Dimension-First Evaluation of Voice Assistants: Human Chain-of-Thought and Structured Judges

Anonymous ACL submission

Abstract

Voice assistants need to be evaluated on *content* (C), *voice quality* (VQ), and *paralinguistics* (EQ). Most benchmarks collapse these dimensions into a single overall preference. We show this creates artifacts (forced winners and untyped ties) that overweight textual content and make overall accuracy "hackable" by transcript-only judges. To address this, we re-annotate two public datasets using a Human Chain-of-Thought (HCoT) protocol that elicits *dimension-first* judgments with *typed ties* (*both-good* vs. *both-bad*). This yields reliable, diagnostic labels validated under a protocol aligned with ITU-T guidance on separate dimension ratings. Building on HCoT, we propose *JSON-Judge*, a training-free, two-stage evaluator: (i) assemble a compact JSON blueprint of inexpensive, off-the-shelf signals for C/VQ/EQ; (ii) prompt an LLM for per-dimension decisions, which a deterministic rule maps to the overall label. Across SpeakBench and S2S-Arena, JSON-Judge attains higher overall agreement with HCoT than transcript-only and audio-LLM judges and delivers stronger EQ fidelity, while being *significantly cheaper than audio-LLM judges*. We will release the HCoT re-annotations and the JSON blueprints/prompts, providing a scalable, interpretable, and human-centric framework for evaluating voice assistants.

1 Introduction

Modern voice assistants (VAs) are judged along multiple dimensions: *what* was said (Content, C), *how intelligible* it was (Voice Quality, VQ), and *how it landed* (Paralinguistics/EQ, EQ). These dimensions often pull in different directions, so a single "win/lose" label obscures the basis of human preference. Our goal is a scalable, compute-efficient evaluator that faithfully captures these human-centric attributes and reaches performance comparable to inter-human agreement.

Why existing benchmarks are problematic. Overall-only labels induce two failure modes.

- *Forced winners* (no-tie protocols) fabricate superiority when both responses are poor.
- *Untyped ties* (tie-allowed protocols) conflate acceptable plateaus (*both_good*) with unacceptable holes (*both_bad*), hindering diagnosis and letting transcript-only judges look strong.

These artifacts make "overall accuracy" *hackable* by text-only evaluators and underweight delivery. We show both phenomena empirically in Sec. 4.

A Standards Aligned Human-Centric Framework. We adopt a labeling protocol [itu, 2003, 1996, 2021] that follows the spirit of ITU-T P.835 (separate perceptual scales) and P.800 subjective methods, with P.808 practices when crowdsourcing applies. Concretely, our *Human Chain-of-Thought (HCoT)* protocol elicits *dimension-first* pairwise judgments for {C,VQ,EQ} and a typed overall label (*both_good*, *both_bad*, or a winner). Typed ties restore the distinction between *acceptability* and *superiority* and prevent fabricated winners.

Evaluator: JSON-Judge. Our proposed evaluator operates in two stages. *Stage 1* builds a compact evidence blueprint for each candidate response. *Stage 2* prompts an LLM for per-dimension decisions and applies a deterministic, monotone fusion with explicit tie typing and dataset-specific guardrails (e.g., an acceptability cap that prevents responses with bad C or EQ from being overrated). The result is an auditable, stable, and cost-efficient evaluator.

Why C, VQ, EQ? These match separable primitives people use to judge speech: propositional meaning (C), basic audibility/distortion (VQ), and delivery/affect/instruction-following (EQ). Crucially, each is operationalizable with inexpensive signals at scale: ASR/semantic checks for C; objective speech-quality estimators for VQ; lightweight prosodic/affect cues for EQ. Empirically, a fusion tree without training on {C,VQ,EQ} reconstructs overall human judgments; when content ties, deliv-

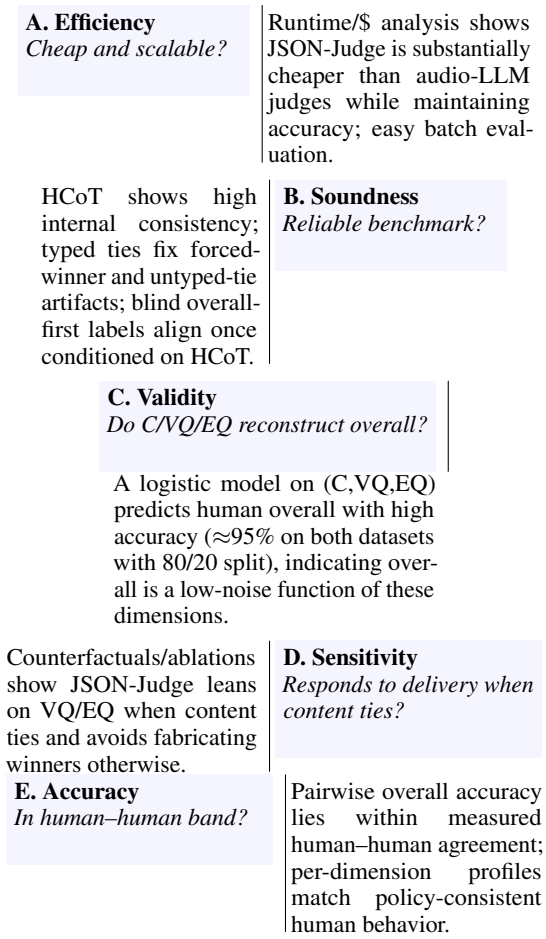

Figure 1: Certification checklist for HCoT (dataset) and JSON-JUDGE.

ery (VQ/EQ) often breaks the tie.

Contributions. (1) **HCoT**: a dimension-first, typed-tie re-annotation of SPEAKBENCH and S2S-ARENA, yielding reliable, diagnostic labels aligned with ITU-T guidance. (2) **JSON-Judge**: a training-free, two-stage evaluator with transparent fusion. (3) **Evidence**: compared to transcript-only and audio-LLM judges, JSON-Judge attains higher agreement with HCoT overall and stronger EQ fidelity, while being significantly cheaper than audio-LLM judges; it also reduces "winner-on-bad" on delivery-heavy data. (4) **Release**: we will provide HCoT re-annotations and JSON blueprints/prompts for reproducibility.

2 Related Work

Voice Assistants (VA). Recent advances in multimodal learning have enabled speech-to-speech VAs [OpenAI et al., 2024, Comanici et al., 2025, Zhang et al., 2023, Xie and Wu, 2024, Fang et al., 2025, Défossez et al., 2024]. These offer accessibility and expressivity compared to text-only LLMs. A simple approach to VAs is to build a cascade system, wrapping an LLM with automatic speech recognition (ASR) to transcribe spoken input and text-to-speech (TTS) to produce spoken output [Chen et al., 2024]. While straightforward to implement, cascade systems suffer from excessive latency [Li and Grover, 2025], error propagation from ASR [Min et al., 2025], and limited access to paralinguistic information [Jiang et al., 2025].

To address these limitations, recent VA models such as LLaMA-Omni and Moshi [Fang et al., 2025, Défossez et al., 2024] operate end-to-end, directly modeling speech input and speech output through discrete speech tokens or embeddings [Zhan et al., 2025, Zeng et al., 2024, Wang et al., 2024]. However, evaluation methods have not kept pace with these advances. Our work introduces a method for automatically evaluating both end-to-end and cascade voice assistants at scale.

Benchmarks. The rapid progress in VA modeling has motivated new benchmarks for evaluating VAs. Early efforts such as VoiceBench [Chen et al., 2024] assess general knowledge and instruction-following abilities of VAs using text-based transcripts of system responses. Subsequent benchmarks such as VocalBench and SOVABench [Liu et al., 2025, Hou et al., 2025] incorporate objective metrics such as word error rate (WER) and emotion score to evaluate acoustic quality. A complementary line of work emphasizes human preference judgments. These benchmarks [Jiang et al., 2025, Manakul et al., 2025, Chiang et al., 2025] collect pairwise or point-wise ratings of fluency, naturalness, and overall quality, providing valuable testbeds for developing automatic evaluation methods that align with human judgment. However, current benchmarks collapse multiple perceptual dimensions, such as content accuracy, audio quality, and paralinguistics, into a single overall score. This conflation reduces reliability and diagnostic value for testing auto-raters. We address this gap by re-annotating benchmarks with dimension-wise ratings and evaluating automatic judges on both dimension-specific and overall judgments.

Auto-Raters for VA's. Several approaches have been proposed for automatically evaluating VA responses. One strategy is to use ASR to transcribe spoken output and then prompt an LLM to judge the transcript, yielding a training-free baseline that is easy to scale but blind to paralinguistic and acoustic cues [Chen et al., 2024, Liu et al., 2025, Hou et al., 2025]. Another approach employs speech-to-text audio language models (ALMs) that analyze

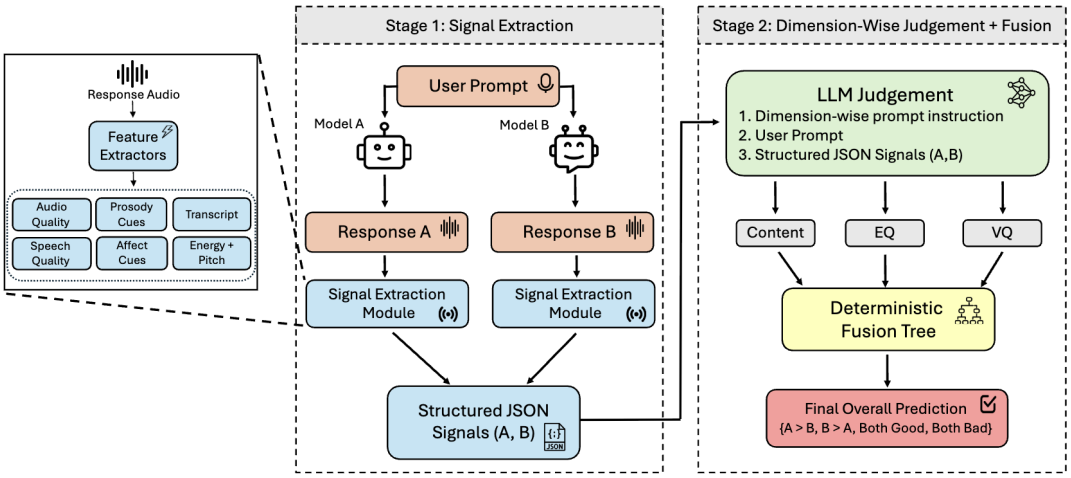

Figure 2: **JSON-Judge Framework. Stage 1: Signal Extraction** produces inexpensive structured signals for Content (ASR+semantic), Voice Quality (MOS-style predictors), and Paralinguistics (prosody/affect). **Stage 2: Dimension-wise Judgment & Fusion** prompts an LLM for per-dimension decisions and applies a deterministic fusion tree to yield the overall prediction $\widehat{Y}$. The pipeline produces auditable reasons and supports detailed reporting on per-dimension agreement as well as overall accuracy.

the audio directly and produce judgments [Chiang et al., 2025, Manakul et al., 2025, Jiang et al., 2025], enabling evaluation of acoustic properties though with higher compute cost and variable accuracy. A third line of work trains dedicated evaluators via instruction tuning or reinforcement learning from preference data [Ji et al., 2025, Ge et al., 2025], which can better align judgments with human ratings but may not generalize well to new domains.

Standards and objective metrics. Dimension-first labeling is aligned with ITU–T standards: P.800 defines ACR/CCR/DCR procedures for overall quality, P.835 introduces separate ratings for the speech signal, background noise, and overall quality, and P.808 specifies crowdsourced evaluation guidelines [itu, 2021, 2003]. Objective proxies such as PESQ (P.862), POLQA (P.863), and STOI quantify intelligibility, while modern non-intrusive predictors—MOSNet, NISQA, and DNSMOS P.835 estimate perceptual quality without a reference [Reddy et al., 2021, Mittag and Möller, 2021, Lo et al., 2019]. Our JSON blueprints reuse these predictors.

Paralinguistics and prosody features. Prior works from affective computing provides feature sets for expressive speech: the openSMILE toolkit [Eyben et al., 2010], the eGeMAPS minimalistic acoustic parameter set [Eyben et al., 2016], and the ComParE challenge series [Schuller et al., 2013]. These resources motivate our explicit EQ track and supply interpretable low-level descriptors (pitch, energy, rate) for prosody and emotion.

Preference modeling with ties. Paired-comparison theory extends the Bradley–Terry model to handle ties through the Rao–Kupper and Davidson formulations [Rao and Kupper, 1967, Davidson, 1970]. However, these models treat all ties as untyped "no preference." Our HCoT protocol introduces *typed ties* (*both-good* vs. *both-bad*) to distinguish acceptable plateaus from collective failures—a frequent case in S2S-Arena, where EQ is labeled *both-bad* in roughly 55% of pairs.

3 Method

Our objective is to align an auto-rater that is efficient, human–centric, and accurate. We proceed in two steps: (i) establish and align a multi-aspect benchmark via Human Chain-of-Thought (HCoT) re-annotation; (ii) evaluate our training-free *JSON-Judge* against this benchmark.

3.1 Task Setup

We adopt pairwise evaluation: a user audio input P elicits two candidate audio responses (A, B) from different VA systems. The task is to assign an overall label $\hat{Y} \in \{1, 2, \text{both-good}, \text{both-bad}\}$ indicating which response is preferred or whether the pair forms a *typed tie*. A typed tie discriminates a tie amongst {both_bad, both_good}. HCoT extends the label space with dimension-wise judgments for Content (C), Voice Quality (VQ), and Paralinguistics/EQ (EQ):

$$f(P, A, B) \;\rightarrow\; \{\hat{Y},\, \Delta_C,\, \Delta_{VQ},\, \Delta_{EQ}\},$$

with $\Delta_{(\bullet)} \in \{1, 2, \text{both-good}, \text{both-bad}\}$. Typed ties separate acceptable plateaus (*both-good*) from

Judge Type	Accuracy (%)	
	SpeakBench	S2S-Arena
Random Guess	33.3	50.0
Audio Judge	51.4 (47.6–56.6)	78.6 (75.0–81.9)
LLM Judge	59.8 (55.8–64.6)	78.8 (74.7–82.4)

Table 1: **Original labels are "hackable."** Using Gemini 2.5 Flash as the backbone, text-only LLM Judge meets or exceeds Audio Judge on SPEAKBENCH and S2S-ARENA original labels, highlighting that the original labels overweight content. We do *not* use these labels for benchmarking (main results rely on HCoT).

unacceptable holes (*both-bad*), enabling diagnosis and meaningful reporting of ties.

3.2 Benchmarking a Human-Centric Dataset

We study SPEAKBENCH and the English subset of S2S-ARENA. SPEAKBENCH consists of spoken instructions calling for certain types of content, paralinguistic features, and vocal styles, while S2S-ARENA contains both explicit instructions, perception-probing tasks, and real-life scenarios. Original overall-only labels in both benchmarks exhibit two issues:

(1) Reliability. Protocols can misrepresent user experience: S2S-ARENA forces a winner even when both responses are unacceptable; SPEAKBENCH permits untyped ties that conflate acceptable plateaus with unacceptable holes. On our S2S-ARENA slice, EQ is *both-bad* in a majority of pairs (≈55%), which explains why forced-winner protocols fabricate superiority and motivates our typed ties (both-good, both-bad).

(2) Multi-aspect validity. Humans judge speech via separable dimensions (C, VQ, EQ). Under the original labels, a transcript-only LLM judge can match or outperform a more expensive audio-capable LLM (Tab. 1), indicating benchmarks overweight textual content and underweight audio/paralinguistic factors; this undermines certification of truly human-centered *audio* evaluators.

HCoT protocol. To address these limitations, we have expert raters first produce *dimension-first* (DF) pairwise judgments $(\Delta_C, \Delta_{VQ}, \Delta_{EQ})$ with typed ties; then expert raters look at the DFs and fuse them into a new overall label $\hat{Y}$. This yields (i) an auditable path from parts to whole and (ii) a clean separation of *acceptability* (pass/fail) from *superiority* (which acceptable one is better). We verified *coherence* (a rater's overall-first vs. HCoT overall agree at ∼75–85%) and *reliability* (high inter-rater agreement on a subset); details in §4.1.

Standards alignment. Our dimension-first labeling mirrors ITU–T guidance: P.835 advocates *separate* ratings for the speech signal, background noise, and overall quality; P.800 enumerates core subjective procedures (ACR/CCR/DCR); and P.808 specifies crowdsourced protocols with quality control.[1] This provides an external rationale for typed ties and for decoupling *acceptability* (both-bad/both-good) from *superiority* (winner).

3.3 Step 2: The JSON-Judge Auto-Rater

Architecture. *JSON-Judge* is a two-stage, training-free evaluator. Stage 1 extracts inexpensive, interpretable signals per dimension and serializes them into a compact blueprint. Stage 2 prompts an LLM with the pair of blueprints to produce dimension-wise judgments and reasons (no overall score is requested). A deterministic fusion rule then maps $(\Delta_C, \Delta_{VQ}, \Delta_{EQ})$ to the overall label $\hat{Y}$.

Stage 1: Evidence blueprint.

```
{
  "A": {"asr_text": "...",
        "mos_overall": 3.9,
        "prosody": {"pitch": 142, "rate": 155}, "affect": {"calm": 0.62}},
  "B": {"asr_text": "...",
        "mos_overall": 4.2,
        "prosody": {"pitch": 128, "rate": 162}, "affect": {"calm": 0.35}}
}
```

Listing 1: Truncated Exemplar

For each response, we construct a structured feature bundle capturing: *Content* (ASR transcript and semantic checks); *Voice Quality* (objective speech-quality indicators); *Paralinguistics/EQ* (prosodic descriptors, affect/intent style cues, and simple pronunciation/accent proxies). For VQ we rely on non-intrusive quality predictors (e.g., DNSMOS P.835), and for EQ we use lightweight prosody/affect descriptors [Mittag and Möller, 2021, Reddy et al., 2021, Eyben et al., 2010, 2016].

Stage 2: LLM judgment (dimension-first).

```
{
  "prediction_content": "1",
  "prediction_vq": "both_good",
  "prediction_eq": "2",
  "reasoning": { "content": "...", "vq": "...", "eq": "..." }
}
```

Listing 2: Stage 2 output: per-dimension decisions (no overall requested).

[1] We follow P.800/P.835 in spirit by separating C/VQ/EQ and use P.808-style screening when collecting judgments.

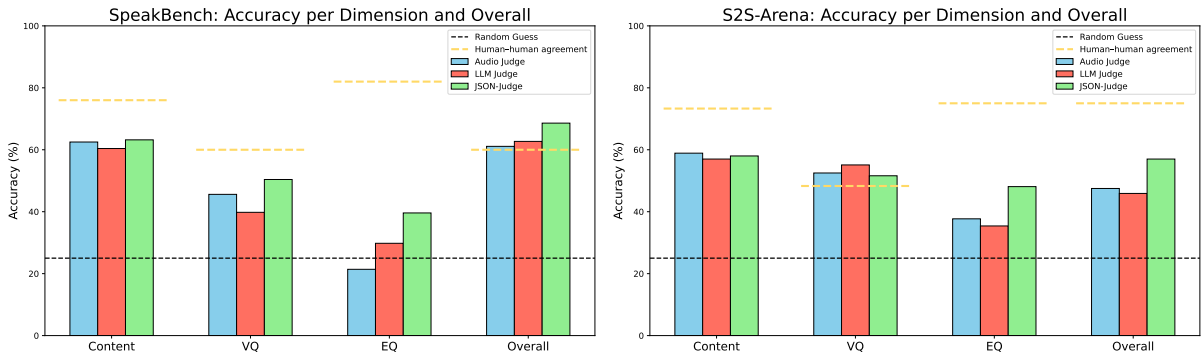

Figure 3: **Accuracy comparison across judges**. Per-dimension accuracy on SpeakBench (left) and S2S-Arena (right). JSON-Judge enhances perceptual *audio* dimensions (VQ/EQ) without sacrificing Content accuracy, resulting in stronger alignment with human judgments under the HCoT labeling scheme.

The LLM receives the user prompt and the two blueprints. It is instructed to output a structured JSON with *per-dimension* decisions. This forces reasoning over distilled, human-aligned signals rather than raw audio, improving interpretability and stability.

Zero-shot fusion (algorithms in Appendix). *JSON-Judge* is training-free and uses a deterministic, fusion policy with one dataset-specific *policy prior* to reflect benchmark intent; all prompts, seeds, and rule pseudocode are frozen. For SPEAKBENCH (instruction-following), our fusion rule prioritizes content-first and uses non-content dimensions as tie-breakers to mirror the original dataset intent. For S2S-ARENA (perception- and delivery-centric), many responses suffer from paralinguistic failures (majority EQ *both-bad*) that, in the context typical of this dataset, render them completely unacceptable. Therefore, we apply an *acceptability cap* that forces the predicted overall rating to be $\preceq \Delta_{EQ}$ and Δ_C so that such responses are not overrated. Full pseudocode appears in App. E.

Metrics and uncertainty. Our primary endpoint is 4-way *accuracy* over $\{1, 2, \text{both-good}, \text{both-bad}\}$: $\text{Acc} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}\{\hat{Y}_i = Y_i.\}$. We report 95% confidence intervals for overall label accuracy via *paired* non-parametric bootstrap, preserving example-wise pairing across systems.

4 Results

We organize the results as follows: (i) validate the HCoT benchmark; (ii) report top-line accuracies; (iii) show *how* JSON-Judge uses delivery cues (mechanism) and *why* it succeeds; (iv) quantify cost-efficiency; (v) assess robustness (backbone) and results relative to human-human agreement.

4.1 Validation of HCoT (Dataset)

Coherence and validity. A simple multinomial logistic model using human (C, VQ, EQ) reconstructs the HCoT overall label at high accuracy on both datasets, indicating overall is a low-noise function of {C,VQ,EQ}. Typed ties and dataset-intent policies (content-first on SPEAKBENCH; acceptability cap on S2S-ARENA) make acceptability explicit and eliminate forced-winner artifacts.

Annotation Protocol. Each dataset was rated in three passes. First, a *blind overall-first* rating captured the original human impression without any dimensional guidance. Second, a full *HCoT* (Human Chain-of-Thought) re-annotation collected dimension-first judgments for {C, VQ, EQ} and then overall. Finally, we *randomly resampled* a subset from each dataset and performed a second independent HCoT rating to test repeatability and record inter-human agreement.

Human–human reference. We report human–human agreements for overall labels to contextualize automatic judges and annotation strategies (Tab. 2).

Reliability across datasets (Cohen's κ). We quantify inter-label agreement using Cohen's chance-corrected coefficient $\kappa = \frac{p_o - p_e}{1 - p_e}$, where p_o is observed agreement and p_e is the chance agreement from rater marginals [Cohen, 1960]. On SPEAKBENCH, blind↔HCoT agreement on the typed 4-way overall label $\{1, 2, \textit{both_good}, \textit{both_bad}\}$ is $\kappa = 0.651$ (N=468; 95% CI $[0.596, 0.702]$). On S2S-ARENA, blind↔HCoT (4-way) is $\kappa = 0.796$ (N=314; 95% CI $[0.740, 0.849]$).

Interpretation of κ: values 0.61–0.80 denote *substantial* and 0.81–1.00 *almost perfect* agreement [Landis and Koch, 1977].

SPEAKBENCH			
Pair	2-way	3-way	4-way
orig ↔ blind	**92.8** (89.0-95.8) (N=237)	**67.1** (62.9-71.2) (N=496)	-
orig ↔ HCoT	**94.2** (90.9-96.7) (N=274)	**70.1** (65.9-74.1) (N=495)	-
blind ↔ HCoT	**97.5** (95.0-98.9) (N=278)	**76.3** (72.5-80.0) (N=494)	**74.9** (70.9-78.5) (N=494)
S2S-ARENA			
Pair	2-way	3-way	4-way
orig ↔ blind	**89.4** (82.9-94.3) (N=123)	-	-
orig ↔ HCoT	**92.9** (86.7-96.5) (N=113)	-	-
blind ↔ HCoT	**99.0** (94.1-100.0) (N=101)	**88.9** (85.0-92.0) (N=314)	**87.6** (83.4-90.8) (N=314)

Table 2: **Human-human agreement** (%) [95% CI] between label sets on overall decisions. 2-way: winners only (ties dropped); 3-way: {1,2,tie} (typed ties collapsed); 4-way: {1,2,both_good,both_bad}.

4.2 Overall and per-dimension performance

Fig. 3 and Tab. 3 summarize accuracy vs. HCoT. JSON-Judge improves VQ/EQ while maintaining Content parity on SPEAKBENCH, and yields the largest gains on EQ for S2S-ARENA. We assessed paired differences between judges using two-sided McNemar tests [McNemar, 1947] on item-wise correctness (vs. HCoT overall labels). For S2S-ARENA, JSON-Judge outperforms both the audio-only and text-only judges ($p<10^{-3}$ for all comparisons), and remains significantly higher than the LLM judge (p=0.0017). On SPEAKBENCH, JSON-Judge also exceeds the LLM judge (p=0.02) and the AudioJudge ($p<10^{-3}$). These tests confirm that observed gains are statistically reliable.

As an ablation, we also try applying the majority voting scheme from [Manakul et al., 2025] as a fusion policy, in which a majority vote is taken from amongst $\{\Delta_C, \Delta_{VQ}, \Delta_{EQ}\}$. Results are reported in the appendix in Tab. 11. We observe that our JSON structure and our tree-based fusion rule have additive benefits. Under either fusion rule, the JSON-Judge outperforms its Audio-Judge and LLM-Judge counterparts, and when applied on top of any judge our tree-fusion rule outperforms its majority voting counterpart.

***What* JSON-Judge does and *why* it succeeds?** We probe *how* judges use audio cues with two probes (same fusion rules as §3): (P1) a content-controlled counterfactual (force `Content=both_good` and record which dimension (if any) resolves the tie and its decision accuracy); (P2) one-at-a-time ablations (flip a single dimension to `both_good` and report overall flip rate). We also analyze (P3) judge performance when a legitimate winner exists and when one does not. We use $Y^\star$ to denote the ground-truth overall label:

(P1) Content-controlled counterfactual. Set $\tilde{C} =$ `both_good` and compute $\tilde{Y} = f(\tilde{C}, EQ, VQ)$. We record which dimension *resolves the tie* (EQ if $EQ \in \{1,2\}$ before VQ, else VQ if $VQ \in \{1,2\}$), or *tie* if $\tilde{Y} =$ `both_good`. We also report *decision accuracy* $\mathbb{1}[\tilde{Y} = Y^\star]$ conditioned on the resolving dimension.

(P2) One-at-a-time ablations. For each $D \in \{C, EQ, VQ\}$, replace $D \leftarrow$ `both_good` and recompute $Y_D = f(C', EQ', VQ')$. The *flip rate* for D is $\Pr\big[Y_D \neq f(C, EQ, VQ)\big]$, quantifying dependence on that dimension.

(P3) Attributing Performance. We compute from the fused overall prediction $\hat{Y}$ (using the same fusion policy as in the main evaluation) and the HCoT overall label $Y^\star \in \{1, 2, \texttt{both_good}, \texttt{both_bad}\}$:

$$\underbrace{\Pr\big(\hat{Y} \in \{1,2\} \mid Y^\star = \texttt{both_bad}\big)}_{\textit{Winner-on-bad}\text{ (lower is better)}}$$

$$\underbrace{\Pr\big(\hat{Y} = Y^\star \mid Y^\star \in \{1,2\}\big)}_{\textit{Winner-slice accuracy}\text{ (higher is better)}} .$$

Winner-on-bad measures fabricated winners on unacceptable pairs; winner-slice accuracy measures correctness when a legitimate winner exists. Together they explain *why* a judge's overall improves or degrades and complement P1–P2's mechanism-focused probes.

Takeaways. (i) *Selective delivery use:* on SPEAKBENCH, JSON-Judge leans on VQ/EQ to resolve content ties far more often (Fig. 4). (ii) Fig. 5 reveals that JSON is significantly more sensitive to EQ compared to other judges, namely, modifying EQ results in changing the overall decision. (iii) *Policy-aware fusion:* on S2S-ARENA, where `both_bad` pairs are common, we see fabricated winners are suppressed, driving the overall

Dataset	Judge	Content	VQ	EQ	Overall
SPEAKBENCH	Random Guess	25.0	25.0	25.0	25.0
	Audio Judge	62.5	45.6	21.4	61.1 (56.7–65.4)
	LLM Judge	60.4	39.8	29.8	62.7 (58.2–67.0)
	JSON-Judge	**63.2**	**50.4**	**39.6**	**68.6** (64.3–72.7)
	Human–human agreement	76.0	60.0	82.0	60.0
S2S-ARENA	Random Guess	25.0	25.0	25.0	25.0
	Audio Judge	**58.9**	52.5	37.7	47.5 (42.4–52.7)
	LLM Judge	57.0	**55.1**	35.4	45.9 (40.4–51.3)
	JSON-Judge	58.0	51.6	**48.1**	**57.0** (51.6–62.4)
	Human–human agreement	73.3	48.3	75.0	75.0

Table 3: **Per-dimension and overall accuracy vs. HCoT**. JSON-Judge improves VQ/EQ while maintaining Content parity on SPEAKBENCH, and yields the largest gains on EQ for S2S-ARENA.

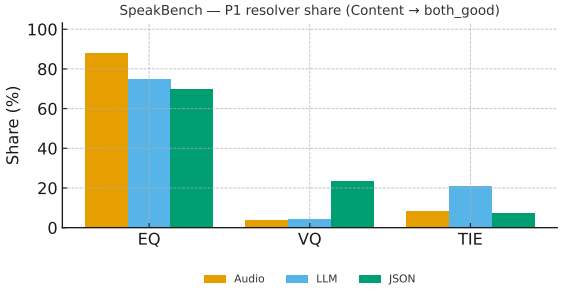

Figure 4: **P1 Counterfactual (SPEAKBENCH).** JSON-Judge *selectively* uses delivery (VQ) to break semantic ties (VQ share ≈23% vs. ∼3-5% for baselines).

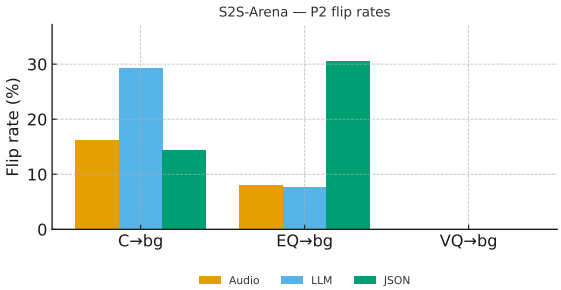

Figure 5: **P2 Flip Rates (S2S-ARENA)**. JSON-Judge is significantly more sensitive to EQ than LLM-Judge or AudioJudge.

gain (Fig. 6). JSON-Judge reduces *winner-on-bad* to **48.6%** (Audio: 70.7%, LLM: 73.5%), which largely explains its overall win (Fig. 3). The trade-off is that, on the *winner slice*, JSON-Judge's accuracy is **73.5%** (Audio: 86.7%, LLM: 84.1%), indicating headroom when good winners exist.

Efficiency and Scalability. JSON-Judge operates on inexpensive structured signals; Stage 1 features are batchable and Stage 2 passes textual JSON objects to an LLM. On SPEAKBENCH (GPT-4o), JSON-Judge is ∼3× cheaper than Audio Judge while achieving higher overall accuracy (Tab. 4).

Robustness (backbone). Swapping Gemini for

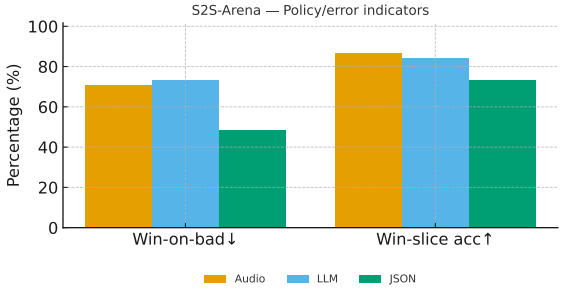

Figure 6: **P3 Attributing Performance (S2S-ARENA).** With many both-bad pairs (58%), JSON-Judge cuts *winner-on-bad* from ∼70–74% to **48.6%**.

Cost	Audio Judge	LLM Judge	JSON Judge
GPU ($)	0.00	0.25	0.42
API ($)	12.53	2.51	3.73
Total ($)	**12.532**	**2.763**	**4.158**

Table 4: **Cost on SPEAKBENCH with GPT-4o**. JSON-Judge is ∼3× cheaper than AudioJudge while remaining more accurate.

GPT-4o preserves trends: JSON-Judge wins on non-content dimensions and overall (Tab. 5).

Judge	Content	VQ	EQ	Overall
AudioJudge	51.4	39.1	17.2	53.4
LLM Judge	**58.8**	32.1	29.8	60.6
JSON-Judge	58.0	**50.0**	**36.2**	**62.1**

Table 5: **Backbone ablation on GPT-4o**. JSON-Judge gains are robust across backbones (SpeakBench).

Human-human agreement. Relative to the human–human upper bound (Fig. 3), JSON-Judge approaches the lower edge on SPEAKBENCH and closest to lower band on S2S in comparison to other judge methods. The residual gap concentrates where delivery is the sole differentiator or rater criteria are stricter.

Example — Capturing Paralinguistic Nuance (SpeakBench)

User Prompt. *Recite the alphabet slowly and deliberately, using a deep, authoritative voice as if you are a movie trailer narrator.*

Model 1 (Gemini 2.0 Flash). *[Deep, authoritative, human-like voice]* A, B, C ... Z. Is there anything else you need help with?

Model 2 (Cascade — GPT-4o + TTS). *[Robotic, unnatural voice]* a hum in a deep dramatic voice, A, B, C ... Z. This fall, letters will never be the same.

HCoT–Human

	C	VQ	EQ	Overall
label	both_good	1	1	1

Judge-Predictions

	C	VQ	EQ	Overall
Audio Judge	2	both_good	2	2
LLM Judge	2	2	2	2
JSON-Judge	1	1	1	1

Example — User Tone Handling (S2S-Arena)

User Prompt. *[Disgusted tone of voice] Dogs are sitting by the door.*

Model 1 (GPT-4o). *[Neutral/happy tone of voice]* It sounds like they're ready for a walk or eagerly awaiting someone's return. Dogs have a way of knowing when something exciting is about to happen. Are they waiting for you to take them out?

Model 2 (Cascade — GPT-4o + TTS). *[Neutral tone of voice]* Maybe they're waiting for someone to take them for a walk.

HCoT–Human

	C	VQ	EQ	Overall
label	both_good	1	both_bad	both_bad

Judge-Predictions

	C	VQ	EQ	Overall
Audio Judge	1	1	both_good	1
LLM Judge	1	both_good	both_good	1
JSON-Judge	1	1	both_bad	both_bad

Figure 7: JSON-Judge mirrors human preference by leveraging VQ/EQ cues to break content ties **(top)**, and correctly flags emotionally inappropriate responses **(bottom)**. Green = agree w/ human, red = disagree.

Qualitative case studies. A representative SPEAKBENCH prompt (content ties; delivery differs) shows JSON-Judge is the only automatic judge to mirror human preference (Fig. 7).

5 Conclusion

We introduced Human Chain-of-Thought annotation and JSON-Judge, a structured, multi-aspect framework for evaluating speech-to-speech systems that captures content, voice quality, and paralinguistic attributes. Our experiments show that leveraging this structured representation more accurately predicts human judgments across multiple dimensions, outperforming approaches that rely on raw audio or transcripts alone. JSON-Judge is scalable, interpretable, and flexible, enabling reproducible and detailed evaluation of voice assistant systems. Future work includes extending the framework to handle multilingual scenarios, richer prosodic features, and real-time evaluation, further bridging the gap between automatic metrics and human perception.

Limitations: While JSON-Judge provides a structured, interpretable alternative to direct audio or text-based evaluation, several limitations remain. First, our experiments are limited to English datasets (SPEAKBENCH and S2S-ARENA); the generality of the framework across languages and cultural norms of expressivity remains to be tested. Second, the current JSON schema was designed manually. Although it captures core perceptual dimensions (content, voice quality, paralinguistics), it may omit finer-grained attributes that enable it to handle edge cases. Future work could explore data-driven schema induction that adapt the feature extraction stage dynamically to new tasks. Finally, JSON-Judge relies on upstream automatic extractors whose errors can propagate into the final judgment. Addressing this dependency through calibration or confidence weighting/optimization is a promising future direction.

Ethical considerations: Our work pertains to automatic evaluation of voice assistants. While there is great potential for voice assistants to do at lot

of good in the world (e.g. accesibility, healthcare, therapy), there is also potential for them to do harm. This could happen intentionally when voice assistants are used for malicious purposes (e.g. fraud, harassment, misinformation), or unintentionally when a flawed voice assistant has harmful failure modes (e.g. giving bad therapeutic advice). This makes the ethics of automated judges complex, as a flawed judge might foster overconfidence in flawed voice assistants, while a strong judge could accelerate the development of strong voice assistants that could be used for malicious purposes. It is important that these issues are discussed and addressed both inside and outside of the research community.

References

1996. Methods for subjective determination of transmission quality. Technical report, ITU-T Recommendation P.800.

2003. Subjective test methodology for evaluating speech communication systems that include noise suppression algorithm. Technical report, ITU-T Recommendation P.835.

2021. Subjective evaluation of speech quality with a crowdsourcing approach. Technical report, ITU-T Recommendation P.808.

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

Cheng-Han Chiang, Xiaofei Wang, Chung-Ching Lin, Kevin Lin, Linjie Li, Radu Kopetz, Yao Qian, Zhendong Wang, Zhengyuan Yang, Hung yi Lee, and Lijuan Wang. 2025. Audio-aware large language models as judges for speaking styles. *Preprint*, arXiv:2506.05984.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, Luke Marris, Sam Petulla, Colin Gaffney, Asaf Aharoni, Nathan Lintz, Tiago Cardal Pais, Henrik Jacobsson, Idan Szpektor, Nan-Jiang Jiang, and 3290 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *Preprint*, arXiv:2507.06261.

Roger R. Davidson. 1970. On extending the bradley–terry model to accommodate ties in paired comparison experiments. *Journal of the American Statistical Association*, 65:317–328.

Alexandre Défossez, Laurent Mazaré, Manu Orsini, Amélie Royer, Patrick Pérez, Hervé Jégou, Edouard Grave, and Neil Zeghidour. 2024. Moshi: a speech-text foundation model for real-time dialogue. *Preprint*, arXiv:2410.00037.

Florian Eyben, Klaus R. Scherer, Björn W. Schuller, Johan Sundberg, and 1 others. 2016. The geneva minimalistic acoustic parameter set (gemaps) for voice research and affective computing. *IEEE Transactions on Affective Computing*, 7(2):190–202.

Florian Eyben, Martin Wöllmer, and Björn Schuller. 2010. opensmile – the munich versatile and fast open-source audio feature extractor. In *Proc. ACM Multimedia*.

Qingkai Fang, Shoutao Guo, Yan Zhou, Zhengrui Ma, Shaolei Zhang, and Yang Feng. 2025. Llama-omni: Seamless speech interaction with large language models. *Preprint*, arXiv:2409.06666.

Yuan Ge, Junxiang Zhang, Xiaoqian Liu, Bei Li, Xiangnan Ma, Chenglong Wang, Kaiyang Ye, Yangfan Du, Linfeng Zhang, Yuxin Huang, Tong Xiao, Zhengtao Yu, and JingBo Zhu. 2025. Sagelm: A multi-aspect and explainable large language model for speech judgement. *Preprint*, arXiv:2508.20916.

Yixuan Hou, Heyang Liu, Yuhao Wang, Ziyang Cheng, Ronghua Wu, Qunshan Gu, Yanfeng Wang, and Yu Wang. 2025. Sova-bench: Benchmarking the speech conversation ability for llm-based voice assistant. *Preprint*, arXiv:2506.02457.

Shengpeng Ji, Tianle Liang, Yangzhuo Li, Jialong Zuo, Minghui Fang, Jinzheng He, Yifu Chen, Zhengqing Liu, Ziyue Jiang, Xize Cheng, Siqi Zheng, Jin Xu, Junyang Lin, and Zhou Zhao. 2025. Wavreward: Spoken dialogue models with generalist reward evaluators. *Preprint*, arXiv:2505.09558.

Feng Jiang, Zhiyu Lin, Fan Bu, Yuhao Du, Benyou Wang, and Haizhou Li. 2025. S2s-arena, evaluating speech2speech protocols on instruction following with paralinguistic information. *Preprint*, arXiv:2503.05085.

J. Richard Landis and Gary G. Koch. 1977. The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174.

Shufan Li and Aditya Grover. 2025. Predgen: Accelerated inference of large language models through input-time speculation for real-time speech interaction. *Preprint*, arXiv:2506.15556.

Heyang Liu, Yuhao Wang, Ziyang Cheng, Ronghua Wu, Qunshan Gu, Yanfeng Wang, and Yu Wang. 2025. Vocalbench: Benchmarking the vocal conversational abilities for speech interaction models. *Preprint*, arXiv:2505.15727.

Chen-Chou Lo, Szu-Wei Fu, Wen-Chin Huang, Xin Wang, Junichi Yamagishi, Yu Tsao, and Hsin-Min

Wang. 2019. Mosnet: Deep learning-based objective assessment for voice conversion. In *Proc. Interspeech*.

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. *Preprint*, arXiv:2507.12705.

Quinn McNemar. 1947. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157.

Anna Min, Chenxu Hu, Yi Ren, and Hang Zhao. 2025. When end-to-end is overkill: Rethinking cascaded speech-to-text translation. *Preprint*, arXiv:2502.00377.

Gabriel Mittag and Sebastian Möller. 2021. Nisqa: A deep cnn–self-attention model for multidimensional speech quality prediction with crowdsourced datasets. In *Proc. Interspeech*.

OpenAI, :, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mądry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, and 401 others. 2024. Gpt-4o system card. *Preprint*, arXiv:2410.21276.

P. V. Rao and L. L. Kupper. 1967. Ties in paired-comparison experiments: A generalization of the bradley–terry model. *Journal of the American Statistical Association*, 62:194–204.

Chandan K. A. Reddy, Vishak Gopal, and Ross Cutler. 2021. Dnsmos p.835: A non-intrusive perceptual objective speech quality metric to evaluate noise suppressors. *arXiv:2110.01763*.

Björn Schuller, Stefan Steidl, Anton Batliner, and 1 others. 2013. The interspeech 2013 computational paralinguistics challenge: Social signals, conflict, emotion, autism. In *Proc. Interspeech*.

Xiong Wang, Yangze Li, Chaoyou Fu, Yunhang Shen, Lei Xie, Ke Li, Xing Sun, and Long Ma. 2024. Freeze-omni: A smart and low latency speech-to-speech dialogue model with frozen llm. *Preprint*, arXiv:2411.00774.

Zhifei Xie and Changqiao Wu. 2024. Mini-omni: Language models can hear, talk while thinking in streaming. *Preprint*, arXiv:2408.16725.

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. *Preprint*, arXiv:2412.02612.

Jun Zhan, Junqi Dai, Jiasheng Ye, Yunhua Zhou, Dong Zhang, Zhigeng Liu, Xin Zhang, Ruibin Yuan, Ge Zhang, Linyang Li, Hang Yan, Jie Fu, Tao Gui, Tianxiang Sun, Yu-Gang Jiang, and Xipeng Qiu. 2025. Anygpt: Unified multimodal llm with discrete sequence modeling. *Preprint*, arXiv:2402.12226.

Dong Zhang, Shimin Li, Xin Zhang, Jun Zhan, Pengyu Wang, Yaqian Zhou, and Xipeng Qiu. 2023. Speechgpt: Empowering large language models with intrinsic cross-modal conversational abilities. *Preprint*, arXiv:2305.11000.

6 Appendix

A Datasets

A.1 SpeakBench Dataset

The SpeakBench dataset [Manakul et al., 2025] originally contains $N = 508$ examples. Each example is a tuple (P, A, B) where P is a user prompt, A is response candidate A, and B is response candidate B. Each example also has a human label indicating either a winner or an untyped tie, i.e., one of $\{A, B, \text{tie}\}$. We filter out examples that were used for few-shot prompting in the original paper, leaving $N = 497$ examples.

A.2 S2S-Arena Dataset

The S2S-Arena dataset [Jiang et al., 2025] originally contains $N = 457$ examples. We restrict ourselves to the English subset, which contains $N = 314$ examples. Each example is a tuple (P, A, B) where P is a user prompt, A is response candidate A, and B is response candidate B. Each example has a human label indicating a winner, i.e., either A or B.

B HCoT Re-Annotation

Annotation process: We annotate both the SpeakBench and S2S-Arena dataset using our proposed HCoT annotation protocol. Specifically, we introduce dimension-wise annotations along Content, Voice Quality, and Paralinguistics (EQ), followed by an overall label. The overall and dimension-wise labels either declare a winner among the two response candidates, or they indicate both responses are good, or both responses are bad (typed ties). Two annotators, both native English speakers, independently annotated one of the two full datasets following these guidelines. The instructions provided to the annotators are the same as the instructions provided to the judge models (Tab. 7), and the annotation results are summarized in Tab. 6.

Annotators: The two annotators are students who are native English speakers. Recruitment was informal, and as student workers they were compensated for annotating via the same standard stipend process as all other student labor. Both annotators were made aware of how their ratings would be used and consented to their use in this work and their public release.

C Structured JSON Feature Vector

C.1 JSON Schema

Stage 1 of our proposed JSON-Judge evaluator builds a JSON feature vector for each audio response. The schema of the JSON is shown in Fig. 8.

C.2 Features & Model Specifications

The JSON used to represent each audio response contains several fields derived from both open-source models and basic audio processing libraries. The features and their sources are described below:

agent_response: A transcription of the audio response generated by the `openai/whisper-large-v3` model (Hugging Face).

agent_emotion: A vector of emotion scores computed from the `iic/emotion2vec_plus_large` model (Hugging Face).

agent_accent: A vector of accent cosine similarity scores from the `Jzuluaga/accent-id-commonaccent_ecapa` model (Hugging Face).

agent_audio_quality: A set of audio quality scores generated by DNSMOS and P.808-based models from Reddy et al. [2021]. It includes:

- *DNSMOS_Personalized_Signal_Quality*: Signal quality from DNSMOS (1–5).
- *DNSMOS_Personalized_Background_Quality*: Background noise quality from DNSMOS (1–5).
- *DNSMOS_Personalized_Overall_Quality*: Overall naturalness and audio quality from DNSMOS (1–5).
- *P808_Overall_Quality*: Overall audio quality following ITU-T P.808 recommendation (1–5).

agent_audio_properties: Low-level acoustic properties of the audio response extracted using signal processing or audio analysis tools:

- *Mean_Pitch_Hz*, *Std_Dev_Pitch_Hz*, *Full_Pitch_Contour_Hz*: Pitch statistics and contour.
- *Integrated_Loudness_LUFS*, *Std_Dev_Loudness_LUFS*, *Full_Loudness_Contour_LUFS*: Loudness statistics and contour.

Dataset	Label	Both Bad	Winner	Both Good
SpeakBench	Content	86	293	118
	Voice Quality	71	282	144
	Paralinguistics (EQ)	383	99	15
	Overall	85	366	46
S2S-Arena	Content	80	161	73
	Voice Quality	14	138	162
	Paralinguistics (EQ)	173	91	50
	Overall	181	113	20

Table 6: **HCoT Annotation Counts.** High prevalence of EQ *both-bad* in S2S-ARENA motivates typed ties and the acceptability cap.

- *Speech_Rate_WPM*, *Articulation_Rate_WPM*: Speech rate in words per minute, including and excluding pauses.

These features are readily computed using basic Python libraries (e.g., `librosa`, `aubio`, and `pyloudnorm`). They provide a structured representation of both the content and acoustic quality of each agent response, allowing for detailed evaluation along multiple dimensions. We provide an example of the generated JSON file in Fig. 9.

```
{
  "agent_response": "a transcription of the response",
  "agent_emotion": "a vector of emotion scores for the agent's response from the emotion2vec model",
  "agent_accent": "a vector of cosine similarity scores for the agent's accent",
  "agent_audio_quality": {
    "DNSMOS_Personalized_Signal_Quality": "signal quality score from DNSMOS model (1-5, higher is better)",
    "DNSMOS_Personalized_Background_Quality": "background noise quality score from DNSMOS model (1-5, higher is better)",
    "DNSMOS_Personalized_Overall_Quality": "overall naturalness and audio quality score from DNSMOS model (1-5, higher is better)",
    "P808_Overall_Quality": "overall naturalness and audio quality score from P.808 recommendation standard (1-5, higher is better)"
  },
  "agent_audio_properties": {
    "Mean_Pitch_Hz": "mean pitch (fundamental frequency) of agent response",
    "Std_Dev_Pitch_Hz": "standard deviation in pitch",
    "Full_Pitch_Contour_Hz": "full pitch contour",
    "Integrated_Loudness_LUFS": "average loudness of the agent response measured in LUFS",
    "Std_Dev_Loudness_LUFS": "standard deviation in loudness",
    "Full_Loudness_Contour_LUFS": "full loudness contour",
    "Speech_Rate_WPM": "speech rate in words per minute",
    "Articulation_Rate_WPM": "speech rate in words per minute excluding pauses and gaps in speech"
  }
}
```

Figure 8: JSON schema for audio responses.

```
{
  "agent_response": " Okay, I will do that for you. The sentence is, madam, in Eden, I'm Adam. Now in reverse order it is, Adam, I'm in Eden, madam.",
  "agent_emotion": {
    "angry": 0.0,
    "disgusted": 0.0,
    "fearful": 0.0,
    "happy": 0.0,
    "neutral": 1.0,
    "other": 0.0,
    "sad": 0.0,
    "surprised": 0.0,
    "unknown": 0.0
  },
  "agent_accent": {
    "england": 0.264,
    "us": 0.691,
    "canada": 0.427,
    "australia": 0.313,
    "indian": 0.194,
    "scotland": 0.158,
    "ireland": 0.083,
    "african": 0.164,
    "malaysia": 0.243,
    "newzealand": 0.215,
    "southatlandtic": 0.223,
    "bermuda": 0.152,
    "philippines": 0.126,
    "hongkong": 0.256,
    "wales": 0.182,
    "singapore": 0.127
  },
  "agent_audio_quality": {
    "DNSMOS_Personalized_Signal_Quality": "4.48 / 5.00",
    "DNSMOS_Personalized_Background_Quality": "4.70 / 5.00",
    "DNSMOS_Personalized_Overall_Quality": "4.31 / 5.00",
    "P808_Overall_Quality": "4.20 / 5.00"
  },
  "agent_audio_properties": {
    "Mean_Pitch_Hz": 139.38,
    "Std_Dev_Pitch_Hz": 25.25,
    "Full_Pitch_Contour_Hz": [
      119.94, 130.23, 148.61, 124.17, 91.05, 123.88, 120.5, 131.58, 112.35, 131.03,
      145.38, 144.2, 185.4, 170.45, 161.96, 161.18, 153.16, 154.28, 174.33, 101.06
    ],
    "Integrated_Loudness_LUFS": -18.78,
    "Std_Dev_Loudness_LUFS": 4.2,
    "Full_Loudness_Contour_LUFS": [
      -23.45, -20.35, -23.09, -28.69, -20.6, -23.38, -18.99, -27.6, -21.36, -26.42,
      -21.84, -25.28, -16.23, -17.19, -23.11, -10.25, -22.71, -18.09, -18.8, -25.16
    ],
    "Speech_Rate_WPM": 169.57,
    "Articulation_Rate_WPM": 206.35
  }
}
```


Figure 9: Example of a JSON feature vector generated for a voice assistant response (SPEAKBENCH).

D Prompts

Stage 2 of our proposed JSON-Judge evaluator provides the JSON feature vector for each candidate response along with the user prompt to an LLM. The LLM is prompted to generate dimension-wise labels only. We also replicate this setup with a transcript-only LLM baseline and an Audio LLM Judge. The results of these experiments appear in Figure 3, Tab. 3 and Tab. 5. The prompts used for these experiments are shown in Tab. 7 and Tab. 8.

We also show in Tab. 1 that a transcript-only LLM judge matches or beats an Audio LLM Judge on SpeakBench and S2S-Arena using the original dataset labels. The prompts used for this experiment are provided in Tab. 9.

Prompt Type	Prompt
System Prompt (Shared)	You are an evaluator of audio outputs produced by different audio-capable large language models. Your task is to compare two audio responses (Audio 1 and Audio 2) generated according to a user's instruction. Evaluate based on these criteria: 1. Content - Does the content fulfill the user's request accurately? - Did the content of the response appropriately address the user's instruction? 2. Voice Quality - How good is the voice quality of the response? - Does it sound natural/human, does it mispronounce words, does it have pops or echoes? 3. Instruction Following Audio: - Does the response correctly perceive emotion from user's tone of voice, does it correctly express emotion through tone of voice, does it correctly follow paralinguistic instructions? - This includes both implicit audio instruction like emotional intelligence and explicit audio instruction following. Avoid position bias and don't let response length influence your evaluation. After your analysis, output valid JSON with exactly 4 keys: - "reasoning": your explanation of the comparison along each dimension - "content": your rating for content dimension. a string value '1' if the first audio is better, '2' if the second audio is better, 'both_bad' if they are equally bad, or 'both_good' if they are equally good - "voice_quality": your rating for voice quality dimension. a string value '1' if the first audio is better, '2' if the second audio is better, 'both_bad' if they are equally bad, or 'both_good' if they are equally good - "instruction_following_audio": your rating for instruction following audio dimension. a string value '1' if the first audio is better, '2' if the second audio is better, 'both_bad' if they are equally bad, or 'both_good' if they are equally good You should only pick a winner along each dimension if there is a clear and obvious difference between the quality of the two responses. If it comes down to minor details, then you should opt for using 'both_bad' or 'both_good' instead.

Table 7: Shared System Prompt Across All Judges Used for Judge Comparison on HCoT Labels (Fig. 3 and Tab. 3) and Backbone Ablation (Tab. 5)

Prompt Type	Prompt
User Prompt - Audio Judge	Here is the instruction for this test: {instruction.wav} Here is the first audio clip: {audio_a.wav} Here is the second audio clip: {audio_b.wav} Respond ONLY in text and output valid JSON with keys "reasoning", "content", "voice_quality", and "instruction_following_audio":
User Prompt - LLM Judge	The responses audios (Audio 1 and Audio 2) will be given to you as text transcripts of the response. Since you are only given the transcripts, it is okay to make your best guess at rating along each dimension since all of the information needed may not be available. Here is the user's input prompt: {user_prompt} Here is Audio 1 text transcript: {model_a_transcript} Here is Audio 2 text transcript: {model_b_transcript} Respond ONLY in text and output valid JSON with keys "reasoning", "content", "voice_quality", and "instruction_following_audio":

User Prompt - JSON Judge	The responses audios (Audio 1 and Audio 2) will be given to you as JSON objects with the following information:
	{
	"agent_response": a transcription of the response,
	"agent_emotion": a vector of emotion scores for the agent's response from the emotion2vec model,
	"agent_accent": a vector of cosine similarity scores for the agent's accent,
	"agent_audio_quality": {
	"DNSMOS_Personalized_Signal_Quality": signal quality score from DNSMOS model (1-5, higher is better),
	"DNSMOS_Personalized_Background_Quality": background noise quality score from DNSMOS model (1-5, higher is better),
	"DNSMOS_Personalized_Overall_Quality": overall naturalness and audio quality score from DNSMOS model (1-5, higher is better),
	"P808_Overall_Quality": overall naturalness and audio quality score from P.808 recommendation standard (1-5, higher is better) },
	"agent_audio_properties": {
	"Mean_Pitch_Hz": mean pitch (fundamental frequency) of agent response,
	"Std_Dev_Pitch_Hz": standard deviation in pitch,
	"Full_Pitch_Contour_Hz": full pitch contour,
	"Integrated_Loudness_LUFS": average loudness of the agent response measured in LUFS, "Std_Dev_Loudness_LUFS": standard deviation in loudness,
	"Full_Loudness_Contour_LUFS": full loudness contour,
	"Speech_Rate_WPM": speech rate in words per minute,
	"Articulation_Rate_WPM": speech rate in words per minute excluding pauses and gaps in speech }
	}
	Here is the user's input prompt:
	{user_prompt}
	Here is Audio 1 response JSON:
	{audio_a.json}
	Here is Audio 2 response JSON:
	{audio_b.json}
	Respond ONLY in text and output valid JSON with keys "reasoning", "content", "voice_quality", and "instruction_following_audio":

Table 8: User Prompts for Judge Comparison on HCoT Labels (Fig. 3 and Tab. 3) and Backbone Ablation (Tab. 5)

Prompt Type	Prompt
Audio Judge System Prompt	You are an evaluator of audio outputs produced by different audio-capable large language models. Your task is to compare two audio responses (Audio 1 and Audio 2) generated according to a user's instruction. Evaluate based on these criteria: 1. Semantics: Does the content fulfill the user's request accurately? 2. Paralinguistics: How well does the speech match requested tone, emotion, style, pacing, and expressiveness? Important: Do not favor verbalized descriptions of tone over actual tonal expression. A response that says "I am speaking excitedly" but sounds flat should rank lower than one that genuinely sounds excited. Follow this process: 1. Analyze the key characteristics requested in the user's instruction 2. Evaluate how well Audio 1 performs on these characteristics 3. Evaluate how well Audio 2 performs on these characteristics 4. Compare their strengths and weaknesses 5. Decide which is better overall Avoid position bias and don't let response length influence your evaluation. After your analysis, output valid JSON with exactly two keys: 'reasoning' (your explanation of the comparison) and 'label' (a string value: '1' if the first audio is better, '2' if the second audio is better, or 'tie' if they are equally good/bad. Please use "tie" sparingly, and only when you absolutely cannot choose the winner.)
Audio Judge User Prompt	Here is the instruction for this test: {instruction.wav} Here is the first audio clip: {audio_a.wav} Here is the second audio clip: {audio_b.wav} Please analyze which of the two recordings follows the instruction better, or tie. Respond ONLY in text and output valid JSON with keys 'reasoning' and 'label' (string, '1', '2' or 'tie').
LLM Judge User Prompt	You are an evaluator of audio outputs produced by different audio-capable large language models. Your task is to compare two audio responses (Audio 1 and Audio 2) generated according to a user's instruction. Evaluate based on these criteria: 1. Semantics: Does the content fulfill the user's request accurately? Did the content of the response appropriately address the user's instruction? 2. Holistic: How good is the audio response on a holistic user experience level? Avoid position bias and don't let response length influence your evaluation. After your analysis, output valid JSON with exactly two keys: 'reasoning' (your explanation of the comparison) and 'label' (a string value: '1' if the first audio is better, '2' if the second audio is better, or 'tie' if they are equally good/bad.) The responses audios (Audio 1 and Audio 2) will be given to you as text transcripts of the response. Here is the user's input prompt: {user_prompt} Here is Audio 1 text transcript: {model_a_transcript} Here is Audio 2 text transcript: {model_b_transcript} Respond ONLY in text and output valid JSON with keys 'reasoning' and 'label' (string, '1', '2' or 'tie').

Table 9: Prompts used for Audio Judge and LLM Judge on original SPEAKBENCH and S2S-ARENA labels (Tab. 1).

E Deterministic Fusion

E.1 Fusion for SPEAKBENCH (content-first, typed ties)

We use the decision tree shown in Alg. 1 to fuse the dimension-wise predictions from each judge model to an overall prediction on SpeakBench. The motivation for the decision tree logic (summarized in Tab. 10) is to reflect the original intent of the SpeakBench dataset annotations.

E.2 Fusion for S2S-ARENA (acceptability cap, typed ties)

Motivation for the acceptability cap. In our S2S-ARENA slice, EQ is labeled *both-bad* in 55% of pairs (on N=314), so forced-winner protocols would fabricate superiority; capping overall by C/EQ acceptability prevents this artifact. The fusion algorithm is provided in Alg. 2.

E.3 Comparison to Majority Voting Fusion

Previous work similar to ours [Manakul et al., 2025] introduces a multi-aspect Audio LLM Judge which fuses dimension-wise predictions through a majority vote. We compare this method to our tree-based fusion approach in Tab. 11 and confirm that our tree-based fusion approach unanimously improves overall label accuracy by incorporating dataset intent and policy-aware decision logic.

Algorithm 1 SpeakBench Fusion

Require: $\Delta_C, \Delta_{VQ}, \Delta_{EQ} \in \{1, 2, \text{both-good}, \text{both-bad}\}$

```
if Δ_C ∈ {1, 2} then
    return Ŷ ← Δ_C
end if
if Δ_C = both-good then
    if Δ_EQ ∈ {1, 2} then
        return Ŷ ← Δ_EQ
    end if
    if Δ_VQ ∈ {1, 2} then
        return Ŷ ← Δ_VQ
    end if
    return Ŷ ← both-good
end if
if Δ_C = both-bad then
    if Δ_EQ ∈ {1, 2} then
        return Ŷ ← Δ_EQ
    end if
    if Δ_VQ ∈ {1, 2} then
        return Ŷ ← Δ_VQ
    end if
    return Ŷ ← both-bad
end if
```

F Sensitivity analysis details

Procedures. We implement three diagnostics to characterize evaluation sensitivity: (1) **Content-controlled counterfactual:** force `Content=both_good` and record whether EQ or VQ determines the overall label. (2) **One-at-a-time ablations:** replace each dimension by `both_good` and measure the fraction of rows where the fused overall changes. (3) **Correctness-aware and policy metrics:** compute winner-slice accuracy, winner-on-bad rate, and EQ/VQ decision accuracies under these perturbations. All experiments reuse the same fusion functions as the main evaluation: SPEAKBENCH uses the Content→EQ→VQ tree; S2S-ARENA applies a strict acceptability cap $\text{overall} \preceq \min(\text{Content}, \text{EQ})$ and, for diagnostic purposes, a lenient "acceptable" cap that permits winners whenever both cues are not `both_bad`.

Implementation. For each judge we compute the base fused overall label and the counterfactuals defined above. EQ/VQ decision accuracy is measured against the ground truth HCoT label whenever a dimension produces a winner in the forced-content counterfactual. Flip rates report the percentage of examples whose overall label changes after ablation. The same procedure is applied across both datasets.

Dataset	Intent/Priority	One-line Fusion Rule
SPEAKBENCH	Delivery → Content	Content decides; EQ/VQ break ties
S2S-ARENA	Instruction-following → EQ	**Cap**: if C or EQ is both-bad ⇒ overall both-bad

Table 10: **Policy-prior summary.** Deterministic, monotone fusion with dataset-specific prior.

Algorithm 2 S2S-Arena Fusion

Require: $\Delta_C, \Delta_{VQ}, \Delta_{EQ} \in \{1, 2, \text{both-good}, \text{both-bad}\}$

$\Delta_{\text{cap}} \leftarrow \text{RatingMin}(\Delta_C, \Delta_{EQ})$
if $\Delta_C \in \{1, 2\}$ **then**
  return $\hat{Y} \leftarrow \text{RatingMin}(\Delta_C, \Delta_{\text{cap}})$
end if
if $\Delta_{EQ} \in \{1, 2\}$ **then**
  return $\hat{Y} \leftarrow \text{RatingMin}(\Delta_{EQ}, \Delta_{\text{cap}})$
end if
if $\Delta_{VQ} \in \{1, 2\}$ **then**
  return $\hat{Y} \leftarrow \text{RatingMin}(\Delta_{VQ}, \Delta_{\text{cap}})$
end if
return $\hat{Y} \leftarrow \text{RatingMin}(\Delta_C, \Delta_{\text{cap}})$

Dataset	Judge	Overall
SPEAKBENCH	Audio Judge (voting)	58.0 (53.4–62.3)
	Audio Judge (tree)	61.1 (56.7–65.4)
	LLM Judge (voting)	61.2 (56.5–65.5)
	LLM Judge (tree)	62.7 (58.2–67.0)
	JSON-Judge (voting)	66.5 (62.1–70.6)
	JSON-Judge (tree)	**68.6** (64.3–72.7)
S2S-ARENA	Audio Judge (voting)	37.1 (32.3–41.9)
	Audio Judge (tree)	47.5 (42.4–52.7)
	LLM Judge (voting)	34.4 (29.3–39.8)
	LLM Judge (tree)	45.9 (40.4–51.3)
	JSON-Judge (voting)	40.4 (35.0–45.9)
	JSON-Judge (tree)	**57.0** (51.6–62.4)

Table 11: **Majority Vote Fusion vs. Tree-Based Fusion.** Tree-based fusion consistently improves overall accuracy across judges and datasets by incorporating policy-aware decision logic.

Algorithm 3 RatingMin operator

Require: $\Delta_A, \Delta_B \in \{1, 2, \text{both-good}, \text{both-bad}\}$

$$\pi_A \leftarrow \begin{cases} (1,0) \text{ if } \Delta_A = 1 \\ (0,1) \text{ if } \Delta_A = 2 \\ (1,1) \text{ if } \Delta_A = \text{both-good} \\ (0,0) \text{ if } \Delta_A = \text{both-bad} \end{cases}$$

$$\pi_B \leftarrow \begin{cases} (1,0) \text{ if } \Delta_B = 1 \\ (0,1) \text{ if } \Delta_B = 2 \\ (1,1) \text{ if } \Delta_B = \text{both-good} \\ (0,0) \text{ if } \Delta_B = \text{both-bad} \end{cases}$$

$\pi_C \leftarrow \min(\pi_A, \pi_B)$ ▷ elementwise min

$$\textbf{return } \Delta_C \leftarrow \begin{cases} 1 \text{ if } \pi_C = (1,0) \\ 2 \text{ if } \pi_C = (0,1) \\ \text{both-good if } \pi_C = (1,1) \\ \text{both-bad if } \pi_C = (0,0) \end{cases}$$

G Visual Judge Comparison

We include a visual (confusion matrix) comparing judge performance against our HCoT overall labels in Fig. 10. JSON-Judge achieves the highest recall on both datasets and the highest precision on SPEAKBENCH, indicating stronger alignment with the HCoT annotations.

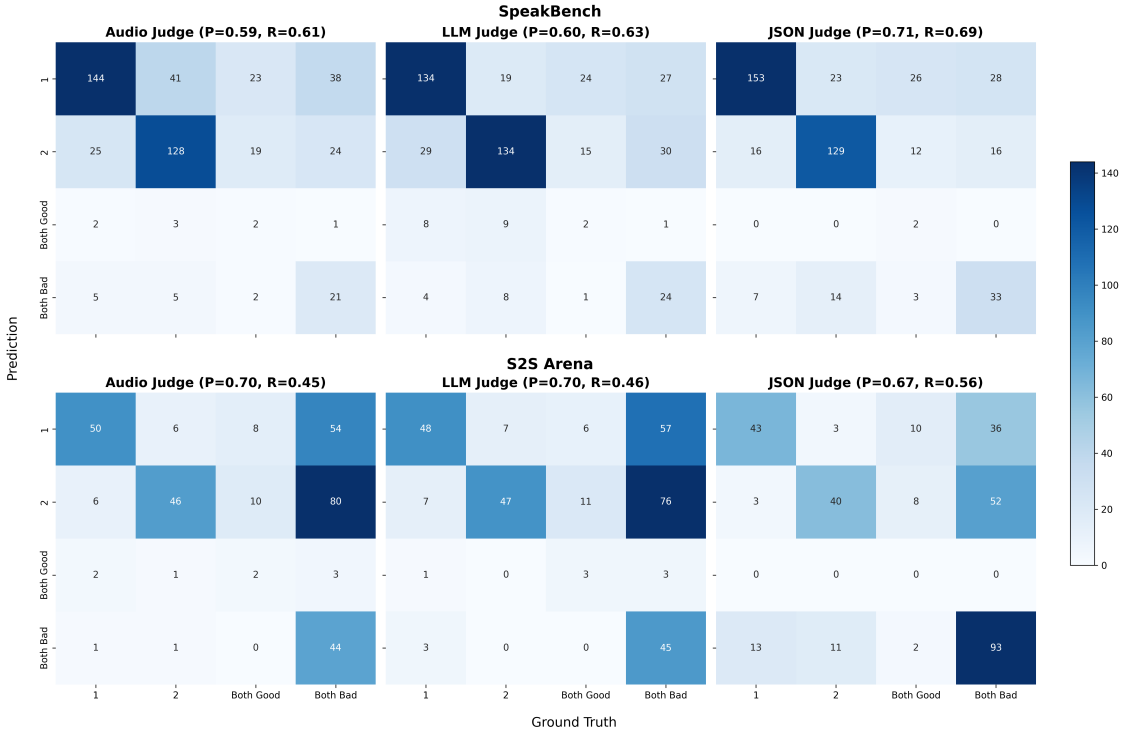

Figure 10: **Confusion Matrix Comparison.** Fused overall labels predicted by the Audio Judge, LLM Judge, and JSON-Judge against our HCoT overall labels. The underlying model is Gemini 2.5 Flash. "P" and "R" in the plot titles denote precision and recall weighted by class frequency. JSON-Judge achieves the highest recall on both datasets and the highest precision on SpeakBench, indicating stronger alignment with human HCoT annotations.