RUBRIVOX: FROM RULE-BASED REWARD MODELING TO SPOKEN DIALOGUE POLICY OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

RL (Reinforcement learning) has become the mainstream post-training paradigm for large language models, but its application in end-to-end speech dialogue models is limited by the difficulty of reward modeling. Unlike text-only assistants, spoken dialogue models must be evaluated along both semantic and paralinguistic dimensions. However, existing audio-language reward models often collapse these factors into a single holistic and opaque score, making failures hard to interpret and providing weak learning signals for downstream optimization. We propose **RubriVox**, a verifier-based RL framework for spoken dialogue models. At its core, RubriVox trains an audio-language **Reasoning Reward Model** (RubriVox-Verifier) as a structured reward verifier. RubriVox-Verifier decomposes spoken-response evaluation into semantic and paralinguistic critic rules, routes each rule to the proper evidence source for verification, and aggregates the verified rule-level judgments into a scalar reward. This design yields more interpretable and fine-grained feedback than holistic black-box reward scoring. Experiments show that when used for downstream RL, RubriVox provides a principled reward signal for aligning spoken dialogue models with user intent and vocal appropriateness. Demo could be found in `https://rubrivox.github.io/`.

1 INTRODUCTION

End-to-end spoken dialogue models can perceive speech, reason over language, and generate spoken responses within a unified framework (Ji et al., 2024; KimiTeam et al., 2025; Team et al., 2025; Wu et al., 2025). This enables more natural human–machine interaction, but also broadens the alignment problem beyond text. In spoken dialogue, response quality depends not only on what is said, but also on how it is spoken: a useful response should be semantically correct, relevant, and complete, while also being delivered with appropriate prosody, affect, and vocal style.

This expanded notion of quality makes spoken dialogue a natural yet challenging setting for reinforcement learning. Supervised fine-tuning mainly teaches a model to imitate reference responses, whereas reinforcement learning can optimize open-ended preferences over helpfulness, instruction following, emotional appropriateness, and acoustic realization (Ouyang et al., 2022; Bai et al., 2022). The effectiveness of RL, however, critically depends on the reward signal. In spoken dialogue, reward design is difficult because semantic and paralinguistic qualities can fail independently and require different evidence: a response may be factually correct but emotionally inappropriate, or sound natural while being incomplete or irrelevant.

Existing spoken reward sources only partially address this challenge. One route is to use a strong external audio-language judge to assign either a holistic score or separate semantic and acoustic scores. This avoids training a reward model, but the supervision remains judge-dependent, weakly localized, and unstable across speech understanding, audio understanding, instruction following, and voice-related tasks (Chen et al., 2024; Zhan et al., 2025). Recent evidence further suggests that such judges align less reliably with human ratings on acoustic and paralinguistic quality than on semantic quality (Chen et al., 2026). Another route is to train a compact reward model for efficient downstream RL. Such models are attractive for policy optimization, but efficiency alone does not solve spoken-response evaluation: compact reward models may lack the factual grounding, reasoning ability, and acoustic robustness needed to judge open-domain spoken interactions (Zheng et al., 2023; Lambert et al., 2024; Lin et al., 2024).

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

These limitations suggest a spoken reward model should not be a black-box scorer. Although RL requires a scalar objective, the reward should be constructed through an explicit evaluation process before aggregation. In particular, the model should decompose spoken response quality into semantic and paralinguistic criteria, route each criterion to an appropriate evidence source, and aggregate criterion-level judgments into an optimization-ready reward. This structure provides more interpretable failure diagnosis and precise credit assignment than holistic or weakly factorized scoring.

We propose **RubriVox**, a verifier-guided reinforcement learning framework for spoken dialogue models. Its core component, **RubriVox-Verifier**, is a generative audio-language reward verifier that decomposes each spoken interaction into a compact set of weighted atomic rules. These rules cover both semantic requirements, such as factual correctness, completeness, reasoning, and instruction following, and paralinguistic requirements, such as emotion, speaking rate, loudness, and vocal style. RubriVox-Verifier resolves each rule through an evidence-routed mechanism: direct judgment for internally decidable criteria, acoustic tools for objective speech properties, and external multimodal verification for knowledge-heavy or nuanced judgments. The final reward remains scalar for GRPO, but is computed from explicit rule-level scores and is therefore decomposable and traceable.

As shown in Figure 1, the RubriVox-Verifier is used as the reward source for downstream spoken-policy GRPO (Shao et al., 2024). To obtain this verifier, RubriVox introduces the contrastive rule-search and verifier-optimization pipeline shown in Figure 2. We construct spoken verifier data through speech-suitable prompt filtering, difficulty-aware response selection, and Monte Carlo rule-tree search, retaining prompts with meaningful positive–negative response variation and rules that distinguish better responses from worse ones. We then optimize RubriVox-Verifier with contrastive verifier GRPO, where each verifier rollout evaluates positive and negative spoken responses and receives rewards for format validity, label calibration, rule stability, and contrastive discrimination.

Experiments evaluate RubriVox from both reward-model and downstream policy perspectives. RubriVox-Verifier improves semantic and paralinguistic reward discrimination over generic audio-language judging and direct-scoring ablations. In downstream reinforcement learning, RubriVox-Verifier rule-level rewards outperform Gemini semantic–paralinguistic factorized rewards, which in turn outperform Gemini holistic rewards. These results show that the benefit comes from producing scalar rewards through structured, evidence-routed, and contrastively trained verification rather than through holistic or coarse judge scoring. Our contributions are summarized as follows:

1. We develop a **contrastive spoken-verifier data pipeline** that combines speech-suitable prompt filtering, difficulty-aware response selection, and Contrastive rule-tree search to construct discriminative rule-level verifier traces.
2. We propose **RubriVox-Verifier** and **contrastive verifier GRPO**. RubriVox-Verifier produces evidence-routed rule-level reward traces, while contrastive verifier GRPO optimizes verifier rollouts with format, calibration, stability, and discrimination rewards.
3. We build **RubriVox**, a full verifier-guided RL framework for spoken dialogue models, and show that structured RubriVox-Verifier rewards outperform Gemini factorized and holistic rewards in downstream policy optimization.

2 Related Work

Recent research adapts preference alignment and reward modeling to spoken dialogue to capture multi-turn dynamics. Works like ORISE focus on real-time conversational behaviors such as turn-taking and interruption handling, while WavAlign (Chen et al., 2026) addresses post-training policy stability when sparse preference rewards are applied to dense speech generation. To evaluate semantic, emotional, or acoustic quality, speech-aware reward or judge models have been explored, including WavReward (Ji et al., 2025), UniSRM (Wang et al., 2026b), SpeechLLM-as-Judges (Wang et al., 2026a), and GSRM (Shen et al., 2026). These works show that spoken dialogue rewards must go beyond text correctness; however, they either emphasize policy-side optimization, produce holistic or coarsely factorized scores, or focus on a specific speech-quality dimension. Concurrently, to overcome the weak interpretability and ambiguous credit assignment of conventional single-scalar reward models, verifier and rubric-based methods—including process reward models (Lightman et al., 2023), ThinkPRM (Khalifa et al., 2025), Rubrics as Rewards (Gunjal et al., 2025), and Open

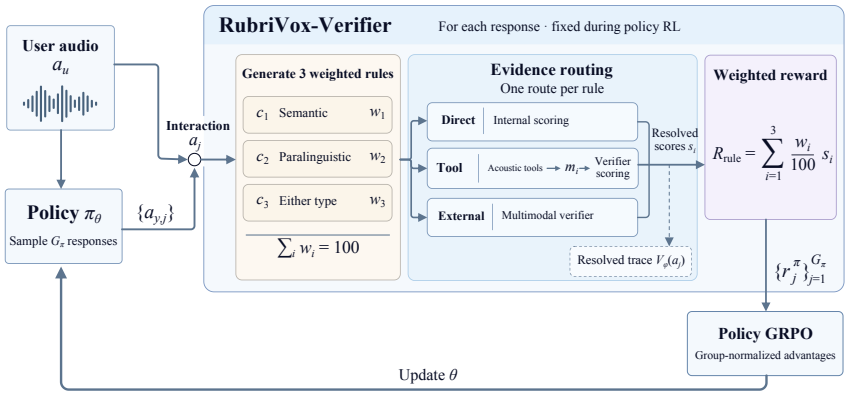


Figure 1: Evidence-routed policy optimization in RubriVox. The trained RubriVox-Verifier provides weighted rule-level rewards for downstream spoken-policy GRPO.

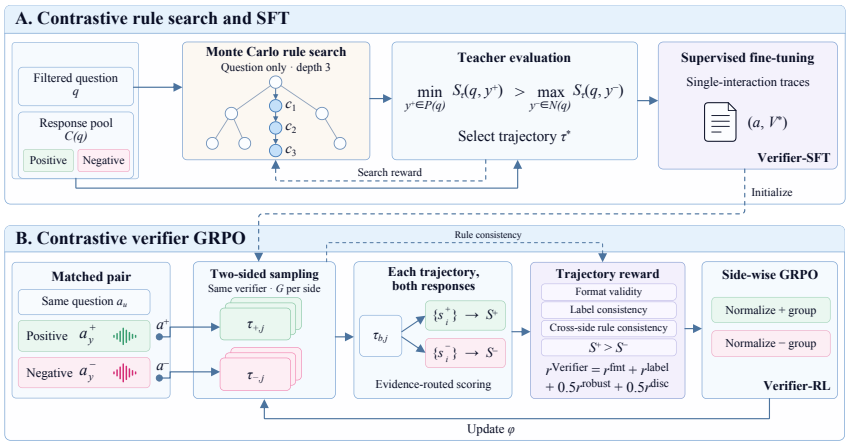


Figure 2: Contrastive training of RubriVox-Verifier. The framework builds spoken-response pools, searches rule trajectories for verifier SFT, and optimizes the verifier with contrastive GRPO.

Rubrics system(Jia et al., 2026)—evaluate responses through explicit reasoning steps or natural-language criteria. Nevertheless, these methods are confined to text tasks, relying solely on linguistic evidence. **In contrast, RubriVox studies reward architecture design, extending verifier-style modeling to speech-to-speech dialogue.** It decomposes responses into semantic and paralinguistic rules and routes them to apt evidence: semantic content is judged from language, measurable traits require acoustic tools, and nuanced style requires multimodal verification. By aggregating these scores, RubriVox jointly learns rule generation, evidence routing, and internal scoring, then uses the frozen verifier for policy optimization.

3 METHOD

3.1 PROBLEM FORMULATION AND OVERVIEW

We study reward modeling for spoken-dialogue policy optimization. Given a spoken question a_u, a policy π_θ generates a response $a_y \sim \pi_\theta(\cdot \mid a_u)$. Response quality includes both semantic content, such as correctness and instruction following, and paralinguistic delivery, such as emotion, speaking rate, and vocal style. Our objective is to learn a reward model that generates interaction-specific evaluation criteria, selects evidence routes, and produces discriminative criterion-level judgments. RubriVox models this procedure with a generative audio-language verifier π_ϕ and aggregates its judgments into a scalar policy reward.

The design addresses two requirements: criteria should be supported by appropriate evidence, and evaluation should remain consistent across responses to the same question while distinguishing their quality. Evidence routing addresses the first; contrastive rule search and verifier optimization address the second. The trained verifier then supplies rewards for spoken-policy optimization. Figures 1 and 2 summarize policy optimization and verifier training, respectively.

3.2 Evidence-Routed Reward Verification

An atomic rule is a self-contained, question-specific requirement concerning one aspect of response quality. Given the interaction $a = [a_u;\ 1\text{s silence};\ a_y]$, the verifier generates a rule trajectory $\tau = \{(c_i, t_i, w_i, \rho_i, z_i)\}_{i=1}^{K}$. Here, c_i is the criterion, $t_i \in \{\text{sem}, \text{para}\}$ its type, $w_i \geq 0$ its percentage weight, $\rho_i \in \{\text{direct}, \text{tool}, \text{external}\}$ its evidence route, and z_i its verification instruction. We use $K = 3$ rules, including at least one of each type, with $\sum_i w_i = 100$. Weights reflect the user intent and candidate response. Executing the routes produces the resolved trace

$$\mathcal{V}_\phi(a) = \{(c_i, t_i, w_i, \rho_i, z_i, s_i)\}_{i=1}^{K}, \tag{1}$$

where $s_i \in \{1, 2, 3, 4, 5\}$ measures satisfaction of c_i.

The rule type specifies what is evaluated; the route specifies how the judgment is supported. The *direct* route uses internal judgment for criteria assessable from the interaction. The *tool* route supplies acoustic measurements or estimates, including speaking and articulation rates, loudness, true peak, and emotion probabilities. The *external* route delegates knowledge-intensive semantic or nuanced multimodal judgments, including tone, prosody, style, and audio–content alignment. Scores are resolved as

$$s_i = \begin{cases} D_\phi(a, c_i, z_i), & \rho_i = \text{direct}, \\ D_\phi(a, c_i, z_i, m_i), & \rho_i = \text{tool}, \\ E(a, c_i, z_i), & \rho_i = \text{external}, \end{cases} \tag{2}$$

where D_ϕ is the verifier's internal scoring function, m_i contains tool outputs extracted from a_y, and E is an external multimodal verifier receiving audio when required. For tool-routed rules, D_ϕ interprets m_i as evidence for the final score. Appendix D details route execution.

The scalar policy reward is the weighted mean

$$R_{\text{rule}}(a_u, a_y) = \sum_{i=1}^{K} \frac{w_i}{100} s_i. \tag{3}$$

The trace records the criteria, evidence routes, instructions, and scores used to construct this reward. For type-specific analysis, we compute

$$R_h = \frac{\sum_{i:t_i=h} w_i s_i}{\sum_{i:t_i=h} w_i}, \qquad h \in \{\text{sem}, \text{para}\}, \tag{4}$$

when the corresponding total weight is positive.

3.3 Contrastive Training Data Construction

Prompt filtering and response pools. We filter public text prompts to exclude requirements for text-only artifacts, such as code, URLs, and tables. For each retained prompt q, Qwen2.5-Omni (Xu et al., 2025b) samples eight responses. Their transcripts are judged against reference labels for semantic correctness. We retain prompts with one to seven correct responses to obtain observed correctness variation. We then synthesize questions and responses using multiple TTS systems and collect outputs from spoken-dialogue models. Each resulting pool $\mathcal{C}(q) = \mathcal{P}(q) \cup \mathcal{N}(q)$ contains four spoken responses, with at least one positive and one negative response.

Contrastive rule-tree search. We select criteria according to their ability to distinguish response quality within a pool. A teacher performs depth-three Monte Carlo rule-tree search, where each path defines a three-rule trajectory τ. The search enforces coverage of both rule types and conditions successive criteria on the user request and selected prefix to reduce redundancy. Candidate responses

are hidden during criterion generation. Candidates are then revealed for evaluation through Eq. 2, with Gemini providing external multimodal verification.

Let $S_\tau(q, y)$ be the weighted score of candidate y under a fixed trajectory τ, computed from its spoken interaction using Eq. 3. A trajectory separates the pool when

$$\min_{y^+ \in \mathcal{P}(q)} S_\tau(q, y^+) > \max_{y^- \in \mathcal{N}(q)} S_\tau(q, y^-). \tag{5}$$

Successful trajectories receive positive search rewards, which are propagated to their prefixes. We retain the highest-reward trajectory and obtain resolved traces, including analyses, for all pool responses. Each record contains the text–audio interaction (q, y, a_u, a_y) and its trace. Individual records provide verifier SFT supervision; within-pool positive–negative pairs provide verifier-RL instances. The constructed data also support policy RL. Teacher criterion generation is response-blind, whereas the learned verifier generates trajectories from complete interactions. Appendix B provides construction details.

3.4 Contrastive Verifier Optimization

Supervised initialization. We initialize the verifier by fitting teacher-resolved traces. For a serialized target trace $\mathcal{V}^*$, the objective is

$$\mathcal{L}_{\text{sft}}(\phi) = -\mathbb{E}_{(a,\mathcal{V}^*)} \sum_t \log \pi_\phi(\mathcal{V}^*_t \mid \mathcal{V}^*_{<t}, a). \tag{6}$$

This stage supervises the structured output format and evaluation procedure.

Shared-trajectory evaluation. Verifier RL optimizes the evaluation procedure using matched positive–negative responses. For each triple (a_u, a_y^+, a_y^-), we sample G trajectories $\tau_{b,j}$ from each response-conditioned interaction, where $b \in \{+, -\}$ and $j = 1, \ldots, G$. Each trajectory evaluates both responses while keeping its criteria, types, weights, routes, and instructions fixed. Scores are resolved separately from each response's interaction using Eq. 2. Let $s^d_{b,j,i}$ denote rule i's resolved score on response $d \in \{+, -\}$, and let $S^d_{b,j} = \sum_i w_{b,j,i} s^d_{b,j,i}/100$. This provides a common criterion-level basis for comparing positive and negative responses.

Contrastive verifier reward. Each rollout receives

$$r^{\text{Verifier}}_{b,j} = r^{\text{fmt}}_{b,j} + r^{\text{label}}_{b,j} + 0.5\, r^{\text{robust}}_{b,j} + 0.5\, r^{\text{disc}}_{b,j}. \tag{7}$$

These terms encourage valid outputs, agreement with quality labels, cross-response criterion consistency, and preference discrimination, respectively. Below, $\mathbb{I}[\cdot]$ is the indicator function.

Format validity. The format reward is 5 when the output contains K rules, their weights and routes, and the required <score> field, and 1 otherwise.

Label-direction consistency. Let $\ell^{h,d} \in \{0, 1\}$ be response d's quality label for type h, with 1 indicating high quality. We binarize scores as $B(s) = \mathbb{I}[s \geq 4]$ and average agreement over both responses and all rules:

$$r^{\text{label}}_{b,j} = 1 + \frac{4}{2K} \sum_{d \in \{+,-\}} \sum_{i=1}^{K} \mathbb{I}\left[B(s^d_{b,j,i}) = \ell^{t_{b,j,i},d}\right]. \tag{8}$$

Semantic rules are supervised by semantic response labels, and paralinguistic rules by paralinguistic response labels.

Rule consistency. Let $C_{b,j} = \{(c_{b,j,i}, t_{b,j,i})\}_{i=1}^K$ contain the criteria and their types, and let $\bar{b}$ denote the opposite conditioning side. An external judge compares $C_{b,j}$ with all G sets generated from the opposite response:

$$r^{\text{robust}}_{b,j} = 1 + \frac{4}{G} \sum_{k=1}^{G} \text{Sim}_{\text{type}}(C_{b,j}, C_{\bar{b},k}). \tag{9}$$

Here, $\text{Sim}_{\text{type}} \in [0, 1]$ is the fraction of anchor rules with a semantically matching rule of the same type in the reference set. Each trajectory receives its own cross-response consistency reward,

encouraging agreement in criterion content and type across responses to the same question. The judge prompt appears in Appendix E.

Discrimination. The preference-discrimination reward is

$$r_{b,j}^{\text{disc}} = 1 + 4\,\mathbb{I}[S_{b,j}^{+} > S_{b,j}^{-}]. \tag{10}$$

For full-pool evaluation, this comparison is replaced by Eq. 5, with the same trajectory applied to every candidate. Thus, strict separation receives 5, and other outcomes receive 1.

Verifier GRPO update. We form a separate GRPO group for each conditioning side and compute

$$A_{b,j} = \frac{r_{b,j}^{\text{Verifier}} - \mu_b}{\sigma_b + \epsilon}, \tag{11}$$

where μ_b and σ_b are the mean and standard deviation of that side's G rollout rewards, and $\epsilon > 0$ stabilizes normalization. The clipped objective is

$$\begin{aligned}\mathcal{L}_{\text{Verifier}}(\phi) = -\mathbb{E}_{b,j,t}\Big[\min\big(&\varrho_{b,j,t}A_{b,j},\\ &\text{clip}(\varrho_{b,j,t}, 1-\epsilon_{\text{clip}}, 1+\epsilon_{\text{clip}})A_{b,j}\big)\Big],\end{aligned} \tag{12}$$

where ϵ_{clip} is the clipping parameter and

$$\varrho_{b,j,t} = \frac{\pi_{\phi}(\hat{\mathcal{Z}}_{b,j,t} \mid \mathcal{H}_{b,j,t})}{\pi_{\phi_{\text{old}}}(\hat{\mathcal{Z}}_{b,j,t} \mid \mathcal{H}_{b,j,t})}. \tag{13}$$

Here, ϕ_{old} denotes the rollout-generating parameters, $\hat{\mathcal{Z}}_{b,j,t}$ a verifier-generated token included in the loss, and $\mathcal{H}_{b,j,t}$ its execution context: the relevant interaction, preceding generated tokens, and previously returned evidence. Proposal generation uses the conditioning response; internal score resolution uses the response being evaluated. Tool outputs and external scores enter the execution record as evidence. The shared rollout advantage jointly trains rule generation, evidence routing, and internal score resolution for both responses. Training configurations appear in Appendix E.

Learned evidence routing. Evidence requirements vary across criteria and interactions. Motivated by reinforcement-learning-based tool selection (Tong et al., 2026), RubriVox explores route assignments during on-policy rollout generation. Each sampled route receives the trajectory's contrastive reward together with rule generation and internal scoring, encouraging evidence selection that supports accurate and discriminative evaluation.

3.5 Downstream Spoken-Policy Optimization

After verifier training, we freeze RubriVox-Verifier and use it as the reward source for spoken-policy GRPO. For a spoken-question distribution $\mathcal{D}$, the objective is

$$\max_{\theta} \mathbb{E}_{\substack{a_u \sim \mathcal{D} \\ a_y \sim \pi_\theta(\cdot \mid a_u)}} \left[R_{\text{rule}}(a_u, a_y)\right]. \tag{14}$$

For each question, we sample G_π responses $\hat{a}_{y,j} \sim \pi_\theta(\cdot \mid a_u)$ and compute

$$r_j^{\pi} = R_{\text{rule}}(a_u, \hat{a}_{y,j}), \qquad A_j^{\pi} = \frac{r_j^{\pi} - \mu_\pi}{\sigma_\pi + \epsilon}, \tag{15}$$

where μ_π and σ_π are the group reward mean and standard deviation. We update the policy using Eq. 12, replacing the verifier advantage with A_j^{π} and its likelihood ratio with

$$\xi_{j,t} = \frac{\pi_\theta(\hat{a}_{y,j,t} \mid \hat{a}_{y,j,<t}, a_u)}{\pi_{\theta_{\text{old}}}(\hat{a}_{y,j,t} \mid \hat{a}_{y,j,<t}, a_u)}. \tag{16}$$

Here, t indexes autoregressive response tokens and θ_{old} denotes the rollout-generating policy. With the verifier fixed, GRPO updates the policy using scalar rewards constructed from semantic and paralinguistic rule-level judgments.

Table 1: Downstream policy evaluation on VoiceBench and VStyle across reward sources and policy backbones. VoiceBench reports performance on semantic, reasoning, instruction-following, and adversarial spoken QA subsets; VStyle reports voice-style adaptation scores. Higher values indicate better performance. Best results are **bolded**. Based on the VITA-Audio backbone, we compare the main baseline, external API judges, non-rubric controls, and verifier model variants.

Model / Reward	VoiceBench									VStyle				
	Alpaca	Common	Wild	SD-QA	MMSU	OBQA	BBH	IFEval	Adv	Acoustic	Instruct	Role	Empathy	Avg
VITA-Audio backbone														
Baselines: Base & External API Judges														
VITA-Base (Long et al., 2025)	3.83	3.44	3.09	29.2	48.7	74.3	58.2	26.2	94.1	2.26	1.76	2.15	4.01	2.55
VITA + Gemini-Holistic	3.70	3.20	3.08	27.2	48.4	74.4	54.8	25.0	94.7	2.31	1.96	2.15	4.32	2.69
VITA + Gemini-Decoupled	3.92	3.48	3.10	28.9	50.8	75.6	57.2	27.8	95.4	2.45	1.95	2.22	4.61	2.81
VITA + Gemini-Rule-Dec.	4.04	3.50	3.20	29.5	49.3	74.9	57.4	29.1	97.6	2.44	2.06	2.19	4.67	2.84
Ablation 1: Non-rubric Controls														
VITA + Qwen3-Holistic-SFT	3.72	3.17	3.04	26.4	47.6	74.0	52.0	20.4	94.1	2.30	1.83	2.18	4.42	2.68
VITA + Qwen3-Decoupled-SFT	3.78	3.34	3.08	28.1	48.4	75.0	55.1	26.8	94.7	2.33	2.00	2.17	4.59	2.77
Verifier Model Variants														
VITA + Verifier-Direct-SFT	3.98	3.42	3.06	30.0	50.6	75.3	57.6	27.1	96.9	2.27	1.93	2.12	4.55	2.71
VITA + Verifier-Direct-RL	4.02	3.46	3.17	**31.3**	51.1	75.5	57.8	29.3	97.8	2.44	**2.13**	2.23	4.60	2.85
VITA + Verifier-Routed-SFT	3.96	3.47	3.17	29.4	50.8	76.2	57.6	28.2	96.2	2.32	1.87	2.12	4.64	2.73
VITA + Verifier-Routed-RL	**4.14**	**3.50**	**3.24**	31.2	**51.3**	**77.1**	**58.1**	**30.6**	**98.3**	**2.46**	2.07	**2.28**	**4.73**	**2.89**
KimiAudio backbone														
KimiAudio-Base	4.46	3.97	3.42	63.1	62.2	83.5	64.2	61.1	100.0	2.53	2.31	1.73	3.67	2.56
KimiAudio + Verifier-Routed-RL	4.52	4.10	3.54	65.0	63.8	86.1	62.6	64.3	98.5	2.64	2.46	1.98	3.82	2.73
StepAudio2 backbone														
Step-Audio-2-mini-Base	3.52	3.34	3.11	43.1	55.5	78.6	54.6	45.7	86.2	2.66	2.79	2.20	2.09	2.43
StepAudio2 + Verifier-Routed-RL	3.78	3.38	3.20	44.9	57.1	81.4	50.8	48.2	92.0	2.74	2.92	2.24	2.27	2.54

Table 2: **Comprehensive reward-model evaluation on semantic and paralinguistic preference pairs.** Best results are **bolded** and second best are underlined. We compare the main baselines, non-rubric ablations, and reward component ablations.

Model	Sem. Disc. ↑	Sem. Acc. ↑	Para. Disc. ↑	Para. Acc. ↑	Pos.-Neg. Gap ↑
Main Baselines & Models					
Baseline ALM Judge	42.7	63.7	33.1	49.2	0.792
Gemini-2.5-Pro	72.3	74.0	47.2	75.7	1.079
RubriVox-Verifier-Direct-SFT	60.7	70.0	52.5	75.4	1.045
RubriVox-Verifier-Direct-RL	71.4	72.0	58.3	76.2	2.273
RubriVox-Verifier-Routed-SFT	78.8	84.7	77.6	82.5	1.914
Ablation 1: Non-rubric Controls					
Qwen3-Omni-Holistic-SFT	54.8	64.3	50.1	70.9	1.118
Qwen3-Omni-Decoupled-SFT	61.6	66.0	54.2	73.0	1.501
Ablation 2: Verifier Reward Components					
Ours (w/o r^{robust})	82.4	87.2	76.2	80.3	1.945
Ours (w/o r^{disc})	79.5	85.4	75.1	77.0	1.709
Ours (w/o r^{robust} and r^{disc})	80.6	86.3	69.7	74.3	1.633
RubriVox-Verifier-Routed-RL	**86.3**	**92.5**	**82.4**	**84.8**	**2.447**

4 EXPERIMENTS

We evaluate RubriVox from three perspectives: reward verification, downstream policy optimization, and human-aligned rule interpretation.

4.1 EXPERIMENTAL SETUP

Reward models and baselines. We evaluate two RubriVox-Verifier variants. **RubriVox-Verifier-Routed** is the full verifier. It generates three weighted rules for each spoken user–assistant interaction, covering both semantic content and paralinguistic delivery, and resolves each rule through direct judgment, acoustic-tool evidence, or external multimodal verification. **RubriVox-Verifier-Direct** keeps the same rule format but removes evidence routing, forcing all rule scores to be predicted internally. Both variants are evaluated after SFT and after verifier-side RL.

For downstream policy RL, we compare RubriVox-Verifier with three Gemini-based reward sources. **Gemini-Holistic** assigns one overall score to the interaction. **Gemini-Decoupled** separately scores semantic and paralinguistic quality. **Gemini-Rule-Decoupled** conditions the judgment on explicit rules. These baselines distinguish holistic scoring, coarse semantic–paralinguistic factorization, rule-conditioned judging, and trained evidence-routed verification.

Training and evaluation protocol. RubriVox-Verifier is initialized from Qwen3-Omni-30B-A3B-Instruct. Downstream policies are optimized with GRPO using the selected reward source. VITA-Audio is used as the main controlled backbone; KimiAudio and StepAudio2 are included to test transfer across policy backbones. Reward models are evaluated on held-out semantic and acoustic/paralinguistic test sets. Downstream policies are evaluated on VoiceBench and VStyle. Discussions on cost/latency, data construction, tool configurations, and training hyperparameters are provided in Appendices A, B, D, and E, respectively.

Ablation studies. We conduct two ablation studies. The first ablation compares RubriVox against two non-rubric controls trained on the same Qwen3-Omni backbone and SFT data: **Qwen3-Omni-Holistic-SFT** (predicts a single scalar score) and **Qwen3-Omni-Decoupled-SFT** (predicts separate semantic and paralinguistic scores). The second ablation tests the components of the contrastive verifier GRPO objective by removing the rule-robustness reward (r^{robust}), the discriminability reward (r^{disc}), or both from the Verifier-Routed-RL pipeline.

4.2 Reward Verifier Evaluation

We first evaluate the verifier. **Semantic/paralinguistic discrimination** measures if preferred responses score higher than dispreferred ones. **Semantic/paralinguistic accuracy** reflects agreement with corresponding labels. Finally, **Pos.-Neg. Gap** measures the average reward margin between positive and negative responses.

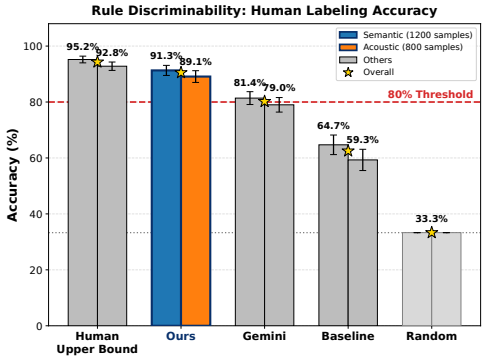
The **Main Baselines & Models** of Table 2 support three conclusions. First, **RubriVox-Verifier-Routed-RL** ranks first, raising Gemini-2.5-Pro's semantic discrimination from 72.3 to 86.3 and accuracy from 74.0 to 92.5. Second, the gain is largest on paralinguistic discrimination, indicating generic audio judges are less reliable for tone and emotion than for semantics. Third, it achieves the largest positive–negative gap, offering a stronger reward margin for policy optimization. Model variants clarify sources of improvement. Direct-RL improves over Direct-SFT, showing verifier RL goes beyond supervised imitation. However, Routed-SFT outperforms Direct-RL on most metrics, proving rule formats alone are insufficient without evidence routing: semantic criteria are judged internally, acoustic criteria benefit from tools, and multimodal criteria require external verification. Routed-RL combines this structure with contrastive optimization, yielding the best calibration.

Ablation 1 compares RubriVox against two non-rubric models trained on the same backbone and SFT data. Qwen3-Omni-Decoupled-SFT achieves 61.6 Sem. Disc. and 54.2 Para. Disc. RubriVox-Verifier-Routed-SFT achieves 78.8 Sem. Disc. and 77.6 Para. Disc. This shows rule generation and evidence routing enable evaluation beyond in-domain SFT. Ablation 2 evaluates the contrastive verifier GRPO objectives. Removing the robustness reward (r^{robust}) decreases Sem. Disc. to 82.4 and Para. Disc. to 76.2. Removing the discriminability reward (r^{disc}) reduces the Pos.-Neg. Gap to 1.709. Removing both terms reduces Para. Disc. below Routed-SFT. Results indicate both objectives are required to optimize stable and discriminative verifier traces.

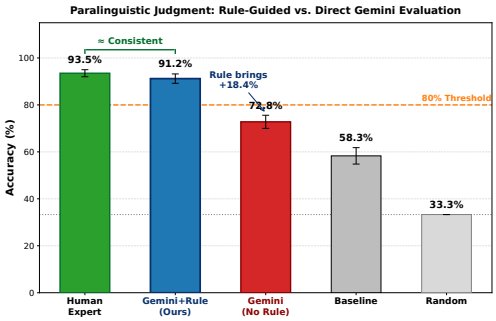
4.3 Downstream Policy Evaluation

Table 1 leads to five main findings. **Finding 1: holistic rewards are insufficient for spoken policy RL.** Gemini-Holistic is the weakest reward source in VITA-Audio, indicating that collapsing semantic, acoustic, and behavioral criteria into one score creates ambiguous signals. Gemini-Decoupled's superiority shows semantic and paralinguistic qualities should be evaluated separately.

Finding 2: rule-level feedback is more effective than coarse factorization. Gemini-Rule-Decoupled beats Gemini-Decoupled on several VoiceBench subsets, suggesting explicit criteria provide more precise credit assignment. However, **VITA + Verifier-Routed-RL** achieves the top VITA-Audio performance and highest VStyle average. This proves rules are most effective when generated, routed, and calibrated by a trained verifier instead of merely prompting external judges.



(a) Human evaluation results measuring the discriminative power of the generated rules.



(b) Comparison of paralinguistic judgment agreement among human experts and Gemini with/without explicit rules.

Finding 3: the gains are especially relevant to spoken delivery. The strongest VStyle improvements appear on Acoustic, Role, and Empathy, where evaluation depends on both response content and vocal/social appropriateness. This supports the central motivation of RubriVox: spoken dialogue alignment requires semantic–paralinguistic verification, not text-style reward scoring alone.

Finding 4: the reward transfers across policy backbones. Verifier-Routed-RL lifts VStyle averages for KimiAudio and StepAudio2 while improving most VoiceBench subsets. These non-uniform gains suggest rewards interact differently with distinct base capabilities. Nevertheless, consistent improvements indicate RubriVox-Verifier is not tied to a single policy backbone.

Finding 5: downstream improvements stem from structured verification, not generic domain adaptation. Ablation 1 shows non-rubric controls provide weak optimization signals, occasionally degrading performance below VITA-Base. This confirms RubriVox's downstream gains originate from structured, rule-guided evaluation traces rather than in-domain SFT of the verifier backbone.

4.4 Human Evaluation

Automatic metrics cannot fully determine whether rule-level speech evaluation matches human perception, particularly for paralinguistic quality. We conduct two human evaluations (Appendix F).

Rule discriminativeness. We sample 2,000 examples (1,200 semantic, 800 paralinguistic). For each, RubriVox-Verifier generates three rules, and one is randomly selected for human annotation (*positive*, *negative*, or *indistinguishable*). Figure 3a shows RubriVox-Verifier rules approach the human upper bound. This indicates that the learned verifier does not merely generate plausible rubrics; it generates criteria that help distinguish better spoken responses from worse ones.

Rule-guided paralinguistic judgment. We evaluate whether explicit rules help external judges on 1,000 paralinguistic examples, comparing human experts, Gemini with the target rule, and Gemini without rules. Figure 3b shows rule-conditioned Gemini closely approaches human experts. This result suggests that paralinguistic evaluation is difficult not only because audio perception is challenging, but also because the judge must know which aspect of delivery is relevant to the current user intent. Explicit rules reduce this ambiguity and make the judgment more stable.

5 Conclusion

We presented RubriVox, a verifier-guided reinforcement learning framework for spoken dialogue models. By decomposing spoken-response evaluation into semantic and paralinguistic rules, routing each rule to suitable evidence, and aggregating rule-level judgments into scalar rewards, RubriVox provides more interpretable and effective reward signals than holistic scoring. Experiments demonstrate improved reward discrimination and spoken-policy optimization across multiple backbones.

AI USE STATEMENT

In this work, we used generative AI tools for data processing, automated evaluation, and framework optimization as part of our core methodology. Specifically, we employed Large Language and Audio-Language Models (including Qwen2.5-Omni, Qwen3-Omni-30B, and Gemini-2.5-Pro) for speech-suitable prompt filtering, difficulty-aware response selection, contrastive rule-tree search, and external multimodal verification. Additionally, we used generative AI tools (such as ChatGPT and Claude) for minor text polishing, grammatical correction, and code-level scripting assistance during the implementation phase. We have reviewed all AI-assisted work. LLM-generated training data and evaluation rules were extensively audited through manual sampling and rigorous human evaluation (as detailed in Section 4.4) to ensure high quality, discriminability, and alignment with human judgment. The authors take full responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

To ensure the full reproducibility of the RubriVox framework, we have detailed our methodologies, data, and configurations extensively throughout the paper and its Appendix. For dataset reproducibility, Appendix B provides a comprehensive breakdown of our training data construction pipeline, including the exact sources, sample quantities, and the rigorous two-stage speech-suitability filtering rules. For model and algorithmic reproducibility, all system prompts utilized for semantic/paralinguistic rule generation, evaluation routing, and reward scoring are fully disclosed in the supplementary figures. For training reproducibility, detailed hyperparameter configurations across all three training stages (Verifier-Routed-SFT, Verifier-Routed-RL, and Policy-RL) are provided in Table 7, alongside hardware and latency profiling in Appendix A. We also provide a project website (`https://rubrivox.github.io/`) to help readers empirically verify the qualitative improvements achieved by our proposed method.

REFERENCES

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Kamile Lukosuite, Liane Lovitt, Michael Sellitto, Nelson Elhage, Nicholas Schiefer, Noemi Mercado, Nova DasSarma, Robert Lasenby, Robin Larson, Sam Ringer, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Tamera Lanham, Timothy Telleen-Lawton, Tom Conerly, Tom Henighan, Tristan Hume, Samuel R. Bowman, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, and Jared Kaplan. Constitutional ai: Harmlessness from ai feedback, 2022. URL `https://arxiv.org/abs/2212.08073`.

Yifu Chen, Shengpeng Ji, Qian Chen, Tianle Liang, Yangzhuo Li, Ziqing Wang, Wen Wang, Jingyu Lu, Haoxiao Wang, Xueyi Pu, Fan Zhuo, and Zhou Zhao. Wavalign: Enhancing intelligence and expressiveness in spoken dialogue models via adaptive hybrid post-training, 2026. URL `https://arxiv.org/abs/2604.14932`.

Yiming Chen, Xianghu Yue, Chen Zhang, Xiaoxue Gao, Robby T. Tan, and Haizhou Li. Voicebench: Benchmarking llm-based voice assistants, 2024. URL `https://arxiv.org/abs/2410.17196`.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge, 2018. URL `https://arxiv.org/abs/1803.05457`.

Anisha Gunjal, Anthony Wang, Elaine Lau, Vaskar Nath, Yunzhong He, Bing Liu, and Sean Hendryx. Rubrics as rewards: Reinforcement learning beyond verifiable domains, 2025. URL `https://arxiv.org/abs/2507.17746`.

Shengpeng Ji, Yifu Chen, Minghui Fang, Jialong Zuo, Jingyu Lu, Hanting Wang, Ziyue Jiang, Long Zhou, Shujie Liu, Xize Cheng, Xiaoda Yang, Zehan Wang, Qian Yang, Jian Li, Yidi Jiang,

Jingzhen He, Yunfei Chu, Jin Xu, and Zhou Zhao. Wavchat: A survey of spoken dialogue models, 2024. URL https://arxiv.org/abs/2411.13577.

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

Ruipeng Jia, Yunyi Yang, Yuxin Wu, Yongbo Gai, Siyuan Tao, Mengyu Zhou, Jianhe Lin, Xiaoxi Jiang, and Guanjun Jiang. Open rubric system: Scaling reinforcement learning with pairwise adaptive rubric, 2026. URL https://arxiv.org/abs/2602.14069.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William W. Cohen, and Xinghua Lu. Pubmedqa: A dataset for biomedical research question answering, 2019. URL https://arxiv.org/abs/1909.06146.

Muhammad Khalifa, Rishabh Agarwal, Lajanugen Logeswaran, Jaekyeom Kim, Hao Peng, Moontae Lee, Honglak Lee, and Lu Wang. Process reward models that think, 2025. URL https://arxiv.org/abs/2504.16828.

Tushar Khot, Peter Clark, Michal Guerquin, Peter Jansen, and Ashish Sabharwal. Qasc: A dataset for question answering via sentence composition, 2020. URL https://arxiv.org/abs/1910.11473.

KimiTeam, Ding Ding, Zeqian Ju, Yichong Leng, Songxiang Liu, Tong Liu, Zeyu Shang, Kai Shen, Wei Song, Xu Tan, Heyi Tang, Zhengtao Wang, Chu Wei, Yifei Xin, Xinran Xu, Jianwei Yu, Yutao Zhang, Xinyu Zhou, Y. Charles, Jun Chen, Yanru Chen, Yulun Du, Weiran He, Zhenxing Hu, Guokun Lai, Qingcheng Li, Yangyang Liu, Weidong Sun, Jianzhou Wang, Yuzhi Wang, Yuefeng Wu, Yuxin Wu, Dongchao Yang, Hao Yang, Ying Yang, Zhilin Yang, Aoxiong Yin, Ruibin Yuan, Yutong Zhang, and Zaida Zhou. Kimi-audio technical report, 2025. URL https://arxiv.org/abs/2504.18425.

Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi-Rui Tam, Keith Stevens, Abdullah Barhoum, Nguyen Minh Duc, Oliver Stanley, Richárd Nagyfi, Shahul ES, Sameer Suri, David Glushkov, Arnav Dantuluri, Andrew Maguire, Christoph Schuhmann, Huu Nguyen, and Alexander Mattick. Openassistant conversations – democratizing large language model alignment, 2023. URL https://arxiv.org/abs/2304.07327.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. Race: Large-scale reading comprehension dataset from examinations, 2017. URL https://arxiv.org/abs/1704.04683.

Nathan Lambert, Valentina Pyatkin, Jacob Morrison, LJ Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, Noah A. Smith, and Hannaneh Hajishirzi. Rewardbench: Evaluating reward models for language modeling, 2024. URL https://arxiv.org/abs/2403.13787.

Jia LI, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Costa Huang, Kashif Rasul, Longhui Yu, Albert Jiang, Ziju Shen, Zihan Qin, Bin Dong, Li Zhou, Yann Fleureau, Guillaume Lample, and Stanislas Polu. Numinamath. [https://huggingface.co/AI-MO/NuminaMath-CoT](https://github.com/project-numina/aimo-progress-prize/blob/main/report/numina_dataset.pdf), 2024.

Jijie Li, Li Du, Hanyu Zhao, Bo wen Zhang, Liangdong Wang, Boyan Gao, Guang Liu, and Yonghua Lin. Infinity instruct: Scaling instruction selection and synthesis to enhance language models, 2025. URL https://arxiv.org/abs/2506.11116.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step, 2023. URL https://arxiv.org/abs/2305.20050.

Zicheng Lin, Zhibin Gou, Tian Liang, Ruilin Luo, Haowei Liu, and Yujiu Yang. Criticbench: Benchmarking llms for critique-correct reasoning, 2024. URL https://arxiv.org/abs/2402.14809.

Zuwei Long, Yunhang Shen, Chaoyou Fu, Heting Gao, Lijiang Li, Peixian Chen, Mengdan Zhang, Hang Shao, Jian Li, Jinlong Peng, Haoyu Cao, Ke Li, Rongrong Ji, and Xing Sun. Vita-audio: Fast interleaved cross-modal token generation for efficient large speech-language model, 2025. URL https://arxiv.org/abs/2505.03739.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022. URL https://arxiv.org/abs/2203.02155.

Nazneen Rajani, Lewis Tunstall, Edward Beeching, Nathan Lambert, Alexander M. Rush, and Thomas Wolf. No robots. https://huggingface.co/datasets/HuggingFaceH4/no_robots, 2023.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

Maohao Shen, Tejas Jayashankar, Osama Hanna, Naoyuki Kanda, Yancheng Wang, Kateřina Žmolíková, Ruiming Xie, Niko Moritz, Anfeng Xu, Yashesh Gaur, Gregory Wornell, Qing He, and Jilong Wu. Gsrm: Generative speech reward model for speech rlhf, 2026. URL https://arxiv.org/abs/2602.13891.

Core Team, Dong Zhang, Gang Wang, Jinlong Xue, Kai Fang, Liang Zhao, Rui Ma, Shuhuai Ren, Shuo Liu, Tao Guo, Weiji Zhuang, Xin Zhang, Xingchen Song, Yihan Yan, Yongzhe He, Cici, Bowen Shen, Chengxuan Zhu, Chong Ma, Chun Chen, Heyu Chen, Jiawei Li, Lei Li, Menghang Zhu, Peidian Li, Qiying Wang, Sirui Deng, Weimin Xiong, Wenshan Huang, Wenyu Yang, Yilin Jiang, Yixin Yang, Yuanyuan Tian, Yue Ma, Yue Yu, Zihan Zhang, Zihao Yue, Bangjun Xiao, Bingquan Xia, Bofei Gao, Bowen Ye, Can Cai, Chang Liu, Chenhong He, Chunan Li, Dawei Zhu, Duo Zhang, Fengyuan Shi, Guoan Wang, Hailin Zhang, Hanglong Lv, Hanyu Li, Hao Tian, Heng Qu, Hongshen Xu, Houbin Zhang, Huaqiu Liu, Jiangshan Duo, Jianguang Zuo, Jianyu Wei, Jiebao Xiao, Jinhao Dong, Jun Shi, Junhao Hu, Kainan Bao, Kang Zhou, Linghao Zhang, Meng Chen, Nuo Chen, Peng Zhang, Qianli Chen, Qiantong Wang, Rang Li, Shaohui Liu, Shengfan Wang, Shicheng Li, Shihua Yu, Shijie Cao, Shimao Chen, Shuhao Gu, Weikun Wang, Wenhan Ma, Xiangwei Deng, Xing Yong, Xing Zhang, Xu Wang, Yifan Song, Yihao Zhao, Yingbo Zhao, Yizhao Gao, Yu Cheng, Yu Tu, Yudong Wang, Zhaojun Huang, Zhengju Tang, Zhenru Lin, Zhichao Song, Zhipeng Xu, Zhixian Zheng, and Zihan Jiang. Mimo-audio: Audio language models are few-shot learners, 2025. URL https://arxiv.org/abs/2512.23808.

Siqian Tong, Xuan Li, Yiwei Wang, Baolong Bi, Yujun Cai, Shenghua Liu, Yuchen He, and Chengpeng Hao. Autagent: A reinforcement learning framework for tool-augmented audio reasoning, 2026. URL https://arxiv.org/abs/2602.13685.

Shubham Toshniwal, Ivan Moshkov, Sean Narenthiran, Daria Gitman, Fei Jia, and Igor Gitman. Openmathinstruct-1: A 1.8 million math instruction tuning dataset, 2024. URL https://arxiv.org/abs/2402.10176.

Hui Wang, Jinghua Zhao, Yifan Yang, Shujie Liu, Junyang Chen, Yanzhe Zhang, Shiwan Zhao, Jinyu Li, Jiaming Zhou, Haoqin Sun, Yan Lu, and Yong Qin. Speechllm-as-judges: Towards general and interpretable speech quality evaluation, 2026a. URL https://arxiv.org/abs/2510.14664.

Yuanyuan Wang, Dongchao Yang, Yayue Deng, Zhiyong Wu, Yiwen Guo, Helen Meng, and Xixin Wu. Unisrm: A unified speech reward model for reasoning-based fine-grained assessment, 2026b. URL https://arxiv.org/abs/2605.23261.

Johannes Welbl, Nelson F. Liu, and Matt Gardner. Crowdsourcing multiple choice science questions, 2017. URL https://arxiv.org/abs/1707.06209.

Boyong Wu, Chao Yan, Chen Hu, Cheng Yi, Chengli Feng, Fei Tian, Feiyu Shen, Gang Yu, Haoyang Zhang, Jingbei Li, Mingrui Chen, Peng Liu, Wang You, Xiangyu Tony Zhang, Xingyuan Li, Xuerui Yang, Yayue Deng, Yechang Huang, Yuxin Li, Yuxin Zhang, Zhao You, Brian Li, Changyi Wan, Hanpeng Hu, Jiangjie Zhen, Siyu Chen, Song Yuan, Xuelin Zhang, Yimin Jiang, Yu Zhou, Yuxiang Yang, Bingxin Li, Buyun Ma, Changhe Song, Dongqing Pang, Guoqiang Hu, Haiyang Sun, Kang An, Na Wang, Shuli Gao, Wei Ji, Wen Li, Wen Sun, Xuan Wen, Yong Ren, Yuankai Ma, Yufan Lu, Bin Wang, Bo Li, Changxin Miao, Che Liu, Chen Xu, Dapeng Shi, Dingyuan Hu, Donghang Wu, Enle Liu, Guanzhe Huang, Gulin Yan, Han Zhang, Hao Nie, Haonan Jia, Hongyu Zhou, Jianjian Sun, Jiaoren Wu, Jie Wu, Jie Yang, Jin Yang, Junzhe Lin, Kaixiang Li, Lei Yang, Liying Shi, Li Zhou, Longlong Gu, Ming Li, Mingliang Li, Mingxiao Li, Nan Wu, Qi Han, Qinyuan Tan, Shaoliang Pang, Shengjie Fan, Siqi Liu, Tiancheng Cao, Wanying Lu, Wenqing He, Wuxun Xie, Xu Zhao, Xueqi Li, Yanbo Yu, Yang Yang, Yi Liu, Yifan Lu, Yilei Wang, Yuanhao Ding, Yuanwei Liang, Yuanwei Lu, Yuchu Luo, Yuhe Yin, Yumeng Zhan, Yuxiang Zhang, Zidong Yang, Zixin Zhang, Binxing Jiao, Daxin Jiang, Heung-Yeung Shum, Jiansheng Chen, Jing Li, Xiangyu Zhang, and Yibo Zhu. Step-audio 2 technical report, 2025. URL https://arxiv.org/abs/2507.16632.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, Qingwei Lin, and Daxin Jiang. Wizardlm: Empowering large pre-trained language models to follow complex instructions, 2025a. URL https://arxiv.org/abs/2304.12244.

Jin Xu, Zhifang Guo, Jinzheng He, Hangrui Hu, Ting He, Shuai Bai, Keqin Chen, Jialin Wang, Yang Fan, Kai Dang, Bin Zhang, Xiong Wang, Yunfei Chu, and Junyang Lin. Qwen2.5-omni technical report, 2025b. URL https://arxiv.org/abs/2503.20215.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. Metamath: Bootstrap your own mathematical questions for large language models. *arXiv preprint arXiv:2309.12284*, 2023.

Jun Zhan, Mingyang Han, Yuxuan Xie, Chen Wang, Dong Zhang, Kexin Huang, Haoxiang Shi, DongXiao Wang, Tengtao Song, Qinyuan Cheng, Shimin Li, Jun Song, Xipeng Qiu, and Bo Zheng. Vstyle: A benchmark for voice style adaptation with spoken instructions, 2025. URL https://arxiv.org/abs/2509.09716.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena, 2023. URL https://arxiv.org/abs/2306.05685.

A Cost and Latency Analysis

To evaluate the operational efficiency, deployment cost, and privacy implications of our proposed framework, we conduct a comprehensive profiling on a test set of 1,400 samples.

Latency under Optimized Deployment. Our training and deployment configuration utilizes an optimized inference pipeline with vLLM on NVIDIA A800 GPUs. Table 3 breaks down the latency of individual components and the theoretically expected time per sample under this optimized setting.

Table 3: vLLM-optimized latency breakdown and expected time per sample.

Profiling Item	**Latency (s)**
First-Round Qwen3-Omni Inference	4.11
Second-Round Inference (Tool-result integration)	3.89
Acoustic-Tool Call	8.50
Gemini API Call	19.74
Verifier-Direct (Complete reward per sample)	**4.11**
Verifier-Routed (Expected route-weighted per sample)	**12.44**

During actual GRPO training, evaluation is explicitly batched across the rollout dimension ($G = 4$). Due to synchronous parallel processing, the total time required to obtain the complete rewards for a full group is governed by the slowest route executed within that specific group. Table 4 reports the empirical maximum and average latencies at the group level.

Table 4: Empirical group-level batched latency ($G = 4$) during RL training.

Reward Pipeline	Max Time for a Group (s)	Avg Time for a Group (s)
Verifier-Direct-RL	6.15	4.29
Verifier-Routed-RL	26.86	17.04

Pipeline Cost and API Usage. Our dynamic multi-routing pipeline demonstrates significant cost efficiency by intelligently delegating evaluation. By routing only 24% of generated rules to the external Gemini model (with 47% routed to `direct` and 29% to `tool`), it strictly caps commercial API calls to 1,008 per 1,400 samples. In contrast, pure commercial pipelines (e.g., Gemini-Decoupled) incur massive API call volumes (up to 2,800 calls) and high input token overheads, introducing prohibitive financial costs for large-scale RL.

B Details of Data Construction

This appendix provides additional details for the speech-oriented data construction pipeline described in Section 3.3. The goal of this pipeline is to transform heterogeneous text instruction data into prompts that are suitable for spoken interaction and useful for spoken reward-verifier learning.

B.1 Source Data

To ensure broad domain coverage and instruction diversity, we aggregate raw text prompts from a wide range of open-source datasets. These sources encompass reading comprehension and dialogue (e.g., RACE (Lai et al., 2017), OASST1 (Köpf et al., 2023)), general and code-related instruction following (e.g., Infinity (Li et al., 2025), OpenAssistant(Köpf et al., 2023), No Robots (Rajani et al., 2023), WizardLM (Xu et al., 2025a)), as well as complex logic and mathematical reasoning (e.g., SlimOrca[1], MetaMathQA (Yu et al., 2023), NuminaMath-CoT (LI et al., 2024), OpenMathInstruct-1 (Toshniwal et al., 2024)). Additionally, we incorporate science question-answering data (e.g., ARC (Clark et al., 2018), PubMedQA (Jin et al., 2019), QASC (Khot et al., 2020), SciQ (Welbl et al., 2017)) and open-domain creative instructions (e.g., airoboros[2], Capybara[3], Pure-Dove). The detailed composition and retained sample counts for each source dataset, following all speech-suitability filtering stages, are comprehensively reported in Table 5.

B.2 Two-Stage Speech-Suitability Filtering

The raw text data undergoes a rigorous two-stage filtering pipeline. This process ensures the removal of interface-dependent artifacts and guarantees that all retained prompts are strictly suited for purely spoken interactions.

Stage 1: Rule-Based Heuristic Filtering. We first apply regular expressions and rule-based scripts to eliminate samples that rely on visual formatting or structural artifacts.

- **Format & Syntax Restrictions:** Complete removal of samples containing URLs, Markdown formatting (e.g., #, $\star\star$, ```), JSON structures, and programming keywords (e.g., `def`, `class`).
- **Character & Length Constraints:** Non-ASCII characters are strictly bounded below 15% (or ASCII ¿ 80%). To align with natural spoken dialogue pacing and prevent excessively long audio generation, user prompts are heavily restricted to a length of 0–250 words.

[1] https://huggingface.co/datasets/Open-Orca/SlimOrca
[2] https://huggingface.co/jondurbin/airoboros-13b
[3] https://huggingface.co/datasets/trl-lib/Capybara

Table 5: Detailed statistics of the retained prompt pool after speech-suitability filtering.

Tasks	Source	Quantity
Article	RACE	93,792
Code	Infinity	4,017
	OpenAssistant	15
Creative	airoboros-create	5,563
	Infinity	736
Dialogue	OASST1	7,159
Instruct	No Robots	1,240
	WizardLM	3,555
Logic	SlimOrca	10,456
Math	Infinity	6,938
	MetaMathQA	10,493
	OpenMathInstruct-1	744
Science QA	ARC-Challenge	2,588
	ARC-Easy	5,196
	PubMedQA	2,067
	QASC	9,059
	SciQ	13,495
Ultra	Capybara	13,406
	LessWrong-Amplify	58
	Pure-Dove	3,038
	UltraChat	18,321
Total	**–**	**211,936**

Stage 2: LLM-Guided Content Filtering and Categorization. Prompts that pass the heuristic checks undergo content-based filtering. We employ LLMs to evaluate and decouple the dataset into orthogonal task domains, preventing capability overlap. Prompts are filtered based on the following strict inclusion criteria:

- **Dialogue:** Must feature an inferable persona and context (e.g., hotel reception) with a clear conversational objective or conflict resolution.
- **Instruction Following:** Must contain objective, verifiable constraints suitable for spoken execution (e.g., specific vocabulary inclusion, exact sentence counts), completely excluding subjective style requests.
- **Logical:** Encompasses deductive puzzles, state tracking, and condition elimination. The problem must be entirely comprehensible via auditory input, strictly decoupled from numerical calculation.
- **Math:** Requires explicit mathematical formulation (e.g., algebra, probability). As a mandatory adaptation, math and logic prompts are aggressively filtered or rewritten to eliminate reliance on complex visual notation (e.g., nested fractions, spatial grids). The solving process and equations must be naturally readable.
- **Science:** Focuses on objective facts, definitions, and scientific principles with clear causal relationships.
- **Creative:** Requires imaginative generation, such as stories, poems, or monologues, subject to strict output length constraints.

Concurrently, prompts evaluating paralinguistic capabilities are filtered and categorized into *Implicit* (reacting to unstated emotional cues), *Explicit Generation* (following explicit style directives), and *Explicit Understanding* (analyzing features of the user's audio).

B.3 Audio Synthesis Configuration

To synthesize the filtered textual interactions into high-fidelity speech, we employ three distinct state-of-the-art TTS systems to ensure acoustic diversity: **CosyVoice3**, **IndexTTS 2.5**, and **CosyVoice2**.

To guide the generative style, the reference speaker characteristics and acoustic prompt audios (*prompt wavs*) are sampled directly from the **Seed-TTS-Eval** dataset. Finally, these assembled audio samples are filtered using the difficulty-aware selection criteria detailed in Section 3.3 to ensure non-trivial quality variation.

B.4 Prompt Design for Verifier Rule Generation

As established in the audio synthesis configuration, the final acoustic input provided to the verifier is strictly formatted as a concatenated sequence: the synthesized user question, a one-second silence interval, and the candidate assistant response.

To achieve the rigorous heuristic tree search described in Section 3.3, we design a suite of specialized prompts. The specific text for these prompts is provided in the supplementary figures. The purpose of each prompt is concisely outlined below:

Semantic Content Rubric Generation: Illustrated in Figure 5, this prompt instructs the teacher model to generate evaluation criteria derived directly from the user's query intent.

Paralinguistic Rubric Generation: As shown in Figure 6, this prompt guides the model to produce acoustic-specific evaluation rubrics.

Evaluation Routing and Analysis: Depicted in Figure 7, this prompt acts as an evaluation planner. For a given candidate response and a set of three rules, it assigns the most appropriate evidence route (`tool`, `direct`, or `gemini`) based on the rule's requirements and generates a fresh, first-pass qualitative analysis.

Direct Final Scoring: Presented in Figure 8, this prompt is executed exclusively for rules assigned the `direct` route. It assigns a strict quantitative score (1–5) to the candidate response.

C Extended Downstream Evaluation

To account for the inherent complexity and randomness in reinforcement learning, we conduct extended evaluations on the VITA-Audio downstream policy.

Variance Reporting. We re-ran the main downstream experiments across multiple random seeds to report the mean and standard deviation for key metrics on VoiceBench and VStyle. As shown in Table 6, RubriVox-Verifier-Routed-RL consistently maintains a stable algorithmic advantage over both the base model and Gemini-guided baselines.

Table 6: Downstream performance with variance (mean $\pm$ standard deviation) across multiple seeds on VITA-Audio.

Model	VStyle Avg ↑	Alpaca ↑	OBQA ↑	IFEval ↑	Adv ↑
VITA-Base	2.55 ± 0.02	3.84 ± 0.02	74.27 ± 0.18	26.2 ± 0.4	94.1 ± 0.5
Gemini-Holistic	2.69 ± 0.03	3.68 ± 0.06	74.52 ± 0.24	25.0 ± 0.6	94.7 ± 0.4
Gemini-Decoupled	2.81 ± 0.03	3.92 ± 0.06	75.83 ± 0.21	27.8 ± 0.5	95.4 ± 0.5
Gemini-Rule-Decoupled	2.84 ± 0.02	4.07 ± 0.05	74.90 ± 0.33	29.1 ± 0.4	97.6 ± 0.4
RubriVox-Verifier-Routed-RL	$\mathbf{2.89 \pm 0.03}$	$\mathbf{4.18 \pm 0.08}$	$\mathbf{77.09 \pm 0.37}$	$\mathbf{30.6 \pm 0.5}$	$\mathbf{98.3 \pm 0.3}$

D Reward Model Inference Pipeline

As illustrated in Figure 4, RubriVox-Verifier provides interpretable rule-level feedback. The case shows that a spoken response may be fluent and natural while still failing on the core semantic

Parameter	Stage 1 (Verifier-Routed-SFT)	Stage 2 (Verifier-Routed-RL)	Stage 3 (Policy-RL)
Initialized Model	Qwen3-Omni (30B)	Verifier-SFT Checkpoint	VITA-Audio Backbone
Trained Components	LLM & Aligner (ViT frozen)	Full Model	Full Model
Learning Rate	1e-5	1e-6	1e-6
Batch Size (per device)	4	1	1
Gradient Accumulation	2 (for 8 GPUs)	2 (for 16 GPUs)	2 (for 16 GPUs)
Epochs / Steps	2 Epochs	300 Steps	300 Steps
Optimizer	AdamW	AdamW	AdamW
Warmup Ratio	0.05	0.05	0.05
Max Sequence Length	8192	8192	8192
RL Algorithm	–	GRPO	GRPO
Sampling Temperature	–	1.0	1.0
Top-p / Top-k	–	0.99 / 50	0.99 / 50
Group Size (G)	–	4	4

Table 7: Key hyperparameters for the three training stages.

criterion. RubriVox-Verifier separates these factors by assigning different scores to delivery, answer correctness, and commonsense justification. Therefore, the final reward remains directly usable for RL, while the verification trace explains which aspect of the response should be improved.

D.1 Initial Inference

RubriVox-Verifier first processes the multimodal context to dynamically generate three evaluation rubrics, assign distinct routing strategies, and draft an initial analysis. The prompt used for this foundational generation step is presented in Figure 9.

D.2 Post-Processing and Route Execution

Following the initial generation, the pipeline parses the routing decisions and executes specific post-processing procedures for rubrics that are not resolved directly:

The `tool` Route. When the `tool` route is assigned, the pipeline invokes specialized algorithms to extract objective acoustic metrics from the spoken responses. The primary tools integrated into our framework include:

- **my-voice-analysis**[4]: This tool analyzes the temporal characteristics of the speech signal. It returns the *articulation rate* and overall *speech rate*, providing empirical measurements for rubrics evaluating the model's speaking pace, rhythm, and fluency.
- **emotion2vec**[5]: A representation model utilized for fine-grained emotion recognition. It processes the audio segment and returns a confidence probability distribution across nine distinct emotional categories (angry, disgusted, fearful, happy, neutral, sad, surprised, unknown, and other). This serves as the quantitative basis for evaluating paralinguistic tone and emotional alignment.
- **FFmpeg (EBU R128)**[6]: To accurately evaluate volume and energy, we utilize the EBU R128 standard rather than traditional RMS, as it employs K-Weighting and gating algorithms to better simulate the subjective frequency sensitivity of human hearing. This tool outputs the *Integrated Loudness* (LUFS), *Loudness Range* (LRA), and *True Peak* (dBTP), which comprehensively assess the overall perceptual loudness and dynamic range of the audio response.

The `gemini` Route. For highly complex semantic reasoning or nuanced paralinguistic alignment, the pipeline prepares to delegate the final judgment. It compiles the audio, the generated rubric, and RubriVox-Verifier's initial analytical brief into a specialized prompt (Figure 10) to invoke an external multimodal LLM (Gemini).

[4] `https://github.com/Shahabks/my-voice-analysis`

[5] `https://github.com/ddlBoJack/emotion2vec`

[6] Implemented via the standard `ebur128` audio filter in FFmpeg.

D.3 Secondary Inference: Tool Result Integration

Once the external acoustic tools complete their execution, their objective outputs must be interpreted. The pipeline initiates a second conversational turn with RubriVox-Verifier, feeding back the raw tool metrics alongside the original rubric and initial analysis. RubriVox-Verifier performs a final reasoning step to assign the definitive quantitative score. The prompt for this multi-turn tool integration is presented in Figure 11.

E Training Pipeline

The training of our framework follows a rigorous three-stage pipeline: Supervised Fine-Tuning of the Reward Model (**Verifier-Routed-SFT**), Reinforcement Learning for the Reward Model (**Verifier-Routed-RL**), and finally, Reinforcement Learning for the target Policy Model (Policy-RL).

E.1 Three-Stage Training Paradigm

- **Stage 1: Verifier-Routed-SFT.** We perform full-parameter fine-tuning on the Qwen3-Omni (30B) backbone using the `ms-swift` framework. The model is trained on a curated dataset of 46,466 samples for 2 epochs.
- **Stage 2: Verifier-Routed-RL.** To enhance the quality and robustness of the generated rubrics, we apply reinforcement learning directly to the reward model using 4,214 samples.
- **Stage 3: Policy-RL.** In the final stage, the optimized Verifier-Routed-RL acts as the surrogate human judge to train the target policy model backbone. This alignment phase utilizes 8,613 samples.

E.2 RM-RL Reward Signals

During the contrastive verifier GRPO stage (Stage 2), Figure 12 presents the prompt for the robustness reward (r^{robust}), which measures the semantic stability of generated rules across multiple sampled rollouts. The discriminability reward (r^{disc}) is computed directly from the route-resolved scores: it equals 5 when the positive response receives a strictly higher weighted aggregate score than the negative response under the same trajectory, and 1 otherwise.

E.3 Hyperparameter Configurations

Table 7 details the precise hyperparameter settings utilized across our three-stage training pipeline.

F Human Evaluation Setup

This section details the experimental design and sampling methodology for both studies discussed in Section 4.4.

F.1 Rule Discriminativeness (A/B Testing)

The objective of this experiment is to assess whether human annotators can accurately classify a response as positive, negative, or indistinguishable when guided solely by a single generated rule.

- **Data Sampling:** We randomly sample 2,000 contrastive pairs (consisting of a positive and a negative response for the same query) from the training distribution. This set comprises 1,200 semantic-focused examples and 800 paralinguistic-focused examples.
- **Rule Generation & Selection:** For each sampled pair, RubriVox-Verifier infers the query and generates three distinct evaluation rubrics. We randomly select exactly one rubric from this set to serve as the evaluation criterion.
- **Human Annotation:** Human annotators are provided with the selected rule and both the positive and negative audio responses (in randomized order). They are instructed to listen to the

audio and read the rule carefully. For each response, they must assign one of three labels: *Positive* (the response clearly satisfies the rule), *Negative* (the response clearly violates the rule), or *Indistinguishable* (the rule is too vague, irrelevant, or inapplicable to separate the quality of the two responses).

- **Evaluation Metrics:** The human annotations are compared against the ground-truth dataset labels to calculate the discriminative accuracy and the rate of indistinguishable cases.

F.2 Rule-Guided Paralinguistic Judgment

The objective of this second experiment is to determine whether providing an explicit, dynamically generated paralinguistic rule improves the ability of external multimodal LLMs (like Gemini) to judge voice quality, and whether this improvement bridges the gap between machine evaluation and human expert judgment.

- **Data Sampling:** We randomly sample 1,000 paralinguistic-focused examples from the training distribution.
- **Evaluation Conditions:** Each sample is evaluated under three distinct parallel conditions to judge whether the paralinguistic delivery appropriately matches the user's query context:
- **Human Expert:** A panel of five human experts is provided with the specific paralinguistic rule (e.g., instructing them to focus on speaking rate or emotion). They listen to the audio and directly judge the appropriateness of the delivery. The majority vote is taken as the final human judgment.
- **Gemini + Rule:** The Gemini model is provided with the audio response *and* the specific paralinguistic rule generated by RubriVox-Verifier. It is tasked to judge the appropriateness based strictly on that rule.
- **Gemini (No Rule):** The Gemini model is provided with only the audio response and asked to judge the paralinguistic appropriateness directly, without any explicit rule guidance.

F.3 Annotator Background and Inter-Annotator Reliability

To ensure the high quality and validity of our human evaluation results, both studies were conducted by a professional corporate data annotation team consisting of five trained personnel with specialized experience in audio-linguistic analysis.

To rigorously quantify the inter-annotator agreement among the five expert annotators, we computed the Fleiss' kappa (κ) statistic for both evaluation tasks:

- **Rule Discriminability Task:** For the task of assigning samples to the *Positive*, *Negative*, or *Indistinguishable* categories based solely on the generated rules, the inter-annotator agreement was substantial ($\kappa = 0.76$).
- **Rule-Guided Paralinguistic Judgment Task:** For the evaluation of paralinguistic appropriateness, the expert annotators also demonstrated high consistency, achieving a strong agreement score ($\kappa = 0.71$).

RubriVox-Verifier: Semantic and Paralinguistic Evaluation Case Study

Input Context (User & Assistant Audio)

User Question:
Detailed instructions. In this task, you are given two natural language statements with similar wording. You must choose the statement that makes less sense based on common sense knowledge. A separates the statements. Use first or second to indicate which sentence makes less sense. Question. It is customary to offer a drink to guests. A. It is customary to offer a knife to guests.
Assistant Response: I'd say the first statement makes less sense. You don't typically offer a knife to guests like you might offer a drink. Knives are tools usually used in the kitchen and not for welcoming guests at a gathering. What do you think?

RubriVox-Verifier Evaluation

<Phase 1: Dynamic Rubric Generation>
1. Tone and Delivery Clarity [20%] *(Category: Paralinguistic)*
The assistant's tone should be clear, confident, and appropriately paced to convey the reasoning without sounding hesitant or overly casual.
2. Correct Identification of the Nonsensical Statement [50%] *(Category: Content)*
The response must correctly identify that offering a knife to guests is less sensible than offering a drink, based on common social norms.
3. Justification with Common-Sense Reasoning [30%] *(Category: Content)*
The explanation should provide a clear, common-sense justification for why offering a knife is less appropriate, such as safety concerns or cultural norms.

<Phase 2: Multi-Routing Evaluation>
[Rule 1 Evaluation]
Routing: `tool`
Tools Used: `my-voice-analysis`, `Energy_Loudness_Detector`
Analysis: The assistant's delivery should be evaluated for clarity and confidence. The tools can assess speaking rate, energy level, and articulation to determine if the tone is appropriately paced and not hesitant or overly casual. A clear, steady pace with moderate energy would support a confident delivery.

[Rule 2 Evaluation]
Routing: `direct`
Analysis: The assistant incorrectly identifies the first statement as making less sense, when in fact it is the second statement—offering a knife to guests—that is less sensible. This is a clear factual error based on common social norms, where offering a drink is a standard hospitality practice, while offering a knife is not. The response fails to correctly apply common sense to the scenario.
Final Score: <score>1</score>

[Rule 3 Evaluation]
Routing: `direct`
Analysis: The assistant provides a justification that knives are tools used in the kitchen and not for welcoming guests, which is a relevant common-sense point. However, this reasoning is flawed because the assistant misidentifies which statement makes less sense, so the justification is applied to the wrong option. Despite the reasoning being partially valid, it does not align with the correct answer, reducing its effectiveness.
Final Score: <score>2</score>

Figure 4: Case Study of RubriVox-Verifier evaluating a semantic reasoning task. The visualization demonstrates the pipeline where the model first dynamically generates orthogonal rubrics (Red block) and then applies distinct routing strategies (Green blocks). The multi-routing mechanism flexibly adapts to each rule's specific requirement—for instance, selecting acoustic tools to measure targeted delivery traits, while utilizing direct analysis to rigorously penalize semantic reasoning errors with low scores.

Prompt: Semantic Content Rubric Generation

```
You are a rigorous English rubric writer for a Semantic Reward Model.
# Goal
Given ONE user question (a real prompt to an assistant), write ONE content-evaluation
rubric that constrains the SEMANTIC content of any reasonable assistant response.
# What is a "content rubric"
A content rubric is a single-sentence evaluation criterion that judges whether an
assistant's TEXT response is correct, relevant, complete, or well-structured. It must:
  - Be self-contained and testable from the response text alone (do NOT mention voice,
tone, prosody, speaking rate, audio).
  - Begin with "The response must ..." or "The response should ...".
  - Be specific to THIS user question (refer to its concrete subject matter, entities,
intents).
  - State a CONCRETE quality requirement, not a vague "be helpful".
# The rubric MUST cover ONE specific evaluation angle
Pick one evaluation angle from the list below. Anchor the rubric to this angle:
  A. Direct factual / technical correctness
    The response must give correct, accurate facts or technically valid procedures for the
user's specific ask.
  B. Coverage / completeness of the user's sub-questions or required components
    The response must address every part of the user's request (enumerate the parts that
matter for this question), without skipping a key sub-task or constraint.
  C. Specificity / concreteness of recommendations or examples
    The response must offer concrete, named recommendations / examples / values rather than
generic platitudes (name specific tools, libraries, formulas, steps, numbers, etc., as
relevant to THIS question).
  D. Justification / reasoning quality
    The response must justify its recommendations or claims with reasoning, trade-offs, or
evidence appropriate to the user's domain.
  E. Actionability / step-by-step guidance
    The response must provide clear, actionable steps the user can follow, with the right
level of granularity for the task in this question.
  F. Scope discipline / staying on-topic
    The response must stay focused on what the user actually asked (no irrelevant tangents,
no answering a different question).
  G. Domain-appropriate depth / nuance
    The response must demonstrate domain-appropriate depth on the central concept(s) raised
by THIS question (capture the non-trivial nuances, edge cases, or pitfalls relevant to the
specific subject).
# Hard rules
- The rubric must be concretely tied to the user's actual question (reference its real
subject in your wording, e.g., "wine and food pairing application", "sleep hygiene", "Rust
vs Go", as appropriate). Do NOT write a generic rubric that could apply to any question.
- Do NOT mention paralinguistic features (tone, speed, energy, pitch, rhythm, accent,
prosody, voice, audio).
- The rubric is ONE sentence, 12-40 words, no leading number, no period required at start.
# User question (verbatim)
{ user_text }
# Output Format (Strict RAW JSON, no markdown, no code blocks, no explanations):
{  "angle": "<one capital letter from A-G>",  "rubric": "<one-sentence content rubric
tied to this user question>" }
Output JSON only.
```

Figure 5: Prompt used for Semantic Content Rubric Generation.

Prompt: Paralinguistic Rubric Generation

You are an expert audio evaluation rubric designer for a reward model.

 # Goal

TASK: Listen to the user's spoken question (provided as input_audio | this is ONLY the FIRST HALF of the interaction).

Produce EXACTLY ONE atomic, non-redundant paralinguistic evaluation rubric for grading a subsequent spoken response.

 # Task Category Context: [{task_category}]

Understand the interaction type based on the following background definitions, and tailor your rubric's focus accordingly:

- **EXPLICIT GENERATION:** The user explicitly requests the response to be delivered using a specific paralinguistic style or feature. The rubric MUST evaluate whether the model strictly obeyed this explicitly requested feature.
- **EXPLICIT UNDERSTANDING:** The user is explicitly asking the model to analyze or identify specific paralinguistic information present in the question itself. The rubric MUST evaluate whether the model successfully identified and analyzed the correct paralinguistic feature.
- **IMPLICIT TASK:** The question has strong, explicit paralinguistic undertones, but does NOT explicitly ask for a specific response style. The rubric MUST evaluate whether the model responded with high emotional intelligence, matching or appropriately reacting to the user's paralinguistic cues.
- **GENERAL TASK:** Create a standard rubric covering audio delivery quality.

 # Paralinguistic Categories

Autonomously determine the most critical acoustic dimension for this interaction and anchor your rubric to ONE of the following categories:

- **accent:** accent/dialect (e.g., British, Southern, foreign accent)
- **emotion:** emotion/mood (e.g., happy, sad, angry, excited, calm)
- **gender:** gender presentation in voice (e.g., masculine, feminine)
- **age:** age characteristics in voice (e.g., child, adult, elderly)
- **speaking_rate:** speaking rate/tempo (e.g., fast, slow, moderate)
- **pitch:** pitch/tone height (e.g., high pitch, low pitch, monotone)
- **volume:** volume/loudness (e.g., loud, soft, whisper, shouting)
- **prosody:** intonation patterns (e.g., rising/falling intonation, flat)
- **voice_quality:** timbre (e.g., breathy, creaky, clear, hoarse)
- **rhythm:** rhythm/cadence (e.g., rhythmic, staccato delivery)

 # Hard Rules

- The rubric MUST specifically target paralinguistic/acoustic delivery (e.g., emotion, tone, prosody). Do NOT evaluate semantic text content (e.g., factual accuracy, reasoning).
- The rubric must be specific, actionable, and tailored to the exact audio you just heard. Do not use generic placeholders.
- The rubric must be ONE sentence, 12-40 words.

 # Output Format (Strict RAW JSON, no markdown, no code blocks):

{ "category": "<one selected paralinguistic category from the list above>", "rubric": "<one-sentence paralinguistic rubric tied to the input audio>" }

All text MUST be in English. Output JSON only.

Figure 6: Prompt used for Paralinguistic Rubric Generation.

Prompt: Evaluation Routing and Analysis

You are a rigorous evaluation planning assistant for a Semantic Reward Model. For ONE sample, plan how each of the three rules should be evaluated, and write a fresh per-rule analysis as if YOU are the analyst.

 # Background

You will be given the user's QUESTION and the assistant's RESPONSE (asr text). For each of the three rules, independently produce {"tools", "routing", "analysis"}. For each rule, choose ONE of the following routings:

1) "tool" | Use only if the rule is paralinguistic AND its target is directly measurable by: {tool_hint}.

List 1-3 tool names from ALLOWED_TOOL_NAMES in 'tools'.

Analysis must explain (a) what acoustic property of THIS sample the rule cares about, (b) why the chosen tool measures exactly that, (c) which output of the tool should be examined to decide the score. Do NOT state a final score number.

2) "direct" | The rule's verdict on THIS sample can be stated based on plainly observable facts (clearly relevant or off-topic; obvious match/mismatch). Prefer this for content rules whose answer is decisive given the response text.

'tools' MUST be []. Write a concise English analysis stating YOUR judgement of the sample, framed as a fresh observation. Do NOT mention any prior analysis or aggregate score.

** Do NOT write <score>...</score> anywhere; a downstream script appends it.

3) "gemini" | Route to "gemini" when human-level nuanced judgement or a powerful LLM is required. This applies in two main scenarios:

- Audio/Multimodal: When acoustic/prosodic details must be heard and are not cleanly captured by available tools, or when the content rule needs nuanced multimodal judgement.
- Complex Text/Knowledge: When the response involves highly complex science, advanced mathematics, intricate logic reasoning, or strict factual verification of definitive conclusions.

'tools' MUST be []. Analysis should brief the downstream listener/evaluator: which acoustic/content cues to attend to, what specific factual/logical elements need verification, and what evidence would push the score up or down.

 # CRITICAL STYLE RULES for the analysis field

The analysis MUST read as YOUR own first-pass analysis of the sample, written purely in terms of the sample itself.

Forbidden | never use any of these patterns or paraphrases:

- "the original analysis ...", "according to the analysis ..."
- "the original/total/overall score is X"
- any meta-reference to a prior analyst, or any reference to the other two rules.

Instead:

- Write directly about the sample's content/audio.
- Use the user's QUESTION and the assistant's RESPONSE as your evidence.
- Stay strictly within the scope of THIS one rule.

 # Meta-information & Inputs

- Task Category: {task_type} - Subcategory: {subcategory} - Per-rule categories (paralinguistic / content, in order 1/2/3): {categories_block}

The three rules (in order 1/2/3): {rubrics_block}

USER QUESTION (text): {user_text}

ASSISTANT RESPONSE (text, this is what we are evaluating): {asr_text}

 # Output Format (Strict RAW JSON, no markdown, no code blocks):

{ "items": [{"index": 1, "routing": "tool|direct|gemini", "tools": [...], "analysis": "..."}, {"index": 2, "routing": "tool|direct|gemini", "tools": [...], "analysis": "..."}, {"index": 3, "routing": "tool|direct|gemini", "tools": [...], "analysis": "..."}] }

Each tools element must be {"name": "<one of allowed>", "params": {}}. Output nothing except the JSON.

Figure 7: Prompt used for Evaluation Routing and Analysis.

Prompt: Direct Final Scoring

```
You are a strict evaluator scoring ONE single evaluation rubric on ONE assistant response.
# Inputs
- USER_QUESTION: the question that was asked.
- ASSISTANT_REPLY: the assistant's response (what we are scoring).
- RULE_CATEGORY: "paralinguistic" or "content".
- RUBRIC: the single evaluation rubric you must apply.
# Evaluation Criteria
Evaluate the response directly against the provided RUBRIC. Determine the correctness,
relevance, and quality of the response independently, relying solely on the user's question
and the assistant's reply.
# Scoring Guidelines
Determine your score based strictly on the RULE_CATEGORY:
CRITICAL INSTRUCTION FOR PARALINGUISTIC RULES: COMPLETELY IGNORE the semantic correctness
or factual accuracy of the text. Focus EXCLUSIVELY on how it is spoken (tone, emotion,
clarity, pacing, fluency).
  * If evaluating a "paralinguistic" rule (1-5 scale):
    - 5 (Excellent): The voice delivery, tone, and pacing perfectly satisfy the
paralinguistic requirements.
    - 4 (Good): The delivery mostly matches the requirements with only minor acoustic or
stylistic flaws.
    - 3 (Acceptable): The delivery is understandable but generic, or partially misses the
intended paralinguistic context.
    - 2 (Poor): The delivery clearly contradicts the required tone/emotion, or suffers
from unnatural pacing/robotic traits.
    - 1 (Very Poor): The delivery completely violates the rule, or the audio is entirely
inappropriate/unintelligible.
  * If evaluating a "content" rule (1-5 scale):
    - 5 (Excellent): Accurate, deeply insightful, highly relevant, perfectly structured.
    - 4 (Good): Strong and correct, but lacks some depth or could be slightly better
structured.
    - 3 (Acceptable): Generally correct and relevant but superficial, has minor errors, or
is poorly organized.
    - 2 (Poor): Major factual errors, largely irrelevant, or logically incoherent.
    - 1 (Very Poor): Completely wrong, nonsensical, or fails to address the question at
all.
# Apply the rubric ONLY (do not score the response as a whole)
Your score must reflect how well ASSISTANT_REPLY satisfies THIS specific RUBRIC. Other
rubrics are out of scope.
# Output Format (Strict RAW JSON, no markdown, no code blocks, no explanations):
{
  "analysis":
"<3-5 sentence English analysis of how the reply satisfies this rubric, written as a fresh
first-pass observation; do NOT mention other rubrics or this scoring procedure>",
  "score": <int>
}
Output JSON only.
```

Figure 8: Prompt used for Direct Final Scoring.

Prompt: Initial Inference for RubriVox-Verifier

You are an expert Audio-Language Reward Model. Your primary task is to evaluate an AI assistant's spoken response based on the user's spoken question.

 # 1. Input Audio Structure

You will receive a single audio file containing exactly two parts, separated by approximately 1 second of silence: [User Question] -> [1s Silence] -> [Assistant Response]

 # 2. Your Core Tasks

1) Audio Comprehension: Listen to the first half of the audio to deeply understand the user's explicit request and implicit paralinguistic cues (emotion, tone, urgency, vulnerability). Then, listen to the assistant's response in the second half.

2) Dynamic Rule Generation: Dynamically generate EXACTLY THREE atomic, non-overlapping evaluation rules (rubrics) tailored to judge the assistant's response for this specific interaction.

3) Weight Allocation: Assign a percentage weight to each rule based on its importance to the user's core intent. The sum of the three weights must equal 100.

4) Evaluation & Routing: For each rule, select an evaluation routing strategy and provide a detailed analysis.

 # 3. Rule Generation Constraints

- You must output exactly 3 rules.
- Mandatory Coverage:
  - * At least ONE rule MUST be categorized as "paralinguistic".
  - * At least ONE rule MUST be categorized as "content".
  - * The third rule can be either, but must evaluate a completely new dimension.

 # 4. Routing & Analysis Constraints

For each rule, you must choose ONE of the following routing methods:

1) "tool"

- Condition: Use this routing specifically when you need to extract exact acoustic information regarding emotion, speaking rate, and energy/volume.
- Allowed Tools: Choose 1 or more from [{"name": "emotion2vec", "params": {}}, {"name": "Shahabks/my-voice-analysis", "params": {}}, {"name": "Energy_Loudness_Detector", "params": {}}].
- Analysis: Explain what specific acoustic properties of the response need to be extracted by the tools and why. Do NOT output a final score.

2) "gemini"

- Condition: Use this for complex multimodal rules where nuanced human-like judgement of audio/content alignment is needed, OR when the user's question involves overly complex rule constraints, advanced logical reasoning, or scientific problems that exceed your inherent knowledge base.
- Analysis: Describe what specific audio cues, content details, or complex logic needs to be cross-examined by the more capable model. Do NOT output a final score.

3) "direct"

- Condition: Use this when the rule can be plainly and directly judged by you right now based on the audio content.
- Analysis: Provide your detailed, first-person evaluation of the response.
- * CRITICAL: You MUST append your final integer score at the very end of your analysis string using this exact format: <score>X</score> (e.g., <score>5</score>).
- Scoring Guidelines based on rule category:
  - * "paralinguistic" rules: {paralinguistic_scoring_guidelines}
  - * "content" rules: {content_scoring_guidelines}

 # 5. Output Format (Strict RAW JSON, no markdown block formatting, no conversational text)

```
{ "rubrics": [ "1. [Short Label][X%]<Detailed rule description>", "2. [Short Label][Y%]<Detailed rule description>", "3. [Short Label][Z%]<Detailed rule description>" ], "categories": ["paralinguistic", "content", "content|paralinguistic"], "weights": [X, Y, Z], "labels": ["Short Label 1", "Short Label 2", "Short Label 3"], "finals": [ { "tools": [...], "routing": "tool|gemini|direct", "analysis": "..." }, { "tools": [...], "routing": "tool|gemini|direct", "analysis": "..." }, { "tools": [...], "routing": "tool|gemini|direct", "analysis": "..." } ] }
```

Figure 9: Prompt used for the Initial Inference of RubriVox-Verifier.

Prompt: External Delegation via the gemini Route

 # Base Context & Instructions

The provided audio contains two parts separated by a brief pause: [User Question] -> [1s Silence] -> [Assistant Response].

Please evaluate the assistant's response based strictly on the rule and instructions below.

<rule>
{rubric}
</rule>
<reference_analysis>
{analysis}
</reference_analysis>

Please use the rule and the reference analysis above to inform your judgment, then provide your own evaluation based on the following specific criteria:

 # Dynamic Evaluation Module [Injected based on Rule Category]

=== IF CATEGORY == "content" ===

Task: Evaluate the Semantic Quality of the Answer's CONTENT

Your goal is to act as a strict content evaluator. You will assess the **meaning and substance** of the spoken answer in response to the spoken question.

CRITICAL INSTRUCTION:

COMPLETELY IGNORE the acoustic quality, tone, or delivery. Focus EXCLUSIVELY on the transcribed text/meaning of the answer.

Evaluation Criteria (Content Only):

1. Accuracy & Relevance: Is the information factually correct and directly relevant to the question?
2. Depth & Completeness: Does the answer provide sufficient detail and insight?
3. Structure & Coherence: Is the argument or explanation logically structured and easy to follow?

Scoring Guidelines (1-5 for Content):

- 5 (Excellent): Accurate, deeply insightful, highly relevant, perfectly structured.
- 4 (Good): Strong and correct but lacks some depth or could be slightly better structured.
- 3 (Acceptable): Generally correct and relevant but superficial, contains minor errors, or is poorly organized.
- 2 (Poor): Major factual errors, largely irrelevant, or logically incoherent.
- 1 (Very Poor): Completely wrong, nonsensical, or fails to address the question at all.

=== IF CATEGORY == "paralinguistic" ===

Task: Evaluate the Paralinguistic Quality of the Spoken VOICE

Your goal is to judge the **paralinguistic sound quality**|clarity, fluency, accent, emotion, pacing, and overall listening comfort.

CRITICAL INSTRUCTION:

COMPLETELY IGNORE the semantic correctness or factual accuracy of the answer. Focus EXCLUSIVELY on *how* it is spoken.

Evaluation Criteria (Voice Only):

1. Clarity & Intelligibility: Is the speech clear and easy to understand?
2. Fluency & Flow: Is the delivery smooth, without awkward pauses or robotic stuttering?
3. Accent & Pronunciation: Are words pronounced naturally?
4. Emotion & Expressiveness: Does the prosody and emotional tone suit the context of the user's input?

Scoring Guidelines (1-5 for Voice):

- 5 (Excellent): Very clear, highly fluent, with emotion & prosody perfectly suited to the context.
- 4 (Good): Mostly clear and fluent with only minor, unobtrusive flaws.
- 3 (Fair): Understandable but has noticeable paralinguistic flaws (e.g., slightly robotic, minor pacing issues).
- 2 (Poor): Hard to follow due to significant clarity, pacing, or emotional mismatch problems.
- 1 (Very Poor): Largely unintelligible, highly unnatural, or extremely uncomfortable to listen to.

 # Output Format (CRITICAL for both categories)

You must output exactly in the following XML format without any markdown blocks around it:

<justification>
[Explain your score here based *only* on the criteria defined in your specific task above.]
</justification>
<score>[A single integer from 1 to 5]</score>

Figure 10: System prompt used for External Delegation via the `gemini` route. The prompt selectively applies either semantic or paralinguistic scoring guidelines based on the rule category.

Prompt: Secondary Inference for Tool Result Integration

Task: Tool-Assisted Paralinguistic Evaluation

You are an expert Audio-Language Reward Model acting as a Tool-Assisted Evaluator.

 # 1. Audio Context

The audio you are evaluating follows this structure: [User Question] -> [1s Silence] -> [Assistant Response]

 # 2. Evaluation Inputs

Please review the rule, the initial analysis (which tells you what to look for), and the objective measurements extracted by the acoustic tools.

<rule> {rubric} </rule>

<reference_analysis> {analysis} </reference_analysis>

<tool_results> {tool_results} </tool_results>

 # 3. Your Objective

Synthesize the objective metrics provided in <tool_results> with the expectations set in <reference_analysis>. Determine how accurately the assistant's paralinguistic delivery (emotion, energy, speaking rate, etc.) satisfies the <rule>.

 # 4. Scoring Guidelines (Based on Tool Metrics):

- 5 (Excellent Match): The tool metrics perfectly align with the target acoustic properties requested by the rule (e.g., exact target emotion valence, appropriate energy levels, ideal speaking rate).
- 4 (Good Match): The tool metrics strongly align, with only minor or insignificant deviations from the ideal target.
- 3 (Acceptable Match): The tool metrics show partial alignment. The acoustic delivery is not entirely wrong, but clearly lacks the precise qualities or intensity requested.
- 2 (Poor Match): The tool metrics significantly deviate from or lean contrary to the requirement (e.g., measuring high energy when a calm tone was requested).
- 1 (Complete Mismatch): The tool metrics explicitly contradict the rule, showing the exact opposite or highly inappropriate acoustics.

 # 5. Output Format (CRITICAL):

You must output exactly in the following XML format without any markdown blocks around it:

<justification>

[Explain your reasoning here. You MUST cite specific metrics/values from the <tool_results> and explain whether they support or contradict the <rule>.]

</justification>

<score>[A single integer from 1 to 5]</score>

Figure 11: Prompt used for Secondary Inference (Tool Result Integration).

Prompt: Rule Consistency Evaluation (RM-RL)

You are evaluating the semantic robustness of a specific "Anchor" evaluation rubric set against a "Reference" group of rubric sets.

You will be given:

1. **[Anchor Rubric Set]**: A set of 3 rules generated for a specific candidate trajectory.
2. **[Reference Rubric Sets]**: {G} sets of rules generated for the contrastive candidate trajectories on the opposite side.

Your task is to determine, for each rule in the Anchor Rubric Set, how many of the Reference Rubric Sets contain at least one rule that evaluates the same specific semantic or paralinguistic aspect (i.e., they are semantically similar, even if worded differently).

Inputs:

[Anchor Rubric Set]: {anchor_rubric_set}

[Reference Rubric Sets]: {reference_rubric_sets}

For each rule (Rule 1, Rule 2, Rule 3) in the Anchor set, compare it against all {G} Reference sets.

Output your answer as a JSON object with this exact format:

```
{
  "anchor_rule_1_match_count": <integer 0-{G}>,
  "anchor_rule_2_match_count": <integer 0-{G}>,
  "anchor_rule_3_match_count": <integer 0-{G}>
}
```

Where each value represents the exact number of Reference sets that contain a semantically similar rule to the respective Anchor rule.

Only output the JSON, nothing else.

Figure 12: Prompt used during the RM-RL stage to evaluate Rule Consistency.