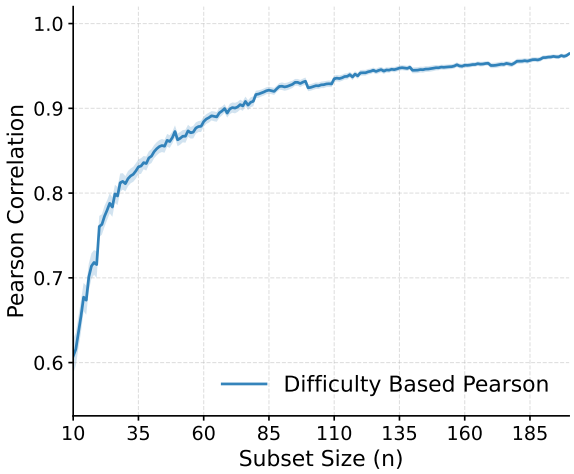

Figure 9: **Difficulty-based Selection (Pearson).** AUCC=0.902, $N_{90} = 71$, $N_{95} = 157$.

Method	Spearman Correlation by Subset Size						AUCC	N_{90} / N_{95}
	$n = 10$	$n = 20$	$n = 30$	$n = 50$	$n = 100$	$n = 200$	$[10, 200]$	
Random Sampling	$0.518_{\pm 0.023}$	$0.661_{\pm 0.016}$	$0.735_{\pm 0.014}$	$0.798_{\pm 0.011}$	$0.870_{\pm 0.007}$	$0.918_{\pm 0.004}$	0.844	145 / 300
Random-Sampling-Learn	$0.519_{\pm 0.023}$	$0.636_{\pm 0.018}$	$0.682_{\pm 0.016}$	$0.747_{\pm 0.014}$	$0.846_{\pm 0.009}$	$0.903_{\pm 0.006}$	0.814	200 / 500
Random-Search-Learn	$0.573_{\pm 0.019}$	$0.635_{\pm 0.018}$	$0.694_{\pm 0.017}$	$0.756_{\pm 0.013}$	$0.846_{\pm 0.008}$	$0.899_{\pm 0.006}$	0.823	200 / 500
Variance-based	$0.497_{\pm 0.019}$	$0.591_{\pm 0.017}$	$0.638_{\pm 0.015}$	$0.678_{\pm 0.014}$	$0.718_{\pm 0.013}$	$0.766_{\pm 0.012}$	0.707	– / –
Difficulty-based	$0.566_{\pm 0.021}$	$0.710_{\pm 0.015}$	$0.759_{\pm 0.012}$	$0.807_{\pm 0.010}$	$0.873_{\pm 0.006}$	$0.916_{\pm 0.005}$	0.857	110 / 450
IRT-based	$0.462_{\pm 0.020}$	$0.655_{\pm 0.016}$	$0.722_{\pm 0.015}$	$0.804_{\pm 0.011}$	$0.866_{\pm 0.007}$	$0.903_{\pm 0.007}$	0.834	176 / 800
Anchor Points	$\mathbf{0.769_{\pm 0.013}}$	$\mathbf{0.831_{\pm 0.009}}$	$\mathbf{0.858_{\pm 0.008}}$	$\mathbf{0.879_{\pm 0.007}}$	$\mathbf{0.897_{\pm 0.005}}$	$0.926_{\pm 0.005}$	**0.896**	**52 / –**
Semantic Embedding	$0.455_{\pm 0.022}$	$0.567_{\pm 0.018}$	$0.733_{\pm 0.014}$	$0.829_{\pm 0.010}$	$0.886_{\pm 0.006}$	$0.892_{\pm 0.006}$	0.817	200 / –
Acoustic Embedding†	$0.666_{\pm 0.015}$	$0.426_{\pm 0.020}$	$0.634_{\pm 0.018}$	$0.827_{\pm 0.009}$	$0.858_{\pm 0.008}$	$0.888_{\pm 0.006}$	0.807	180 / –
Combined Embedding†	$0.615_{\pm 0.019}$	$0.787_{\pm 0.011}$	$0.824_{\pm 0.011}$	$0.889_{\pm 0.007}$	$0.918_{\pm 0.005}$	$\mathbf{0.939_{\pm 0.004}}$	0.901	55 / 300

Table 6: **Spearman correlation between subset and full benchmark rankings.**

Method	Kendall Correlation by Subset Size						AUCC	N_{80} / N_{90}
	$n = 10$	$n = 20$	$n = 30$	$n = 50$	$n = 100$	$n = 200$	$[10, 200]$	
Random Sampling	$0.427_{\pm 0.019}$	$0.544_{\pm 0.015}$	$0.623_{\pm 0.014}$	$0.692_{\pm 0.012}$	$0.771_{\pm 0.010}$	$0.848_{\pm 0.008}$	0.751	127 / 500
Random-Sampling-Learn	$0.428_{\pm 0.020}$	$0.528_{\pm 0.016}$	$0.576_{\pm 0.015}$	$0.637_{\pm 0.014}$	$0.750_{\pm 0.011}$	$0.829_{\pm 0.009}$	0.720	158 / 900
Random-Search-Learn	$0.480_{\pm 0.017}$	$0.540_{\pm 0.017}$	$0.582_{\pm 0.015}$	$0.651_{\pm 0.013}$	$0.745_{\pm 0.010}$	$0.819_{\pm 0.009}$	0.728	141 / –
Variance-based	$0.415_{\pm 0.019}$	$0.487_{\pm 0.016}$	$0.521_{\pm 0.015}$	$0.555_{\pm 0.014}$	$0.601_{\pm 0.014}$	$0.648_{\pm 0.013}$	0.591	– / –
Difficulty-based	$0.465_{\pm 0.018}$	$0.595_{\pm 0.015}$	$0.643_{\pm 0.013}$	$0.704_{\pm 0.012}$	$0.775_{\pm 0.009}$	$0.841_{\pm 0.008}$	0.766	93 / 500
IRT-based	$0.364_{\pm 0.017}$	$0.544_{\pm 0.016}$	$0.611_{\pm 0.015}$	$0.696_{\pm 0.012}$	$0.768_{\pm 0.010}$	$0.821_{\pm 0.009}$	0.736	142 / 800
Anchor Points	$\mathbf{0.659_{\pm 0.014}}$	$\mathbf{0.731_{\pm 0.011}}$	$\mathbf{0.762_{\pm 0.010}}$	$0.790_{\pm 0.010}$	$0.818_{\pm 0.009}$	$0.861_{\pm 0.008}$	0.816	55 / –
Semantic Embedding	$0.371_{\pm 0.019}$	$0.460_{\pm 0.016}$	$0.619_{\pm 0.014}$	$0.719_{\pm 0.012}$	$0.800_{\pm 0.009}$	$0.802_{\pm 0.009}$	0.723	99 / –
Acoustic Embedding	$0.557_{\pm 0.015}$	$0.330_{\pm 0.017}$	$0.528_{\pm 0.016}$	$0.719_{\pm 0.011}$	$0.758_{\pm 0.010}$	$0.797_{\pm 0.009}$	0.709	166 / –
Combined Embedding	$0.523_{\pm 0.017}$	$0.680_{\pm 0.013}$	$0.722_{\pm 0.012}$	$\mathbf{0.802_{\pm 0.009}}$	$\mathbf{0.844_{\pm 0.008}}$	$\mathbf{0.879_{\pm 0.007}}$	**0.826**	**49 / 350**

Table 7: **Kendall correlation between subset and full benchmark rankings.** We report N_{80} and N_{90} thresholds (instead of N_{90} and N_{95}) as Kendall's τ is inherently more conservative than Pearson's r and Spearman's ρ, making higher thresholds difficult to achieve.

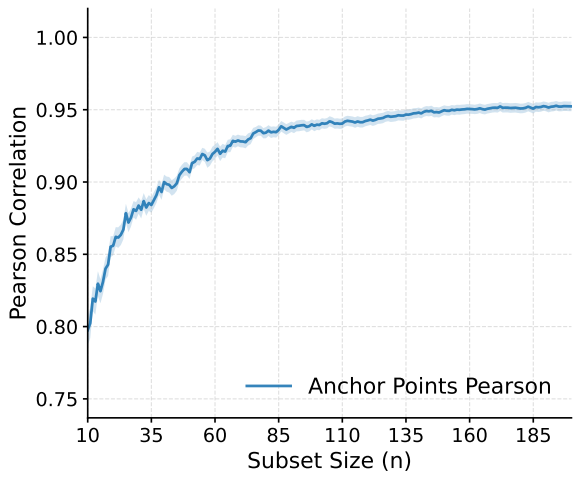

Figure 10: **Anchor Points (Pearson).** AUCC=0.927, $N_{90} = 40$, $N_{95} = 155$.

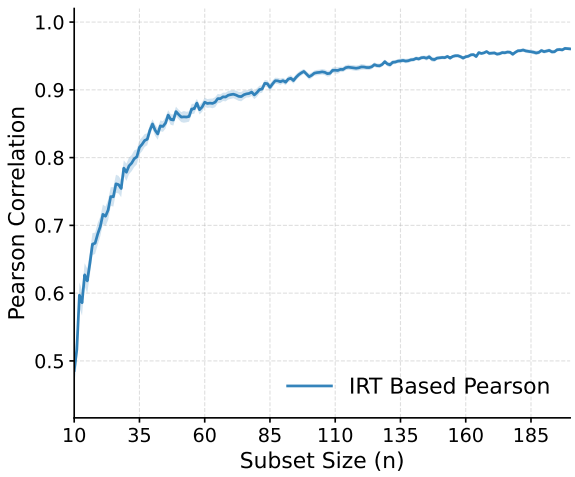

Figure 11: **IRT + Anchor Points (Pearson).** AUCC=0.892, $N_{90} = 81$, $N_{95} = 156$.

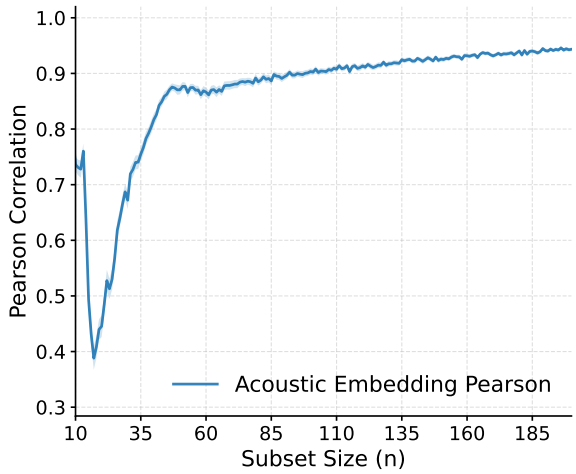

Figure 12: **Acoustic Embedding (Pearson).** AUCC=0.850, $N_{90} = 92$, $N_{95} = 250$.

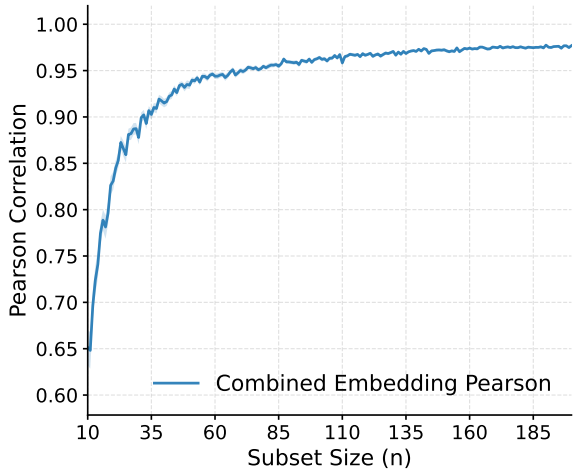

Figure 13: **Combined Embedding (Pearson).** AUCC=0.943, $N_{90} = 32$, $N_{95} = 67$.

E Task Distribution Analysis

To understand what our selection methods prioritize, we analyze the task composition of selected subsets for both Anchor Points (used for n$\leq$30) and Combined Embedding (used for n$\geq$50). Figures 14 and 15 show the percentage distribution of tasks across subset sizes for both methods.

E.1 Selection Patterns Across Subset Sizes

Both methods exhibit a clear hierarchy in how they construct evaluation coverage, progressing from foundational capabilities to refined, multifaceted assessment:

Small Subsets ($n \leq 30$): Foundational Capabilities At minimal subset sizes, both methods heavily prioritize fundamental speech understanding capabilities that form the basis of model performance. In Anchor Points at $n = 10$, basic tasks dominate: Target Speaker ASR (21.6%), Speaker Diarization (18.3%), Text Creation (9.3%), and PoS Estimation (7.9%). Combined Embedding shows similar patterns: SUPERB ASR (13.8%), SUPERB Emotion Recognition (10.0%), PoS Estimation (9.9%), Speaker Diarization (9.6%), Text Creation (9.2%), and Speech TriviaQA (9.0%).

These tasks represent the core building blocks of audio model capabilities—the ability to accurately recognize speech, identify speakers, understand basic linguistic structure, and generate coherent content. A model's performance on these foundational dimensions establishes its baseline competence and determines whether it possesses the prerequisite skills for more sophisticated audio understanding. When evaluation budget is minimal, capturing these fundamental capabilities provides the most essential characterization of what a model can and cannot do.

Large Subsets ($n \geq 100$): Refined and Multifaceted Capabilities As subset size increases, selection shifts toward refined evaluation of paralinguistic and specialized capabilities that reflect more diverse and nuanced aspects of audio understanding. Tasks virtually absent at $n = 10$ gain substantial representation by $n = 100$-200: Human Non-Speech Sound increases from 0.0% to 8.9–9.1% (Anchor Points) and 0.2% to 5.2% (Combined), SpeakBench emerges from 0.0% to 5.2% (Combined), L2 English Accuracy/Fluency Ranking grow from 0.0% to 9.3–10.0% (Anchor Points), and Pronunciation tasks increase modestly.

This shift reflects an expansion in the dimensions along which model capabilities are evaluated. Paralinguistic tasks—understanding prosody, accent, fluency, and non-speech audio—capture sophisticated aspects of audio perception that go beyond literal content understanding. These refined capabilities constitute a model's full competence profile: beyond basic speech recognition and content generation, can it perceive subtle acoustic cues, handle diverse speaker characteristics, and understand audio in its full contextual richness? When evaluation budget allows, incorporating these dimensions provides a more complete picture of model capabilities, revealing strengths and weaknesses across the full spectrum of audio understanding rather than just foundational skills.

E.2 Comparison of Selection Methods

Concentrated vs. Distributed Capability Coverage Combined Embedding (Figure 15) achieves more uniform task distribution than Anchor Points (Figure 14), particularly at small-to-medium sizes. At $n = 10$, Anchor Points concentrates on just ~ 7 tasks, while Combined Embedding distributes across 10+ tasks with $> 5\%$ representation. At $n = 200$, Combined Embedding shows relatively balanced 2–6% across most tasks, while Anchor Points maintains sharper peaks (Human Non Speech Sound 9.1%, L2 English Accuracy Ranking 9.5%). This partly explains their performance differences (Figure 2)—at $n \leq 30$, concentrated coverage of foundational capabilities suffices to characterize model quality for Anchor Points, while at $n \geq 50$, broader capability coverage better approximates the multifaceted nature of comprehensive evaluation for Combined Embedding.

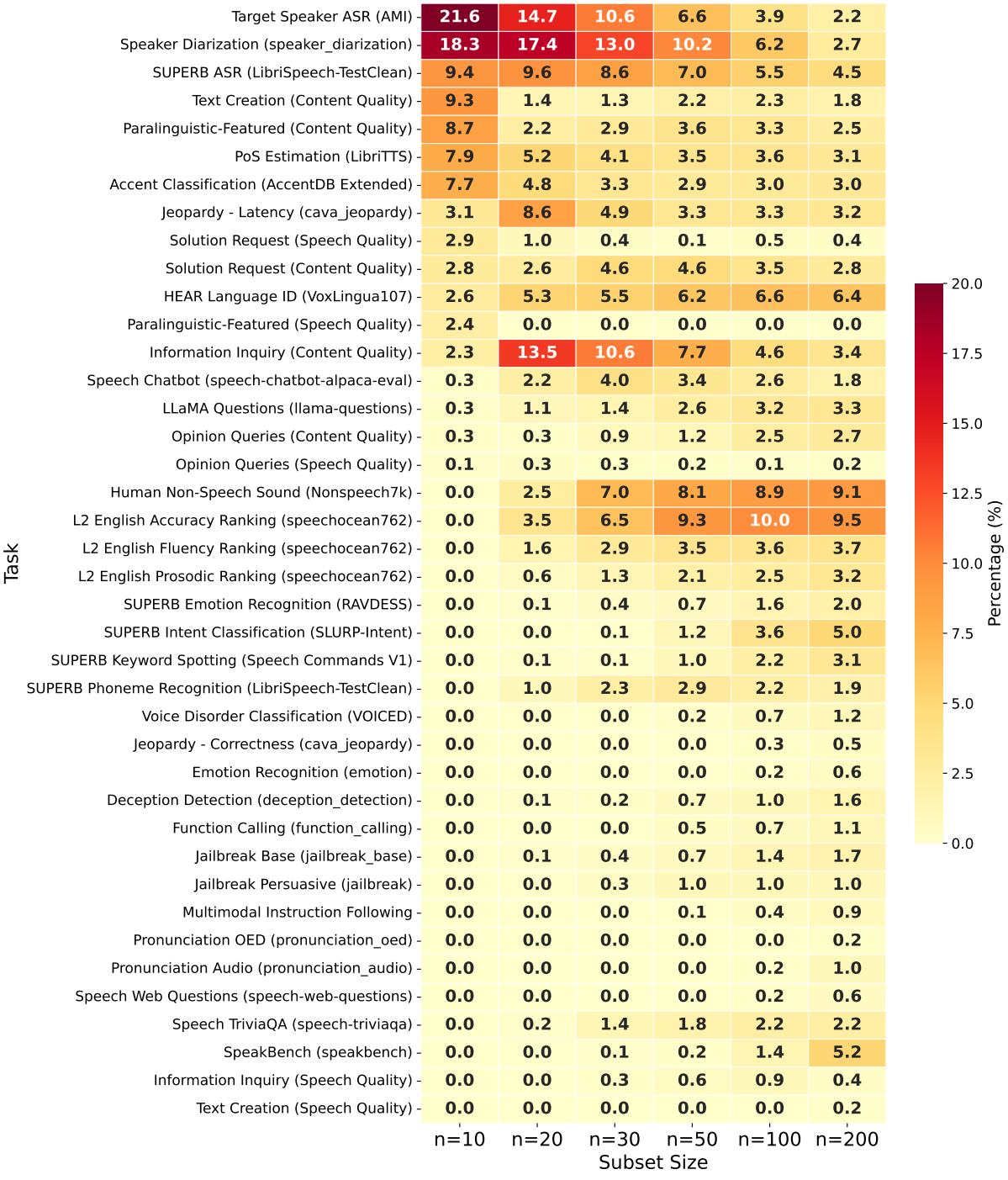

Figure 14: **Task distribution for Anchor Points method.** Heatmap shows the percentage of items from each task in subsets of varying sizes (n=10 to n=200), averaged across 100 random seeds. Darker red indicates higher representation. Tasks are ordered by their representation at n=10.

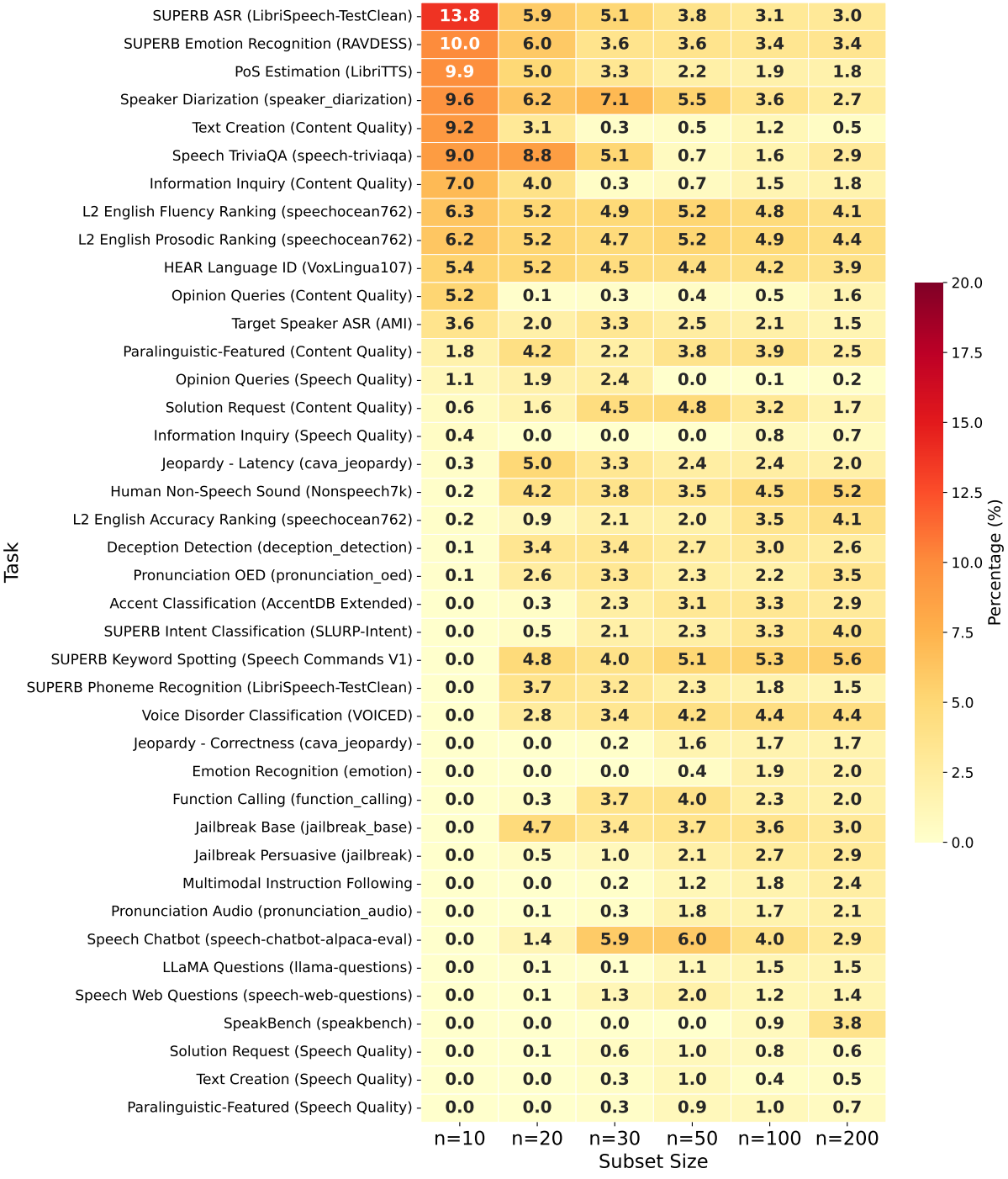

Figure 15: **Task distribution for Combined Embedding method.** Heatmap shows the percentage of items from each task in subsets of varying sizes (n=10 to n=200), averaged across 100 random seeds. Darker red indicates higher representation. Tasks are ordered by their representation at n=10.

F Conversational Agent Framework Implementation Details

F.1 System Architecture

Our conversational agent framework is built on LiveKit Agents 1.0 (LiveKit, Inc., 2024), an open-source WebRTC infrastructure and agent framework that enables real-time audio streaming with low latency and provides high-level abstractions for pipeline voice assistant development. We extend this framework to support three distinct model architectures:

- **End-to-end omni-modal**: Audio-in, audio-out models (e.g., GPT-4o-audio, Qwen2.5-Omni) that natively process and generate speech
- **Speech-to-text**: Audio-in, text-out models that understand audio but generate only text, requiring external TTS
- **Pipeline systems**: STT + text-only model + TTS, augmenting text models with separate speech components

F.2 Audio Processing and Turn Detection

We employ Silero VAD (Silero-Team, 2024) for voice activity detection with the following configuration optimized for conversational interactions:

- `min_speech_duration`: 0.1s (captures short utterances)
- `min_silence_duration`: 2.0s (allows natural pauses)
- `prefix_padding_duration`: 0.2s (pre-speaking buffer)
- `activation_threshold`: 0.4 (balanced sensitivity)
- `sample_rate`: 16000 Hz

All audio is processed at 16kHz sampling rate. The 200ms pre-speaking buffer ensures we capture the beginning of utterances by maintaining a rolling window of audio frames before speech detection triggers. When the user begins speaking, these buffered frames are prepended to the captured audio, preventing cutoff of initial phonemes.

F.3 System Prompts and Conversation Management

All models receive consistent system prompts constructed from scenario-specific instructions plus standardized guidelines:

> *{scenario_prompt}*
>
> You are a voice assistant. *[Architecture-specific instructions]* Respond naturally and conversationally.
>
> **You should never reveal to the user which model you are.** If asked, say you are a voice assistant.

Architecture-specific instructions vary by model type:

- Audio-in/audio-out: "You receive audio input and respond with audio. Speak naturally in English."
- Audio-in/text-out: "You receive audio input and respond with text that will be converted to speech."
- Text-in/text-out: "You receive text transcribed from audio and respond with text that will be converted to speech."

Conversation context for the agent tracks:

- Message roles (user/assistant/system/tool)
- User input audio
- Assistant text output or audio transcription
- Tool calls made by the assistant and their results

F.4 Function Calling and Tool Integration

All models have access to a consistent set of tools regardless of architecture:

- **Web search**: DuckDuckGo API (DuckDuckGo, 2008) integration for retrieving information URLs
- **URL fetching**: Content extraction from web pages
- **Scenario-specific APIs**: Domain-specific tools (travel, shopping, calendar, social media, smart home, filesystem, messaging, job search) dynamically loaded based on evaluation scenario

Tool execution is tracked by a verifier component that monitors function calls and validates scenario goal completion, providing real-time feedback to participants on model progress.

F.5 Real-time State Broadcasting

To enable frontend visualization and interaction monitoring, the system broadcasts state updates via LiveKit's data channel (see Figure 20 in the conversation interface):

- **Agent state changes**: "listening", "thinking", "speaking"
- **User state changes**: "listening", "speaking", "away"
- **Function call execution**: Function name, arguments, results, success/failure
- **Verification status**: Scenario goal completion, function call correctness

G Human Evaluation Protocol and Scenario Design

G.1 IRB Approval and Ethical Oversight

This research was approved by the Institutional Review Board (IRB) at the authors' institution prior to data collection. All procedures followed institutional guidelines for research involving human subjects.

G.2 Participant Recruitment

We recruited native English speakers from the United States via Prolific, a crowdsourcing platform for research participants. Recruitment was limited to participants who were native English speakers, located in the United States, and 18 years of age or older. We balanced recruitment to achieve approximately equal gender representation (50% male, 50% female). A total of 776 participants completed the study.

G.3 Compensation

Participants were compensated $0.25 base payment and $0.25 per minute of conversation ($2.50 for a full 10-minute conversation), with additional bonuses: $1 for successfully completing goal-oriented tasks and $0.25 for providing feedback. This resulted in compensation ranging from $2.75 to $4 for 10-13 minutes of total participation time (including consent, conversation, and rating submission), yielding an hourly rate of at least $15/hour minimum wage rate.

G.4 Informed Consent and Data Usage

Before beginning the study, all participants were presented with a consent form (Figure 16) that first explained the study workflow: assignment to a random conversation scenario and voice assistant, a 10-minute voice conversation, and rating the assistant's performance across multiple dimensions. Following this overview, participants reviewed detailed information about the research purpose, study procedures, voice recording, data usage for research purposes, privacy protections, and their right to withdraw at any time.

The consent form specifically informed participants that:

- Their voices would be recorded during interaction with AI speech models
- All tasks occur in a simulated environment with no real actions or transactions
- No personal information would be recorded
- Study data would be stored securely in compliance with institutional standards
- Risks associated with participation are minimal
- Participation is voluntary and they have the right to withdraw consent at any time without penalty
- Results may be presented at scientific or professional meetings or published in scientific journals, with individual privacy maintained

G.5 Instructions Given to Participants

Upon consenting to participate, Participants then received scenario-specific instructions based on their randomly assigned conversation type:

Open Chat (20% of conversations): Participants were shown a simple instruction screen indicating they would engage in free-form conversation with the AI assistant on any topic of interest for 10 minutes, with no specific goals or constraints (Figure 17).

Goal-Oriented Dialogue (40% of conversations): Participants received a scenario card (Figure 18) containing: a brief scenario title, your goal, situation description, numbered task steps, and clarifying notes.

Function Calling Tasks (40% of conversations): Participants received structured task scenarios (Figure 19) including scenario title, goal, identity if needed, situation, key details, and step-by-step tasks.

G.6 Conversation Interface

During the conversation, participants interacted through a real-time voice interface (Figure 20).

The interface displayed:

- The assigned scenario information in the left panel
- Real-time conversation status indicators showing the status of the agent and user (e.g. listening or speaking)
- A microphone button to control voice input
- Connection status and elapsed time
- For function calling tasks: a right panel showing "Function Verification" with real-time tracking of required function calls and instance state verification progress
- An "End Conversation Early" button for participants who wished to terminate before the 10-minute timer

G.7 Post-Conversation Evaluation

After completing the conversation, participants provided ratings on an evaluation page (Figures 21 and 22). Participants were instructed to use the full 1-6 scale, with guidance that 1-2 indicates significant problems, 3-4 represents typical performance with room for improvement, and 5-6 is reserved for exceptional quality.

Participants rated five dimensions using 6-point Likert scales: (1) Overall Recommendation, (2) Understanding (speech, intent, and paralinguistic cues), (3) Naturalness (conversational flow and conciseness), (4) Response Quality (accuracy, safety, relevance, helpfulness), and (5) Task Effectiveness (efficiency in achieving goals).

Participants then provided required written feedback explaining their ratings and could optionally record audio feedback.

G.8 Scenario Design and Generation

G.8.1 Goal-Oriented Scenario Generation

We created realistic conversation scenarios by adapting real user-chatbot interactions from two complementary text-based datasets: LMSYS-Chat-1M (Zheng et al., 2023), containing diverse Chatbot Arena conversations, and WildChat (Zhao et al., 2024), documenting authentic ChatGPT usage patterns.

Using GPT-4.1 with few-shot prompting and reject sampling, we transformed appropriate conversations into structured scenarios containing: (1) a brief title, (2) 2-3 sentence description providing user context, and (3) 3-5 conversation goals representing logical discussion milestones. Goals serve as conversation helpers to encourage sustained engagement rather than strict requirements. We rejected conversations requiring specialized knowledge, visual aids, external tools, or containing sensitive content. This process yielded 500 candidate scenarios from each dataset (1,000 total).

A second filtering stage using o4-mini validated and improved scenarios for conversational suitability, ensuring appropriateness for general participants in voice-only interactions.

G.8.2 Function Calling Task Scenarios

We adapted the Berkeley Function-Calling Leaderboard (BFCL) v3 (Patil et al., 2025) multi-turn function calling framework to create 40 realistic voice assistant evaluation scenarios spanning 8 domains: calendar management (5 tasks), shopping (5 tasks), travel booking (8 tasks), social media (5 tasks), smart home control (1 task), filesystem operations (7 tasks), messaging (5 tasks), and job search (4 tasks). Each scenario includes: (1) a user goal and situational context, (2) initial system state with pre-populated data, (3) available function definitions, (4) required verifiable actions for successful completion, (5) forbidden operations that constitute violations, and (6) expected final state for verification.

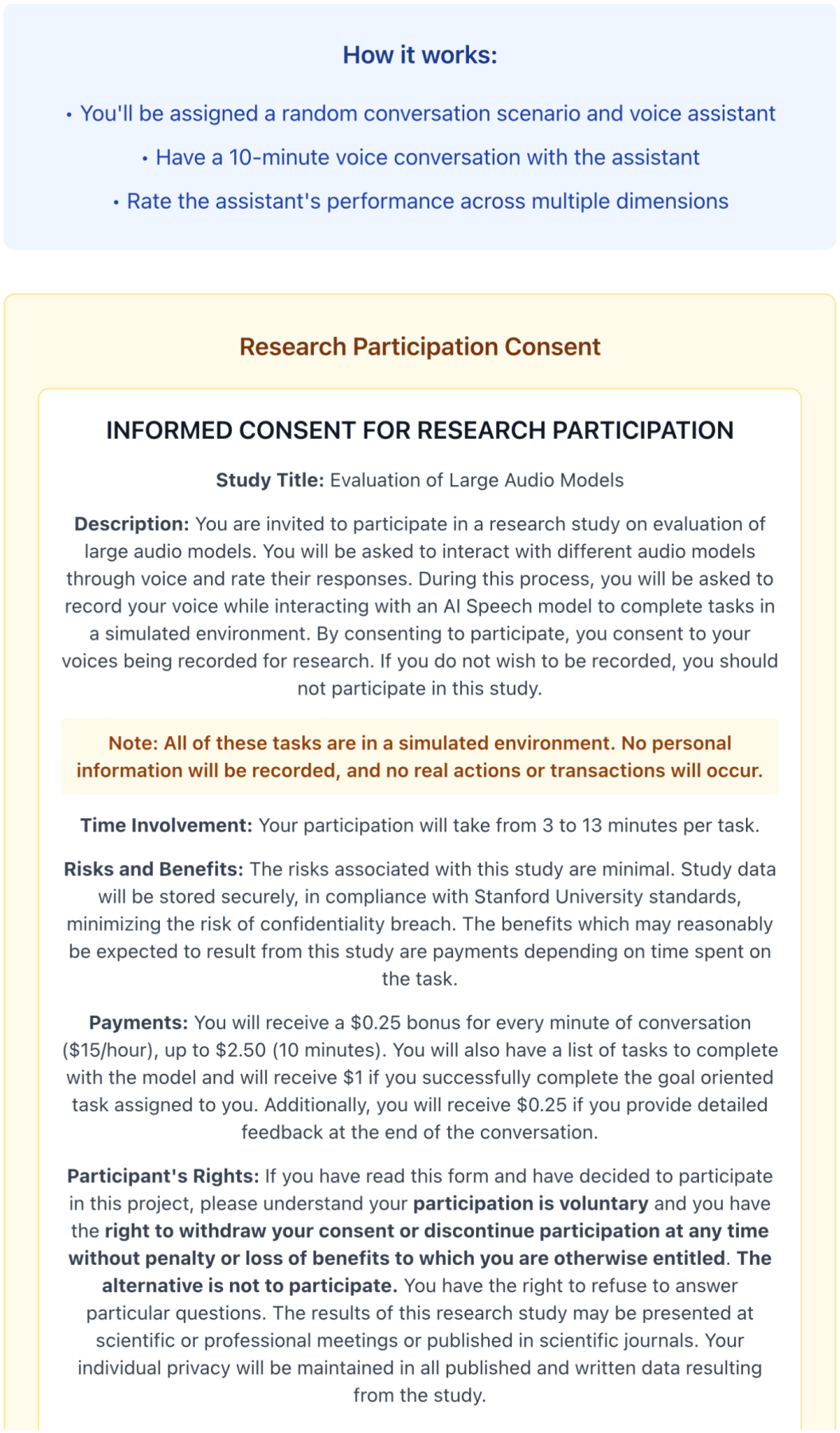

Figure 16: **Study overview and informed consent form.** The page begins with "How it works" explaining the study workflow, followed by the detailed informed consent section covering purpose, procedures, risks, benefits, compensation, and participant rights.

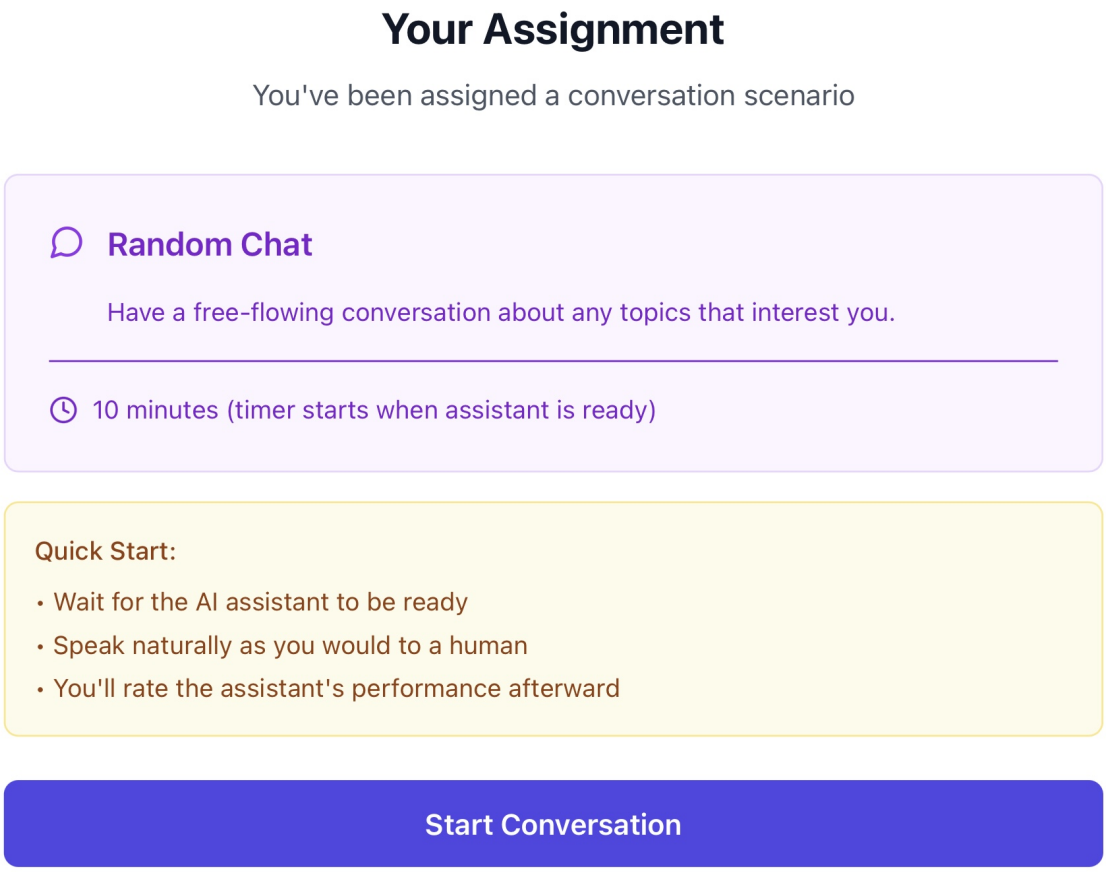

Figure 17: **Open chat scenario assignment.** Example of scenario instructions for free-form conversations.

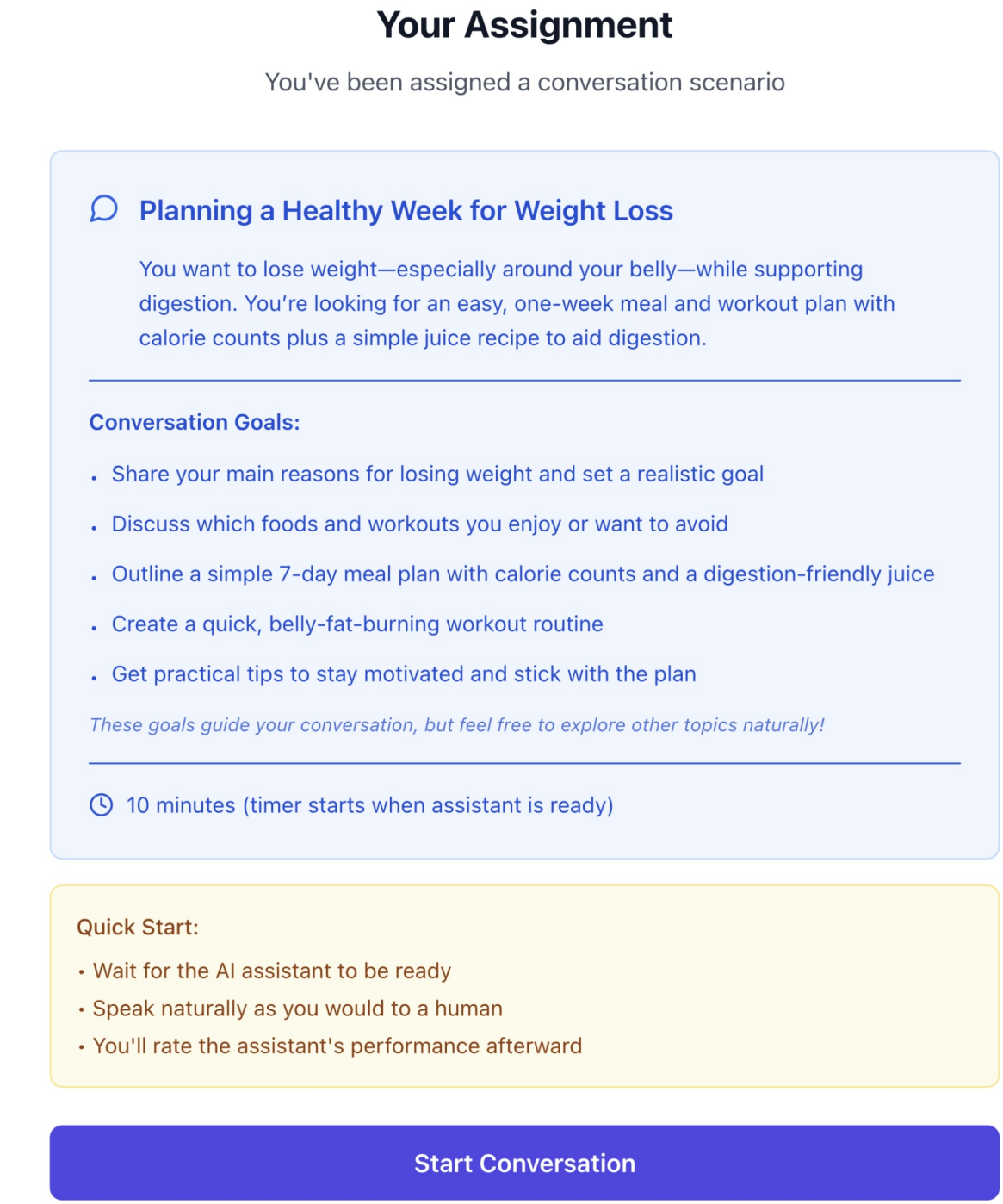

Figure 18: **Goal-oriented scenario assignment.** Example showing the "Weight Loss" scenario with structured context and goals.

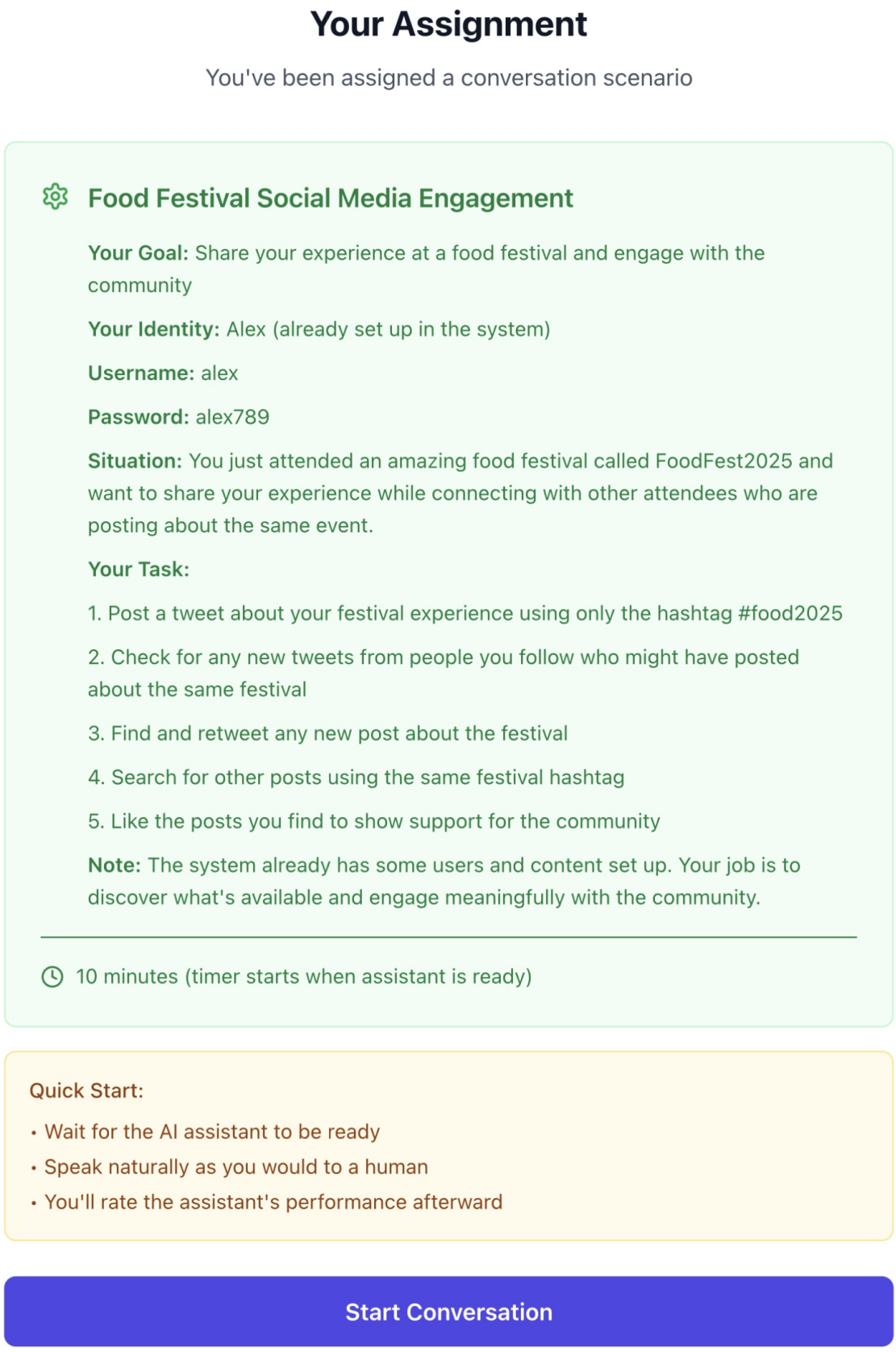

Figure 19: **Function calling scenario assignment.** Example showing the "Social Media Engagement" scenario with detailed task requirements.

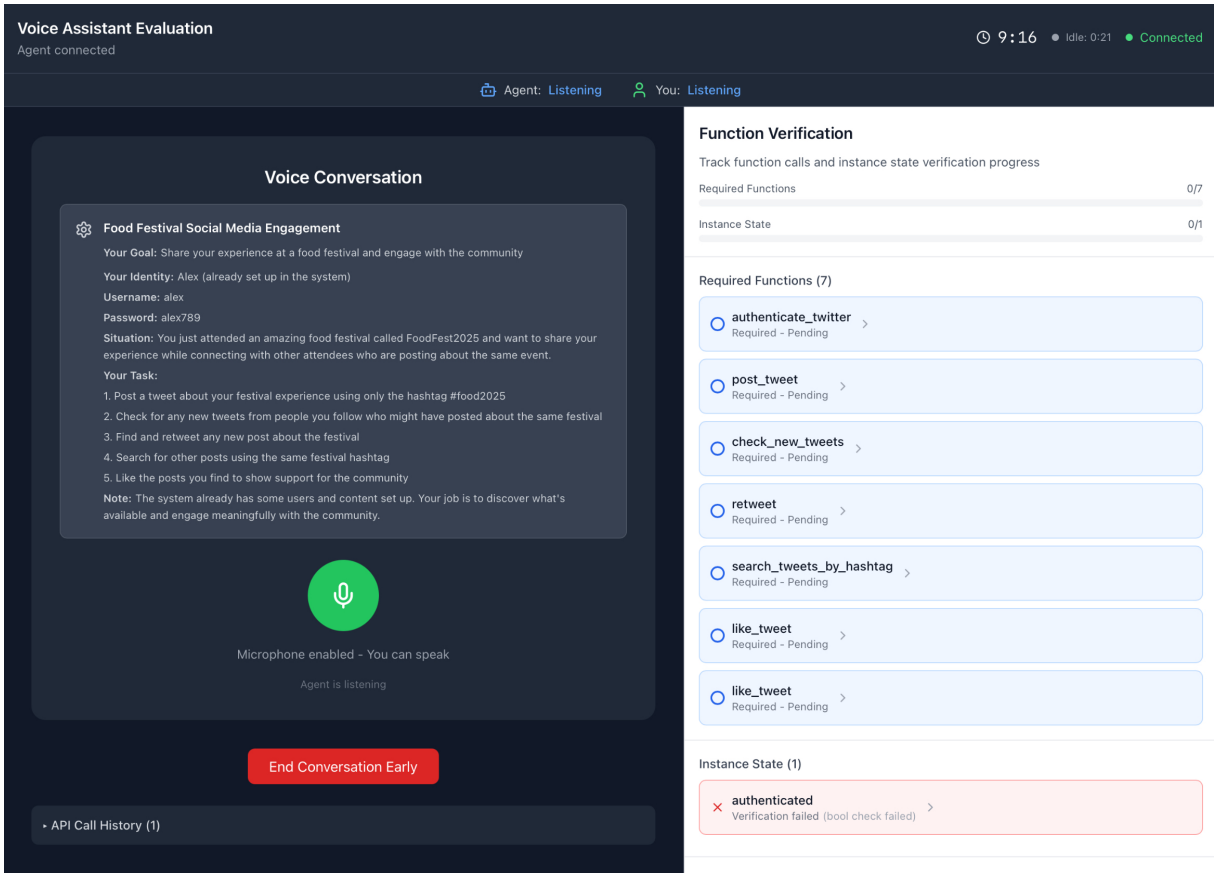

Figure 20: **Real-time conversation interface.** Shows the voice interaction screen with scenario information (left), conversation status indicators, and function verification panel (right) for tracking task completion in function calling scenarios.

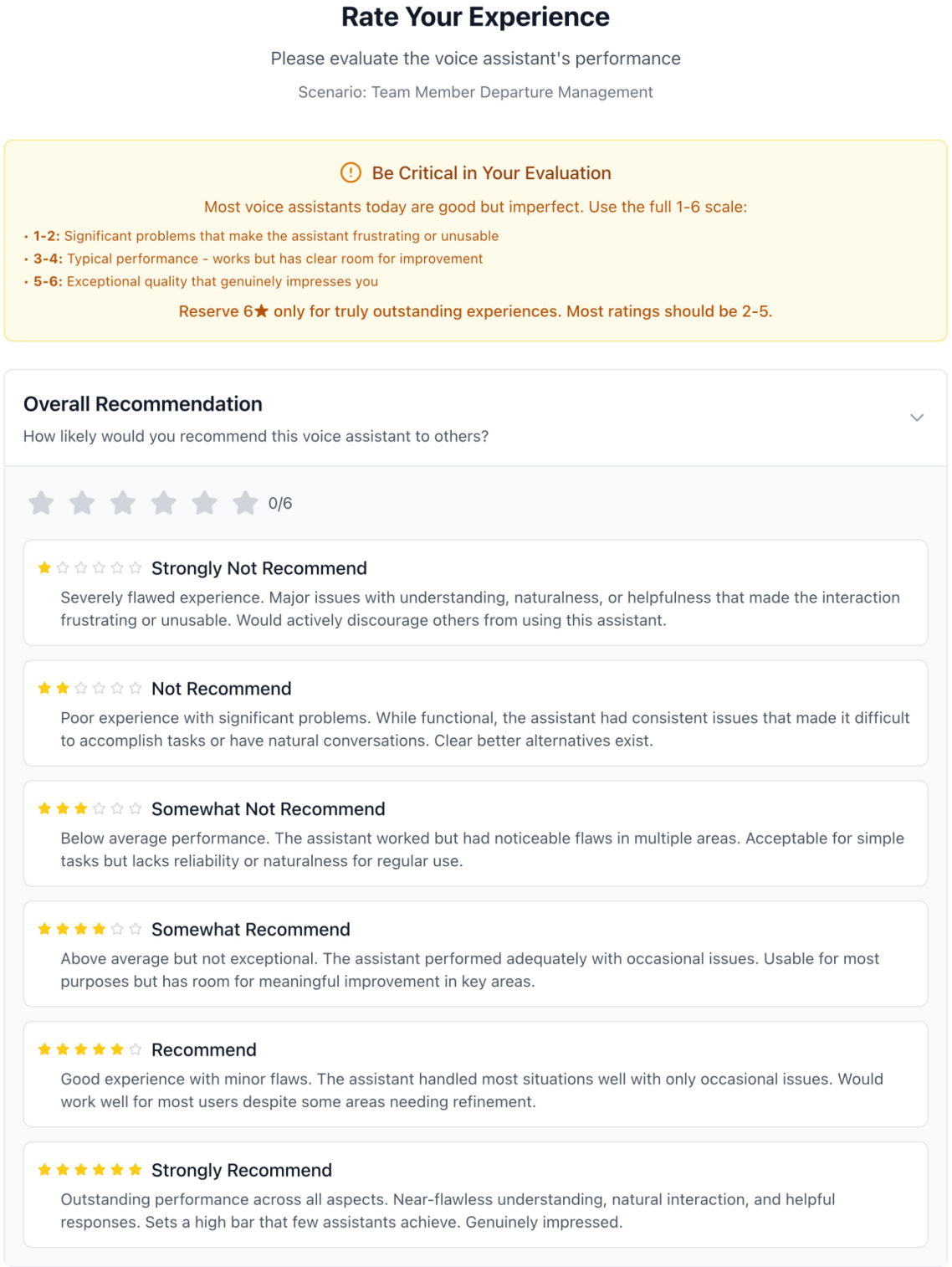

Figure 21: **Overall recommendation rating interface.** Expandable 6-point scale with detailed descriptions for each rating level.

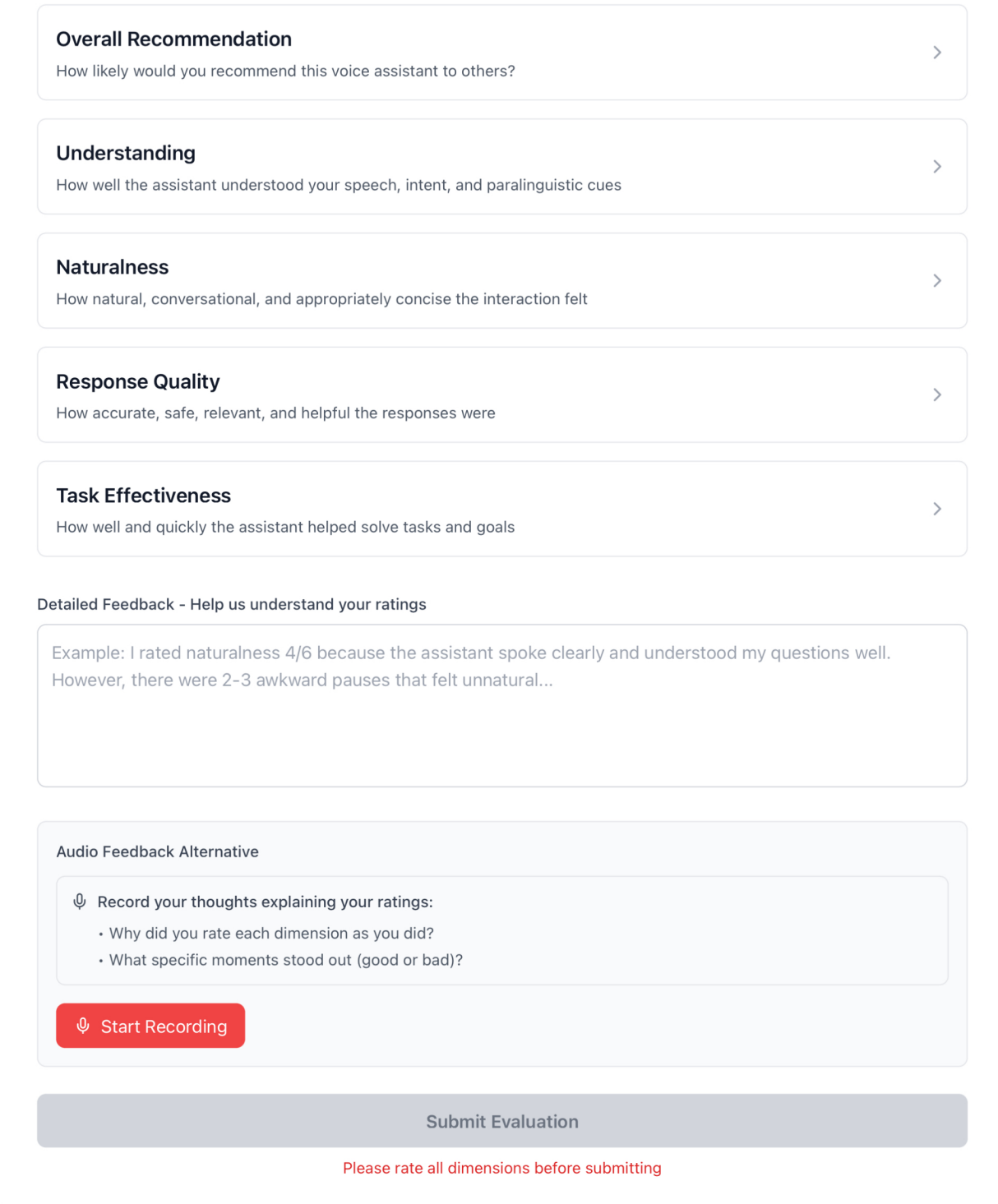

Figure 22: **Multi-dimensional rating interface.** All five evaluation dimensions with expandable scales, text feedback area, and optional audio feedback recording.

H Qualitative Analysis of Human Evaluations

H.1 Failure Modes and User Feedback

To understand model limitations from users' perspective, we analyzed open-ended feedback from all 776 conversations. For participants who provided optional audio feedback, we first transcribed recordings to text using Whisper-large-v3 (Radford et al., 2023) and concatenated with written feedback. To protect participant privacy, all feedback was processed through Microsoft Presidio (Microsoft, 2025) to automatically detect and mask personally identifiable information (PII) including names, addresses, phone numbers, email addresses, and other identifying details. Detected entities were replaced with placeholders before any subsequent analysis or storage.

We then applied a three-stage automated pipeline powered by GPT-5.2 (OpenAI, 2025e) to the privacy-protected feedback. Among the 741 participants who left feedback, 621 expressed some form of dissatisfaction or suggested improvements.

Stage 1: Dissatisfaction Detection and Summarization. We fed each piece of raw feedback to GPT-5.2 to determine whether it expressed dissatisfaction and, if so, to generate a concise 1–2 sentence summary of the specific failure mode.

Stage 2: Category Generation. We provided all 621 dissatisfaction summaries from Stage 1 to GPT-5.2 in a single prompt, instructing it to inductively generate comprehensive failure mode categories that cover the range of problems mentioned. The model generated 25 distinct categories.

Stage 3: Feedback Categorization. Using the 25 categories generated in Stage 2, we classified each dissatisfaction summary by asking GPT-5.2 to assign it to one or more applicable categories, allowing for multi-label classification. We then coded the 621 feedback responses into the resulting categories.

H.1.1 Overall Failure Mode Distribution

Table 8 presents the complete distribution of failure modes. The results reveal clear patterns in user dissatisfaction that are not adequately captured by existing benchmarks.

Conversational Quality Dominates User Dissatisfaction The most striking finding is that conversational quality issues far outweigh technical capability failures. Three related categories dominate user complaints:

Failure Mode	Count (%)
Robotic/Unnatural Speaking Style	266 (42.8%)
Task Execution Failure	147 (23.7%)
Slow Response Latency	143 (23.0%)
Stilted Conversation Flow	117 (18.8%)
Overly Verbose Responses	107 (17.2%)
Unhelpful Response Strategy	95 (15.3%)
Misunderstood User Intent	82 (13.2%)
Looping/Stalling/Non-Responsiveness	70 (11.3%)
Audio Output Glitches	68 (11.0%)
Insufficient Proactivity	57 (9.2%)
Poor Speech Recognition	54 (8.7%)
Poor UX/Interface Support	44 (7.1%)
Missing Confirmation/Progress Transparency	43 (6.9%)
Inconsistent Voice or Language Output	42 (6.8%)
Incomplete Answers	39 (6.3%)
Tool/API/Integration Errors	37 (6.0%)
Incorrect or Unreliable Information	29 (4.7%)
Outdated or Not-Current Information	19 (3.1%)
Tone/Interpersonal Issues	16 (2.6%)
Poor Instruction Following	15 (2.4%)
Over-Agreeable/Lack of Critical Pushback	14 (2.3%)
Context/Memory/State Tracking Issues	12 (1.9%)
Privacy/Security/Auth Friction	11 (1.8%)
Bias/Defensiveness/Trust Issues	5 (0.8%)
Unintended or Premature Actions	4 (0.6%)

Table 8: **Distribution of failure modes from human feedback.** Count and percentage of 621 dissatisfaction cases across 25 categories. Percentages sum to >100% due to multi-label classification where feedback could be assigned to multiple categories.

Robotic/Unnatural Speaking Style (42.8%) emerged as the single most common issue, with users describing outputs as monotone, overly formal, or having unnatural prosody. Representative feedback includes:

- *"The assistant understood the queries and provided sufficient, reasonable answers, but its speech sounded slightly robotic and overly formal—at times like it was reading from a book—reducing naturalness."*
- *"I rated naturalness 4/6 because it felt a little monotone and robotic with little inflection or changes of pace that are typical in normal human conversation."*
- *"I would have liked maybe a little side chatter."*

Stilted Conversation Flow (18.8%) reflects issues with dialogue structure, including excessive numbered lists, scripted delivery, and poor turn-taking:

- *"It was more focused on lists, so I did not get a chance to really have a normal conversation."*
- *"It felt less like a flowing conversation and more like a series of individual responses."*
- *"Overall I recommended her as five the only reason that I gave her a little bit lower score*

of four on the task effectiveness is because she was very cautious and kept repeating herself over and over again."

Overly Verbose Responses (17.2%) captures users' frustration with excessive detail when concise answers were expected:

- *"I think the lengthy response can let people lose focus and forget the task at hand and become distracted by other parts that the AI is talking about."*
- *'Even for simple questions, replies contained too much detail, causing the user to lose attention/tune out despite the information generally being useful."*
- *'I do feel like some of the information could be overwhelming with how much she gives."*

Collectively, these three conversational quality categories appear in 490 mentions. Since feedback can match multiple categories, we find that 352 unique feedback instances (56.7%) mention at least one of these three conversational quality issues, making them the dominant source of user dissatisfaction. While some benchmarks include speech quality metrics (e.g., UTMOS scores in WildSpeech-Bench for speech quality), these measures capture only isolated utterance quality rather than conversational dynamics like verbosity, formality appropriateness, or turn-taking. This gap between static quality assessment and interactive conversational experience helps explain why our full benchmark achieves 0.851 correlation with human overall satisfaction—benchmarks primarily optimize for correctness and isolated speech quality, while users evaluate holistic conversational experience including response length, style appropriateness, and dialogue flow.

Task Execution and Technical Reliability Beyond conversational quality, task execution failure (23.7%) and technical issues significantly impact user experience. Task execution complaints typically involved incomplete function calls or failed tool integrations:

- *"It kept saying something about there was errors on the calendar that I had to manually fix. It wouldn't try to solve them. And overall, it wouldn't book the meeting at the end."*
- *"I felt that the AI assistant was really helpful in what I was trying to do. Unfortunately, she wasn't able to help me find a cheap flight on the day that I wanted to take my trip. She asked if I wanted to change the date or add more money to my card, which I chose not to do."*

Latency and Infrastructure Limitations Slow response latency (23.0%) and audio output glitches (11.0%) represent infrastructure limitations rather than core model capabilities. Notably, feedback in these categories typically expressed mild frustration rather than severe dissatisfaction:

- *'There were a few moments that the assistant took longer than expected to reply to my responses. Other than that, the experience was quite pleasant."*
- *"There was some stuttering at times, but other than that, it was very good."*

These issues highlight the need for real-time processing capabilities, such as streaming audio input processing that begins before users finish speaking. While proprietary APIs like GPT-Realtime API (OpenAI, 2025d) and Gemini Live API (Google DeepMind, 2024) offer such capabilities, they employ model-specific optimizations that conflate infrastructure improvements with core model capabilities, which would not reflect what our benchmarking is trying to optimize for. We did not build a more real-time system that processes audio while speaking either, as this requires engineering changes to the model's internal structure and would not support closed-source models in our evaluation framework. Our standardized processing approach prioritizes isolating model performance from deployment optimizations, though this comes at the cost of increased latency. Importantly, we include latency measurements in our 40 benchmark tasks, so latency impact can be quantified in our regression models.

Speech Recognition Less Problematic Than Expected Notably, poor speech recognition accounts for only 8.7% of complaints. This suggests contemporary LAMs have largely solved basic ASR for native English speakers in controlled conditions. The remaining challenges primarily involve edge cases (accents, background noise, domain-specific terminology) or downstream intent interpretation (13.2%) rather than raw transcription accuracy. This finding aligns with our observation in Table 2 that Understanding scores consistently exceed Overall Satisfaction across all

models—speech comprehension is generally not the limiting factor in user experience.

Implications for Benchmark Design Our qualitative analysis reveals three critical gaps in current LAM evaluation:

1. **Missing conversational quality metrics:** Naturalness, conciseness, and appropriate formality drive 78.9% of user dissatisfaction yet are not systematically evaluated in existing benchmarks.
2. **Static evaluation misses interactive failures:** Latency, turn-taking, error recovery, and real-time audio quality only manifest in live conversation, not in offline benchmark tasks.
3. **Accuracy-usability tradeoff unaddressed:** Benchmarks prioritize correctness (ASR word error rate, task completion) while users weight naturalness and efficiency equally or higher in determining overall satisfaction.

These findings justify our human preference validation approach: benchmark subset selection must be validated against user experience to ensure selected items capture not just the capabilities measured by benchmarks, but the dimensions users actually care about. Our regression models (Section 4.4) partially address this gap by learning to weight benchmark items according to their correlation with human satisfaction, effectively discovering which benchmark tasks serve as proxies for conversational quality that benchmarks do not directly measure.

H.1.2 Model-Specific Failure Patterns

While Section H.1.1 identified overall trends, individual models exhibit distinct failure profiles reflecting architectural choices and optimization priorities. Figure 23 visualizes failure mode distributions across all 7 evaluated models for the most prevalent categories (>8% overall rate).

TTS Quality Degrades Naturalness in Pipeline Systems Models using external TTS (GPT-4o-mini+STT+TTS, Voxtral+TTS, Ultravox+TTS) show elevated robotic style complaints. GPT-4o-mini with full pipeline exhibits the highest rate at 50.6% (1.18× baseline), while Voxtral (45.6%, 1.07×) and Ultravox (44.9%, 1.05×) also exceed the 42.8% average. In contrast, GPT-4o-audio (39.8%, 0.93×) and GPT-4o-mini-audio (40.2%, 0.94×)—which use native generation or higher-quality TTS—perform closer to baseline. This pattern confirms that pipeline architectures incorporating separate TTS components degrade conversational naturalness, with the full STT→LLM→TTS pipeline showing the most severe impact.

Closed-Source Large Models Trade Latency for Quality Proprietary large models exhibit disproportionately high latency issues: GPT-4o-audio (38.6%, 1.68× baseline), Gemini-2.5-Flash (28.7%, 1.25×), and GPT-4o-mini-audio (23.7%, 1.03×) all exceed the 23.0% average, while open-source models like Ultravox (15.3%, 0.67×) and Qwen3-Omni (16.5%, 0.72×) show lower rates. This suggests closed-source providers prioritize response quality over speed, which introduce user-perceptible delays. Interestingly, users appear willing to tolerate this tradeoff—GPT-4o-audio achieves the highest satisfaction (4.98) despite the highest latency complaints, indicating quality outweighs responsiveness for overall experience.

Large Open-Source Models Struggle with Conciseness Qwen3-Omni (27.1%, 1.58× baseline) and Voxtral (23.3%, 1.36×) show the highest rates of overly verbose responses, substantially exceeding the 17.2% average. These models also exhibit elevated stilted conversation flow issues (Qwen3: 22.4%, 1.19×; Voxtral: 23.3%, 1.24×), suggesting systematic problems with conversational brevity and natural dialogue structure. In contrast, GPT-4o-mini-audio achieves remarkably low verbosity (8.2%, 0.48×), indicating successful optimization for concise interaction. This verbosity gap likely stems from instruction-tuning differences—open-source models may be trained to provide comprehensive explanations while proprietary voice assistants are optimized for minimal, conversational responses.

Ultravox Audio Understanding Limitations Ultravox-v0.4-ToolACE-8B still exhibits severe audio comprehension issues: poor speech recognition (17.3%, 1.99× baseline) and misunderstood user intent (32.7%, 2.48×) are both approximately double the average rates. These understanding failures compound with task execution problems (39.8%, 1.68×) and unhelpful response strategies (30.6%, 2.00×), creating a cascading failure pattern that explains the model's lowest overall satisfaction (3.34). Notably, Ultravox's benchmark scores (0.384) correctly predict poor performance, but the failure mode analysis reveals the specific bottleneck: audio input processing rather than output capabilities.

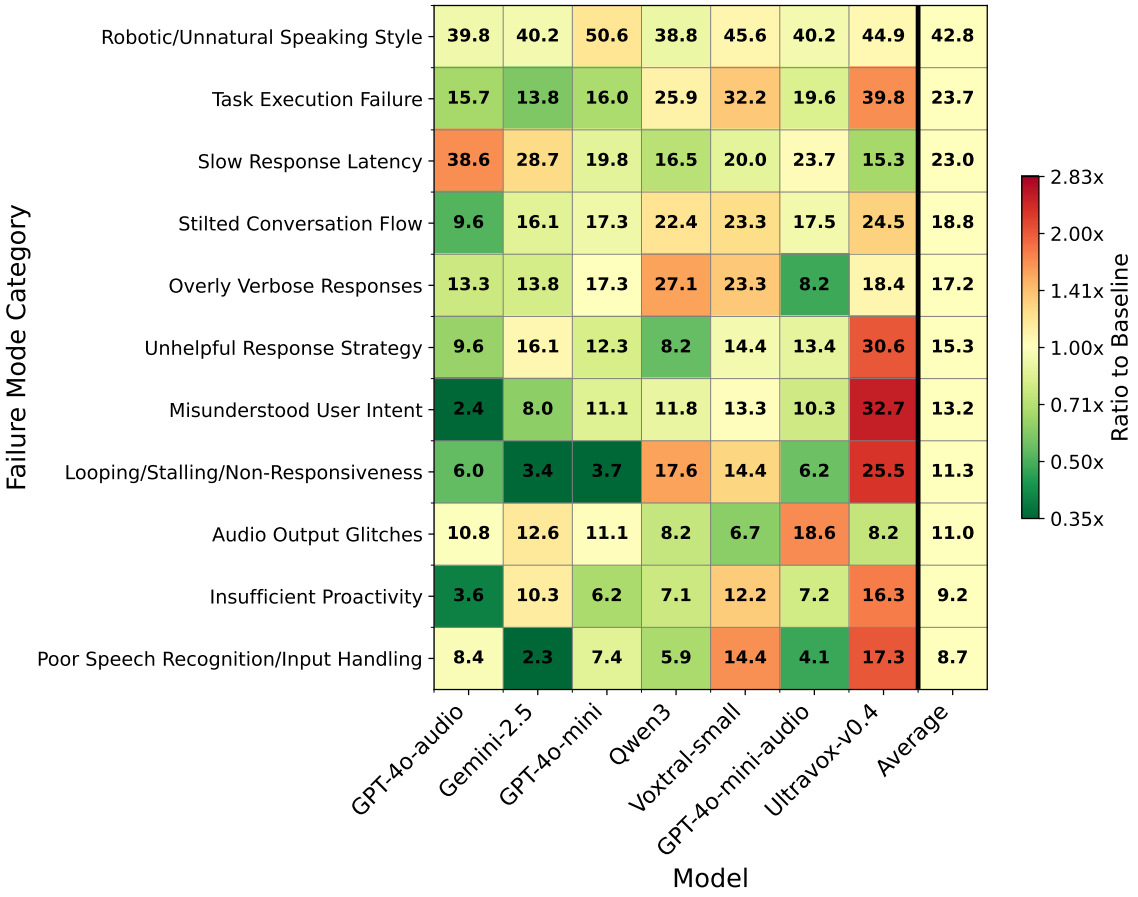

Figure 23: **Model-specific failure mode distributions.** Heatmap shows percentage of dissatisfaction cases mentioning each failure category for each model. Cell color intensity represents the ratio to baseline (average across all models), with darker red indicating higher-than-average rates and darker green indicating lower-than-average rates. Models are ordered by overall human satisfaction (left to right: highest to lowest). Only failure modes with >8% overall prevalence are shown.

Native Audio Output Does Not Guarantee Quality GPT-4o-mini-audio shows 18.6% audio output glitches (1.69× baseline) and 30.9% inconsistent voice output (4.54× the 6.8% baseline, not shown in figure)—dramatically higher than models using external TTS like GPT-4o-mini+TTS (8.2% glitches, 6.2% inconsistency). This counterintuitive finding suggests that **small-scale native audio generation does not necessarily provide advantages over high-quality TTS synthesis**. While end-to-end architectures theoretically enable better prosody control and voice consistency, gpt-4o-mini-audio appear to lack the capacity for stable audio generation, producing artifacts, dropouts, and voice switching issues. This explains why GPT-4o-mini-audio (3.69 satisfaction) substantially underperforms the pipeline-based GPT-4o-mini+STT+TTS (4.51), despite the latter's higher robotic style complaints—reliability trumps naturalness when audio output actively fails.

These findings demonstrate that failure modes are not uniformly distributed—architectural choices, model scale, and optimization priorities create distinct profiles that benchmarks alone cannot reveal. Understanding these patterns enables practitioners to select models aligned with deployment constraints: prioritize large end-to-end models for conversational applications, accept pipeline systems when leveraging existing text-based capabilities, and recognize that small-scale native audio generation currently offers limited advantages over high-quality TTS synthesis.

H.2 Rating Dimension Correlation Analysis

To understand the relationships between different aspects of user satisfaction, we analyzed pairwise correlations between rating dimensions at both the conversation level (N=776 individual conversations) and model level (N=7 models). Table 9 presents the complete correlation matrix. Model-

	Overall	Understanding	Naturalness	Quality	Effectiveness
Overall	—	0.669	0.626	0.773	0.781
Understanding	0.949	—	0.523	0.679	0.658
Naturalness	0.970	0.957	—	0.586	0.528
Quality	0.978	0.990	0.969	—	0.793
Effectiveness	0.957	0.992	0.946	0.986	—

Table 9: **Pairwise correlations between rating dimensions.** Upper triangle: sample-level correlations across 776 conversations. Lower triangle: model-level correlations across 7 models.

level correlations are uniformly high (all r $>$ 0.94) due to the limited sample size (N=7 models) and the fact that aggregation removes individual conversation variance. Consequently, we focus our analysis on sample-level correlations, which provide more nuanced insights into how different aspects of model performance relate to user satisfaction across 776 individual interactions.

Response Quality and Task Effectiveness Drive Overall Satisfaction: At the sample level, Response Quality (r=0.773) and Task Effectiveness (r=0.781) exhibit the strongest correlations with Overall Satisfaction. These two dimensions are also highly interdependent (r=0.793), indicating they capture related aspects of model utility: when a model provides high-quality responses, it tends to complete tasks effectively, and vice versa. Together, these functional capabilities are the primary drivers of user satisfaction in voice assistant interactions and are therefore predictive for the overall rating.

Naturalness Shows Weak Interdependence and Limited Impact: In contrast, Naturalness demonstrates notably weaker correlations with other dimensions. Its correlation with Task Effectiveness (r=0.528) and Understanding (r=0.523) are the lowest pairwise correlations in the matrix, suggesting Naturalness represents a relatively independent aspect of conversational quality. Moreover, Naturalness exhibits the weakest correlation with Overall Satisfaction (r=0.626) among all four specific dimensions, indicating that conversational flow and naturalness, while measurable, contribute less to overall user satisfaction than functional capabilities. This pattern suggests that users prioritize task completion and response quality over conversational naturalness—a model can feel somewhat robotic yet still achieve high satisfaction if it delivers accurate, effective results.

Understanding Shows High Variance but Limited Discriminative Power: Understanding demonstrates reasonable sample-level correlation with Overall Satisfaction (r=0.669), but notably shows the lowest model-level correlation among all dimensions (r=0.949 vs. r=0.970 for Naturalness). This discrepancy likely stems from ceiling effects: as shown in Table 2, Understanding scores are consistently high across all models (range: 4.036-5.368), with even the lowest-performing model exceeding 4.0 on the 6-point scale. This restricted range at the model level reduces discriminative power, making Understanding less useful for distinguishing between models despite its reasonable within-model variance across individual conversations.

H.3 Verifiable Task Completion and User Satisfaction

For the 40% of conversations involving function calling tasks, we analyzed objectively verifiable task progress in terms of steps alongside subjective user satisfaction metrics. Table 10 presents the results across all evaluated models.

On function-calling tasks, GPT-4o-audio-preview achieved the highest objective task progress at 67.7%, followed by GPT-4o-mini+STT+TTS at 64.1% and Gemini-2.5-Flash+TTS at 61.2%. Smaller open-source models showed substantially lower performance, with Ultravox-v0.4-ToolACE-8B+TTS completing only 30.1% of verifiable task steps. The ranking on function-calling scenarios largely mirrors the overall results in Table 2, with GPT-4o-audio-preview maintaining its leading position and the performance gap between commercial and open-source models remaining substantial.

However, objective task progress correlates only moderately with overall satisfaction ($r = 0.87$) and task effectiveness ($r = 0.85$), with notable divergences revealing the complexity of user satisfaction. Most strikingly, GPT-4o-mini-audio-preview achieves 55.7% task progress compared to Qwen3-Omni-30B+TTS's 36.6%—a substantial differ-

Model	Objective Task Progress	Task Effectiveness	Overall Satisfaction	N
GPT-4o-audio-preview	$67.7\% \pm 5.6\%$	4.77 ± 0.20	4.80 ± 0.16	44
GPT-4o-mini+STT+TTS	$64.1\% \pm 5.6\%$	4.64 ± 0.21	4.27 ± 0.21	45
Gemini-2.5-Flash+TTS	$61.2\% \pm 6.3\%$	4.34 ± 0.22	4.43 ± 0.22	35
GPT-4o-mini-audio-preview	$55.7\% \pm 5.6\%$	3.80 ± 0.28	3.52 ± 0.24	44
Voxtral-Small-24B+TTS	$44.0\% \pm 6.8\%$	2.91 ± 0.24	3.03 ± 0.25	35
Qwen3-Omni-30B+TTS	$36.6\% \pm 5.9\%$	3.93 ± 0.26	3.73 ± 0.25	45
Ultravox-v0.4-ToolACE-8B+TTS	$30.1\% \pm 5.9\%$	2.58 ± 0.29	2.50 ± 0.28	36

Table 10: **Function calling performance.** Objective Task Progress shows percentage of verifiable task steps completed ± standard error; Task Effectiveness and Overall Satisfaction show mean user ratings ± standard error on function-calling scenarios only (6-point Likert scale, higher is better). N indicates the number of function-calling evaluations per model.

ence—yet receives lower overall satisfaction (3.52 vs. 3.73). Similarly, GPT-4o-mini+STT+TTS completes 64.1% of task steps but receives lower overall satisfaction (4.27) than Gemini-2.5-Flash+TTS at 61.2% progress and 4.43 satisfaction. These discrepancies reveal that user satisfaction depends not only on task completion but also on interaction quality—conversational naturalness, responsiveness, and perceived effort—which can vary substantially across models even at similar completion rates. This underscores the limitation of benchmarks that measure task success in isolation without capturing holistic user experience.

I Fair Comparison: Pairwise Ranking Accuracy

While the LOMO Pearson correlation in Section 4.4 provides fine-grained comparison across subset selection methods, it includes in-sample predictions for training models, which may overestimate generalization performance. To provide a fair comparison between regression-based predictions and original subset scores, we evaluate pairwise ranking accuracy using a 5-2 train-test split.

I.1 Evaluation Protocol

We perform exhaustive 5-2 cross-validation across all possible splits of the 7 human-evaluated models:

1. For each of the $\binom{7}{2} = 21$ possible held-out pairs (m_i, m_j):
   - Train Ridge regression on the remaining 5 models' subset scores and human ratings
   - Select regularization strength $\alpha \in \{10^{-4}, 10^{-3}, \ldots, 10^{4}\}$ via nested leave-one-out CV on the 5 training models
   - Retrain on all 5 models with the selected α and predict for the 2 held-out models: $\hat{y}_{m_i}, \hat{y}_{m_j}$
   - Check if the pairwise ranking is correct: $(\hat{y}_{m_i} > \hat{y}_{m_j}) \iff (y_{m_i} > y_{m_j})$
2. Compute **pairwise ranking accuracy**: proportion of correctly ranked pairs across all 21 splits

For comparison, we compute pairwise ranking accuracy using original subset scores on the same 21 held-out pairs without any regression training. This provides a fair evaluation where both approaches make predictions on truly unseen models.

I.2 Results

Table 11 presents pairwise ranking accuracy under fair 5-2 cross-validation. Key findings:

- **Regression improves generalization for larger subsets**: For $n \geq 30$, Ridge regression consistently outperforms original scores for best subsets. The advantage is most pronounced at $n = 50$ and $n = 100$ ($\Delta \approx 0.055$). This suggests that regression effectively predicts human preferences by learning how to weight different benchmark dimensions, capturing the compositional nature of user satisfaction better.
- **Small subsets show comparable performance**: At $n \leq 20$, regression and original scores achieve similar accuracy, likely because small subsets lack coverage of all important dimensions needed for regression to learn robust mappings. Additionally, the 5-2 split constraint limits training to only 5 source models. Our final released benchmarks use all 7 models for training, which should yield better human preference prediction.
- **Quality items are essential for effective regression**: Random sampling shows a striking pattern—regression *underperforms* original scores across all sizes (e.g., $\Delta = -0.080$ at $n = 10$, $\Delta = -0.046$ at $n = 200$). This demonstrates

n	"Best" Subset		Random Baseline	
	Regression	Original	Regression	Original
10	0.797 ± 0.008	0.809 ± 0.004	0.625 ± 0.017	0.705 ± 0.012
20	0.817 ± 0.007	0.807 ± 0.004	0.666 ± 0.015	0.751 ± 0.011
30	0.833 ± 0.007	0.816 ± 0.003	0.676 ± 0.014	0.774 ± 0.009
50	0.882 ± 0.004	0.825 ± 0.004	0.715 ± 0.013	0.802 ± 0.008
100	0.883 ± 0.003	0.828 ± 0.004	0.757 ± 0.011	0.830 ± 0.006
200	0.878 ± 0.003	0.850 ± 0.004	0.800 ± 0.008	0.846 ± 0.005
Full	0.857	0.810	–	–

Table 11: **Fair pairwise ranking accuracy comparison.** Proportion of correctly ranked model pairs using 5-2 cross-validation (mean $\pm$ standard error over 100 random seeds). Both regression predictions and original subset scores are evaluated on the same 21 held-out pairs. "Best" Subset uses Anchor Points for $n \leq 30$ and Combined Embedding for $n \geq 50$.

that regression amplifies the signal from high-quality items but also amplifies noise from uninformative ones. Without principled selection, adding regression to random items degrades performance by overfitting to spurious patterns.

- **Quality over quantity**: Consistent with findings in Section 4.4, regression performance peaks at $n = 100$ (0.883) for best subsets before dropping to 0.878 at $n = 200$ and 0.857 for the full benchmark. This confirms that adding more items introduces lower-informative examples that perturb regression weights.

These results validate our main findings: Ridge regression generalizes better to unseen models when applied to high-quality, diverse subsets, but requires principled item selection to avoid amplifying noise from uninformative examples.

J Licenses

We list the licenses for artifacts involved in this work as follows:

Models:

- **GPT-4o-audio-preview**, **GPT-4o-mini-audio-preview**, **GPT-4o-mini**, **GPT-5**, **GPT-realtime**, **GPT-4o-transcribe**, **GPT-4o-mini-tts**, **GPT-4.1**, **GPT-5.2**: Proprietary models. Usage governed by *OpenAI Terms of Service*. https://openai.com/policies/
- **Gemini-2.5-Pro**, **Gemini-2.5-Flash**: Proprietary models. Usage governed by *Google Generative AI Terms of Service*. https://ai.google.dev/gemini-api/terms
- **Qwen2.5-Omni-7B**: *Apache License 2.0*.
- **Qwen3-Omni-30B-A3B-Instruct**: *Apache License 2.0*
- **Ultravox-v0.4-ToolACE-8B**, **Ultravox-v0.5-llama-3.2-1B**, **Ultravox-v0.6-llama-3.1-8b**: *MIT License*.
- **Llama-3.2-3B**: *Llama 3.2 Community License Agreement*. https://www.llama.com/llama3_2/license/
- **Voxtral-Small-24B-2507**, **Voxtral-Mini-3B-2507**: *Apache License 2.0*
- **Granite-speech-3.3-8b**: *Apache License 2.0*.
- **Gemma-3n-e4b**, **Gemma-3n-e2b**: *Gemma Terms of Use*. https://ai.google.dev/gemma/terms

Benchmarks:

- **Dynamic-SUPERB Phase-2**: Individual datasets within the benchmark have varying licenses. Complete license information available at https://github.com/dynamic-superb/dynamic-superb/blob/main/docs/dataset_license.md
- **CAVA**: *CC BY-SA 4.0* © 2024 Talk Arena. https://github.com/SALT-NLP/CAVA
- **UltraEval-Audio**: *Apache License 2.0*.
- **SpeakBench** (AudioJudge): *MIT License*.
- **WildSpeech-Bench**: *Creative Commons Attribution 4.0 International (CC BY 4.0)*.

K Intended Use and Compliance

All artifacts used in this work are employed consistent with their intended purposes as specified by their creators. Existing benchmarks (Dynamic-SUPERB, CAVA, UltraEval-Audio, SpeakBench, WildSpeech-Bench), datasets (LMSYS-Chat-1M, WildChat, BFCL v3), and pre-trained models are used within their documented scope for research evaluation of audio model capabilities. Speech processing components (GPT-4o-transcribe, GPT-4o-mini-tts, Silero VAD, WavLM-Large, Whisper-large-v3) are employed for their intended audio processing purposes.

Our released HUMANS benchmark and human preference dataset are intended solely for research purposes in audio model evaluation and meta-analysis. The human preference dataset is derived from research participants who consented to academic research use, and any derivative use must comply with these original access conditions and privacy protections outlined in Section 6.

L Package Details

We implement our experiments using Python 3.12 with the following key packages:

- PyTorch 2.8.0 with torchaudio 2.8.0 for model inference
- Transformers 4.51.3 (Hugging Face) for model loading and inference
- NumPy 2.2.6 for numerical computations
- Scikit-learn 1.7.0 for Ridge regression, PCA, and K-Means clustering
- SciPy 1.16.0 for statistical computations
- Datasets 3.6.0 (Hugging Face) for dataset loading and processing
- LiveKit 1.0.11 and livekit-agents 1.1.4 for real-time conversational agent infrastructure
- vLLM 0.10.2 for model deployment

All experiments were conducted on NVIDIA A6000 GPUs (48GB VRAM) for open-source model deployments.