LEARNING WHEN TO THINK WHILE LISTENING IN LARGE AUDIO-LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent advances in Large Audio-Language Models (LALMs) have made real-time, streaming spoken interaction increasingly practical. In this setting, reasoning quality and responsiveness are tightly coupled: reasoning that starts after the speech endpoint remains on the response path, while reasoning under partial audio risks missing decisive late evidence. We introduce a learnable wait-think-answer control formulation for LALMs. Motivated by the incremental nature of human conversation, the controller decides under partial audio evidence when to wait, and when to produce a compact reasoning update. Using Qwen2.5-Omni-7B as the base model, we construct aligned wait-think-answer traces from spoken reasoning data, train the controller with supervised fine-tuning (SFT), and then apply Decoupled Clip and Dynamic Sampling Policy Optimization (DAPO). The reward combines answer correctness, action validity, update timing, post-end residual work, reasoning quality, and chain consistency, optimizing the complete wait-think-answer trajectory and not the final answer alone. On a six-task synthetic spoken reasoning question answering (SRQA) benchmark, the six-reward DAPO controller improves the row-weighted accuracy from 67.6% to 70.3% while reducing post-endpoint residual reasoning from 10.44 to 8.99 tokens under the same Qwen streaming replay protocol. These tokens measure remaining explicit reasoning, not elapsed response time. Native-direct accuracy is essentially unchanged after controller post-training. On a 186-item human-recorded Real Audio Bench, a transfer check beyond text-to-speech (TTS)-rendered speech, the controller family remains functional: SFT achieves the strongest accuracy, while the six-reward DAPO controller is the only learned variant whose post-end residual falls below the base. These results show that, in the endpoint-gated Qwen setting, a streaming model can learn when to make intermediate reasoning explicit during the audio stream.

1 INTRODUCTION

Large Audio-Language Models (LALMs) are moving audio research from offline perception toward real-time spoken interaction. Recent systems process streaming audio and generate text or speech with low delay (Defossez et al., 2024; Xie and Wu, 2024; Wang et al., 2024; Xu et al., 2025a;b), making it practical to reason before an utterance ends. Answer quality and responsiveness are coupled: delaying all reasoning creates a silent response gap, while acting early can miss decisive late evidence.

Complete-audio evaluation measures final answer quality but removes the online decision: at each streaming step, the model has only partial acoustic evidence and its previously committed state. For questions that benefit from explicit reasoning, waiting until the endpoint leaves all deliberation on the response path; updating too early can commit to an incomplete interpretation. The control problem is therefore when to update reasoning and what state to retain. We study this input-side decision with the answer gated by a supplied endpoint.

Human listeners often prepare responses before a turn ends, and short response gaps require overlap between comprehension and production planning (Sacks et al., 1974; Stivers et al., 2009; Levinson and Torreira, 2015; Bögels et al., 2015; Castellucci et al., 2022; Stephens et al., 2010). Chain-of-thought and reasoning-oriented post-training likewise improve complex problem solving in text and audio models (Wei et al., 2022; DeepSeek-AI et al., 2025; Yu et al., 2025; Ma et al., 2025; Xie et al., 2025; Wu et al., 2025), while recent streaming systems ask whether models can think while listening

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

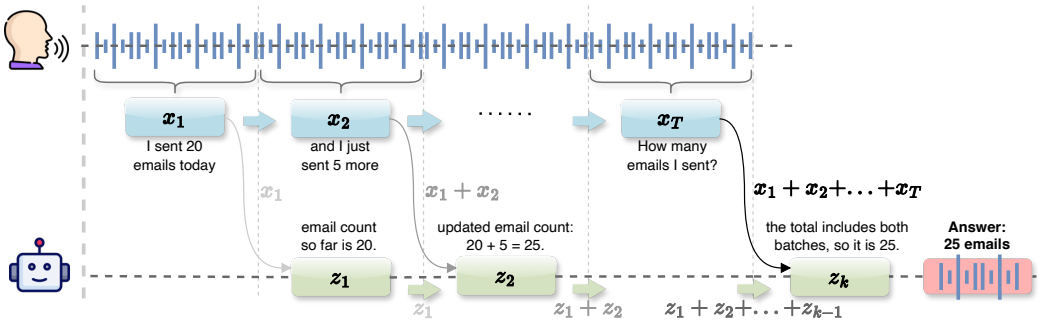

Figure 1: Wait-think-answer controller in full-prefix mode. At decision step k, the controller observes the audio prefix $x_{1:t_k}$ and the committed text memory $z_{<k}$, then emits `<wait/>`, a visible state update, or a final answer. The diagram focuses on controller-level information flow, not the internal multimodal token layout.

or interleave reasoning with speech generation (Shih et al., 2026; Chiang et al., 2026; 2025; Tong et al., 2025). We build on this direction by training a stateful wait-think-answer controller whose visible thoughts accumulate across audio windows, rather than using a single post-audio rationale or an external question-completeness trigger.

The controller uses three serialized actions: `<wait/>`, `<think>...</think>`, and `<answer>...</answer>`. A wait consumes more audio without changing visible memory; a think appends a grounded state update; and an answer commits the response. This small action space makes timing trainable and inspectable while allowing open-ended thought and answer text. Pre-endpoint updates can reduce the explicit reasoning left in the final-think state.

We instantiate the controller on Qwen2.5-Omni-7B (Xu et al., 2025a). The data pipeline generates aligned spoken wait-think-answer traces and chained streaming windows. Supervised fine-tuning (SFT) teaches the action language, after which Decoupled Clip and Dynamic Sampling Policy Optimization (DAPO) optimizes answer correctness, action validity, update timing, residual work, thought quality, and chain consistency over the complete trajectory.

On synthetic spoken reasoning, six-reward DAPO improves the matched streaming Base from 67.6% to 70.3% accuracy and reduces post-end residual tokens from 10.44 to 8.99. On 186 human recordings, the controller family remains functional beyond TTS audio, although confidence intervals overlap. The same DAPO checkpoint shows no detectable loss of native-direct accuracy. Thus the controller adds a reasoning-enabled operating mode; it does not replace direct answering when explicit reasoning is unnecessary.

The contribution is a training formulation for wait-think-answer control: a policy over partial audio, an aligned spoken-controller pipeline, and a trajectory reward for correctness, timing, residual work, and state quality. The paper studies the learning problem and controller semantics. Its validation is on endpoint-gated Qwen input streaming; optimized cache-native serving and full-duplex output control remain separate system extensions.

2 Related Work

2.1 Large Audio-Language and Omni Models

Large audio-language models (LALMs) extend text LLMs with audio encoders and cross-modal alignment, moving from task-specific audio systems toward general instruction-following models (Su et al., 2025; Yang et al., 2025). Qwen-Audio and Qwen2-Audio cover heterogeneous speech, sound, and music tasks (Chu et al., 2023; 2024). Omni models further remove the cascaded ASR–LLM–TTS boundary: Moshi uses parallel speech streams and an inner-monologue channel (Defossez et al., 2024); Mini-Omni and Freeze-Omni target low-latency speech-to-speech dialogue (Xie and Wu, 2024; Wang et al., 2024); and Qwen2.5-Omni uses a streaming Thinker–Talker architecture (Xu et al.,

2025a). We use this omni-model substrate to study how an explicit reasoning state should be updated while audio is still arriving.

2.2 Audio chain-of-thought reasoning and post-training

Chain-of-thought (CoT) can improve arithmetic, commonsense, and symbolic reasoning (Wei et al., 2022). Audio-CoT transfers this idea to LALMs, while later systems improve audio reasoning through structured data, curricula, or reward-based post-training (Ma et al., 2025; Xie et al., 2025; Wen et al., 2025; Kong et al., 2025; Wijngaard et al., 2025; Li et al., 2025; Rouditchenko et al., 2025; Tian et al., 2025). Audio-Thinker decides whether to produce a final pre-answer rationale (Wu et al., 2025). Our controller instead acts before the utterance is complete and repeatedly chooses whether to wait or expose a state update under partial evidence. SFT teaches this action language; DAPO (DeepSeek-AI et al., 2025; Yu et al., 2025) optimizes the complete trajectory for correctness, update timing, thought quality, and post-end residual work.

2.3 Thinking while listening and turn-taking

Short conversational gaps suggest that comprehension and response preparation overlap (Sacks et al., 1974; Stivers et al., 2009; Levinson and Torreira, 2015); EEG and speech-planning studies provide direct evidence that preparation can begin before a speaker finishes (Bögels et al., 2015; Magyari et al., 2017; Castellucci et al., 2022). Shih et al. translate this observation to speech LLMs: a question-completeness signal selects when CoT begins in Moshi's text-monologue stream, and preference learning then optimizes the resulting accuracy–latency trade-off (Shih et al., 2026). LTS-VoiceAgent uses a semantic trigger over streaming ASR text to orchestrate an external Thinker–Speaker pipeline (Zou et al., 2026). STITCH, SHANKS, and StreamingThinker study related simultaneous reasoning over speech or text streams (Chiang et al., 2026; 2025; Tong et al., 2025). Our policy instead consumes native audio prefixes and may emit multiple grounded state updates reusable by later decisions. Its trigger is internal to the audio model and optimized by a trajectory reward, rather than implemented as a separate ASR-text boundary classifier. Appendix D compares these controller layers explicitly.

3 Method

3.1 Architecture

We treat streaming speech reasoning as an online control problem over an explicit text memory (Figure 1). Let $x_{1:T}$ denote the spoken input, let $0 < t_1 < \cdots < t_K \leq T$ be controller decision times, and let $z_{<k}$ be the visible reasoning states already emitted by the model. We use a *full-prefix* controller observation: at decision step k, the audio input is the complete prefix $x_{1:t_k}$ heard so far, including the newest chunk and all earlier audio, while the text input is the visible reasoning state $z_{<k}$. The controller observation is

$$o_k = (x_{1:t_k}, z_{<k}), \qquad a_k = \arg\max_{a \in \mathcal{A}_k} \pi_\theta(a \mid o_k), \tag{1}$$

where o_k denotes the controller observation, a_k denotes the action taken at step k, π_θ is the controller policy with parameters θ, and $\mathcal{A}_k$ is the legal action set at that step. Equation (1) is the deterministic deployment rule. The underlying policy $\pi_\theta(\cdot \mid o_k)$ is a distribution over legal actions, and DAPO training samples from it to generate rollout groups. We omit the fixed controller instruction from the notation because it is constant across examples. Before the speech endpoint, the legal actions are WAIT and THINK. After the endpoint, the controller emits one final THINK and then an ANSWER. We serialize these actions as `<wait/>`, `<think>...</think>`, and `<answer>...</answer>`. Thus, answer timing is not a learned pre-endpoint stopping decision in our experiments; the ANSWER action is retained to train and validate the complete interaction contract. A WAIT advances the stream without changing memory. A pre-endpoint THINK appends a short semantic state update z_k, so every later controller call can condition on both the complete audio prefix and the previously committed thoughts. At the endpoint, the final THINK receives the complete audio $x_{1:T}$ and all earlier visible thoughts, and the answer turn receives the same complete audio plus the final-think state before emitting $\hat{y}$.

The benchmark endpoint is the known end of the supplied waveform; the controller does not infer it. A turn manager or voice activity detector would supply the corresponding event in deployment.

Such an event determines when the answer is permitted; it does not determine when answer-relevant computation may begin. The final THINK provides a standardized span for measuring explicit reasoning after the endpoint and may reduce to a short cue when earlier updates already support the answer. Native direct answering omits this probe and is evaluated separately. An all-wait listening trajectory remains legal, but still includes this final-think step; it is not equivalent to native direct answering. A wait also requires a controller call, even though it adds no reasoning text.

In a cache-native deployment, audio chunks would enter one persistent Qwen2.5-Omni context (Xu et al., 2025a); its KV cache would be retained and intermediate THINK text appended to the visible state. Because the available serving path does not expose this controller-style cache interface, we supply arrived audio and prior reasoning through full-prefix replay. This matches information availability, not cached computation. Appendix B and Section 6 discuss the resulting efficiency boundary.

3.2 TRAINING DATA CONSTRUCTION

Following the spoken reasoning data format of Shih et al. (2026), we first generate semantic controller traces with GPT-4o (OpenAI, 2024) and then ground them in speech. The initial candidate corpus contains 80,000 records: 40,000 verifiable items and 40,000 open-ended items. Generation is stratified by verifiability, difficulty, and topic. Appendix C gives the prompt and field schema. Each record stores a spoken surface form, a TTS style instruction, lexical anchors for answer-relevant state changes, a semantic wait-think-answer trace, and the final answer.

After validation and deduplication, the aligned corpus used for the reported experiments contains 75,723 audio-text records: 38,213 verifiable and 37,510 open-ended. The train/validation split contains 73,675 and 2,048 records, respectively. SFT uses both branches of the training split. DAPO keeps the verifiable training branch, yielding 37,180 scorable training records. To expose the reward to benchmark-style reasoning formats during training, we add a small set of training-split examples from ARC-Challenge, ARC-Easy, GSM8K, PIQA, and SocialIQA. These rows remain below 2% of the DAPO mix and never include held-out evaluation examples.

We randomly audit 200 trajectories across the topic, verifiability, and difficulty strata. The audit checks audio-prefix grounding, action order, informative intermediate updates, and whether the final state supports the answer. Failing trajectories are corrected or removed before the data are frozen; Appendix C details the filtering procedure and the 48 generation strata.

We synthesize each spoken input as a single full utterance with Qwen3-TTS (Hu et al., 2026). This preserves global prosody and avoids artificial breaks at action boundaries. Connectionist temporal classification (CTC)-style forced alignment (Graves et al., 2006) maps transcript words to timestamps, and controller boundaries are snapped upward to the 0.5s decision grid. Table 1 compares three decision grids. Their validation controller-token accuracies are nearly identical; we select 0.5s for finer control resolution, without claiming an optimal downstream accuracy–cost trade-off. In our harness, inputs shorter than 2.0s are left-padded with silence: at 0.5s, the call contains 0.5s of real audio and 1.5s of padding; at 1.5s, these lengths are 1.5s and 0.5s. Padding never exposes future speech. The grid therefore controls how often the policy reconsiders the growing prefix, while the padded input length is a separate implementation constraint.

Table 1: Decision-grid sweep: validation controller-token accuracy, not downstream SRQA accuracy.

Decision grid	0.5s	1.0s	1.5s
Validation token accuracy	0.6108	0.6105	0.6102

3.3 SUPERVISED FINE-TUNING

We first train the controller with supervised fine-tuning on Qwen2.5-Omni-7B (Xu et al., 2025a) using low-rank adaptation (LoRA) (Hu et al., 2022) in MS-Swift (Zhao et al., 2024). The supervised objective teaches the action serialization, the short semantic state style, and the distinction between ordinary waits, answer-relevant pre-endpoint thoughts, final-think compression, and the final answer. The supervised controller is trained from the base Qwen2.5-Omni-7B model for one epoch over the

aligned controller export. Training and validation curves show decreasing loss and increasing token accuracy, with both stabilizing during training. DAPO runs are initialized from supervised controllers trained with this recipe. In the reported tables, the SFT row reports this supervised controller, and the DAPO rows report four-, five-, and six-reward policy-optimization variants. Appendix B reports the training curves, hyperparameters, and compute details.

3.4 DAPO POLICY OPTIMIZATION

After SFT, we optimize the controller with a controller-specific DAPO policy-optimization loop (Yu et al., 2025). SFT can be handled directly in MS-Swift, but DAPO requires live streaming rollouts, controller-action parsing, local reward computation, dynamic resampling, and adapter updates inside the controller runtime. This requirement motivates a custom trainer instead of the generic MS-Swift Group Relative Policy Optimization (GRPO) path. DAPO belongs to the GRPO family (Shao et al., 2024; DeepSeek-AI et al., 2025). Each prompt is sampled into a group of G rollouts scored with the trajectory reward in Eq. 3. We compute group-relative advantages and apply token-level clipped updates to controller, thought, and answer tokens. Each rollout's trajectory-level advantage is shared across its generated tokens; token-level optimization does not imply a separate reward label for every action. Appendix B gives the objective, clipping settings, Kullback–Leibler (KL) regularization, and rollout-validity and dynamic-resampling details.

Training retains groups with enough format-valid rollouts, at least one valid final-think rollout, and at least one valid pre-endpoint thought. An individual all-wait trajectory remains legal, but an all-wait group is resampled. The recipe therefore encourages exposure to intermediate reasoning; it does not learn from unrestricted all-wait groups.

3.5 REWARD DESIGN

The reward targets the main failure modes of online speech reasoning. Answer correctness alone does not distinguish a wait-all policy from one that develops useful state before the endpoint. Optimizing only residual length can produce premature answers, empty thoughts, or malformed action traces. We score the full wait-think-answer trajectory so that a good rollout needs to answer correctly, follow the interaction format, update its state when the evidence changes, and keep the final post-endpoint deliberation compact. Table 2 summarizes the six reward terms, and Figure 2 illustrates how they score a wait-think-answer trajectory.

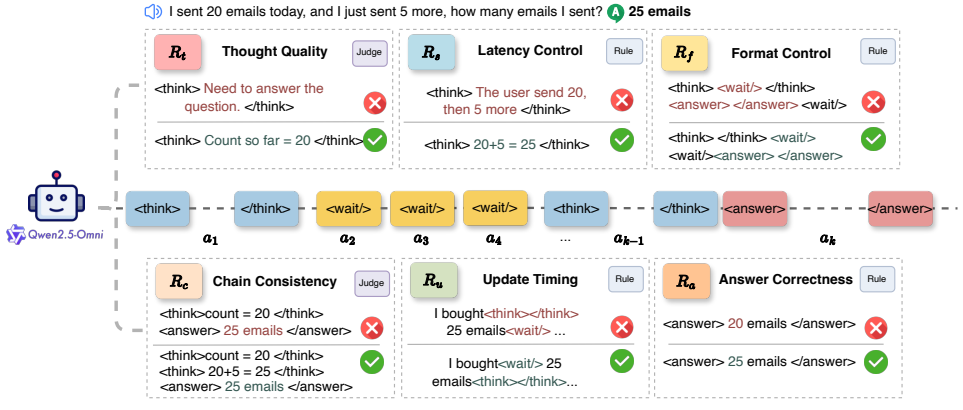

Figure 2: Trajectory reward for wait-think-answer control. "Latency Control" measures post-end tokens, not elapsed time. Other terms score action format, update timing, answer correctness, local thought quality, and chain consistency.

Table 2: Reward terms for wait-think-answer control. Each term targets a specific failure mode in the streaming trajectory, from invalid action format to unsupported final reasoning.

Term	Role	Training signal
R_a	Answer correctness	Applies deterministic task scoring first, then sends every rule-rejected output to a local semantic-equivalence judge.
R_f	Protocol validity	Enforces the wait-think-answer contract: listen-time actions are wait/think, and the answer appears only after the completed utterance.
R_s	Post-end residual work	Favors compact final-think states, so reasoning already resolved during listening is not repeated after the endpoint.
R_u	Update timing	Rewards thoughts near answer-relevant state changes and penalizes both missed updates and irrelevant interruptions.
R_t	Thought quality	Prefers short, concrete semantic states that support the answer over generic or verbose commentary.
R_c	Chain consistency	Checks whether the visible thought chain supports the final answer without unsupported jumps or contradictions.

Let τ be a completed streaming trajectory. The reward combines answer correctness, three deterministic controller terms, and two judge-assisted reasoning terms. R_a measures final answer correctness through the two-stage task scorer defined in Section 4. R_f enforces the controller format, R_s penalizes residual final-think length, and R_u rewards well-timed state updates. Following process-level audio rewards (Wu et al., 2025), R_t scores whether each thought supports the answer; R_c checks chain consistency. The shaped reward for a valid trajectory is

$$\begin{aligned} R_{\text{valid}}(\tau) = \lambda_a R_a + \lambda_f R_f + \lambda_s R_s + \lambda_u R_u + \lambda_t R_t \\ + \mathbf{1}[R_a > 0]\lambda_c R_a R_c. \end{aligned} \tag{2}$$

The final reward applies a protocol gate:

$$R(\tau) = \begin{cases} \lambda_f R_f, & R_f \le 0, \\ R_{\text{valid}}(\tau), & R_f > 0. \end{cases} \tag{3}$$

We use $\lambda_a = 1.0, \lambda_f = 1.0, \lambda_s = 1.0, \lambda_u = 3.0, \lambda_t = 1.0$, and $\lambda_c = 0.45$. The gate gives protocol validity priority over all other objectives: if the model answers before the utterance is complete, omits the required final-think/answer structure, or emits an ill-formed trace, answer correctness cannot rescue the rollout. The consistency term is also gated by $R_a > 0$, so a fluent but wrong chain does not receive extra credit simply for being self-consistent.

The ablations in Section 5 add these terms progressively. The four-term objective (R_a, R_f, R_s, R_u) establishes the basic accuracy–residual-work–timing trade-off. Adding R_t and R_c asks a stricter question: whether improving the content of the intermediate state also improves the ability of the controller to spend reasoning earlier in the stream and less after the endpoint.

4 Experiments and evaluation

Synthetic spoken reasoning question-answering benchmark. We adopt the spoken reasoning question answering (SRQA) evaluation construction introduced by Shih et al. (2026). Following their setup, text reasoning problems are rewritten into spoken questions, rendered with TTS, and then evaluated as audio inputs. Our benchmark contains 8,959 items across ARC-Easy (ARC-E), ARC-Challenge (ARC-C) (Clark et al., 2018), Physical Interaction: Question Answering (PIQA) (Bisk et al., 2020), SocialIQA (SIQA) (Sap et al., 2019), GSM8K (Cobbe et al., 2021), and a 300-item short-answer factual set (LLaMA-QS) following the LLaMA-Questions/Spectron source family (Nachmani et al., 2023). We follow their spoken-rewrite prompt lineage and use a two-stage answer scorer. Deterministic task rules run first; every rejected prediction, including irregular multiple-choice or numeric output, is passed to a local Qwen3.6-35B-A3B semantic-equivalence judge (Qwen Team, 2026). Either stage may accept the answer. Each waveform is processed directly by the native Qwen2.5-Omni audio encoder. The Thinker emits the text used for scoring and conditions the Talker speech output. We score the text directly rather than transcribing generated speech; speech fidelity is not evaluated. Qwen2.5-Omni is the model being tested, whereas the separate Qwen3.6 model judges

answer equivalence. No external ASR is inserted into the input or scoring path. Appendix A gives task counts, conversion details, examples, and an independent-judge diagnostic.

Evaluation protocols. We report three protocols. *Native direct* gives the complete waveform once and requests only an answer; it measures the checkpoint's answer-only capability. *Offline reasoning* is the complete-audio setting in the evaluation: the model receives the full spoken question once, emits one final `<think>...</think>` state, and then emits the answer. This protocol measures answer quality after complete audio observation and provides a complete-audio reasoning reference for models that do not expose a controller interface. *Deployment mode* is our streaming protocol: audio arrives on a fixed 0.5s grid, each controller call sees the full audio prefix plus previous visible thoughts, and the policy chooses `<wait/>` or `<think>...</think>` before the endpoint. At the endpoint, the controller emits one final `<think>...</think>` and then an `<answer>...</answer>`. If a controller call is still running when a later tick arrives, stale triggers can be skipped, matching the practical constraint that an online system cannot spend unbounded work at every trigger. The policy is not required to think at every tick: an all-wait pre-endpoint trajectory is legal, while the update-timing and sparsity rewards favor a state update only when newly arrived evidence makes it useful. The learned behavior therefore allocates reasoning within the streaming protocol rather than imposing continuous CoT on every query.

The endpoint is supplied by the offline benchmark: each complete recording is replayed prefix by prefix, so its last grid point is known during evaluation. Endpoint detection in live conversation would require a separate turn-taking or voice-activity mechanism. This setup isolates input-side reasoning control from that additional system decision.

Deployment mode requires direct control over the Qwen2.5-Omni inference loop (Xu et al., 2025a): prefix replay, action parsing, and persistent thought state. External baselines do not expose a matched controller interface in our harness and are therefore evaluated only offline. To separate model quality from protocol effects, Base Qwen is reported in both offline-reasoning and deployment modes. Native direct is reported separately for Base and DAPO; with no final-think span, it is an answer-only operating point rather than a latency reference for the reasoning-enabled controller.

Human-recorded real audio. We further introduce Real Audio Bench, a human-recorded streaming benchmark collected for this work. Five speakers recorded 200 GPT-4o-generated (OpenAI, 2024) candidate items designed for natural spoken delivery. Human screening removed ambiguous or unanswerable prompts and corrected answer keys, leaving 186 recordings. Scoring uses normalized rules together with the same local Qwen3.6-35B-A3B judge (Qwen Team, 2026) for cases requiring semantic matching. We treat this as a compact transfer benchmark for real human delivery, not as a substitute for a large-scale user study. The spoken items include late corrections and constraints for which an early partial interpretation can be revised as more audio arrives; representative question-and-answer entries appear in Appendix A. This set tests behavior beyond synthesized speech, while its size and speaker coverage limit conclusions about broader conversational audio.

Baselines and metrics. We compare Qwen-family controller stages: the base Qwen2.5-Omni-7B model (Xu et al., 2025a), the SFT controller, and DAPO controllers trained with four, five, or six reward terms. The base model is reported in both offline and deployment modes to isolate the effect of the streaming controller protocol itself. The SFT controller provides the supervised initialization for the DAPO policy-optimization runs. The learned controller variants are evaluated in deployment mode, which is the setting they are trained to optimize. External baselines, including Audio Flamingo 3 with and without AF-Think (Goel et al., 2025), and GLM-4-Voice-9B (Zeng et al., 2024), are evaluated only in offline mode because our benchmark harness cannot impose the same controller interface on third-party models without re-implementing their inference stacks. We also include the literature-reported Moshi rows from Shih et al. (2026) for context.

Our main latency-related metric is *post-end residual tokens*: the length of the final-think state after the user stops speaking. It measures sequential reasoning left on the response path within the same Qwen tokenizer, not seconds or hardware-independent latency. Post-end counts for Audio Flamingo and GLM use their respective tokenizers and are retained separately in Appendix D.3 as descriptive context, not ranked against Qwen. Shih et al. (2026) measure a related boundary in Moshi's native text-monologue stream; the tokenizers, generation clocks, and controller interfaces differ. Appendix B.1 reports replay-harness RTF over pre-endpoint decisions, repeated prefill, final

Table 3: Synthetic SRQA accuracy (%); Avg. is weighted by task size. Complete-audio rows use the reasoning-enabled prompt, not native direct answering. Grey Moshi rows are reported by Shih et al. (2026), not local reproductions; their waveforms and evaluation stack differ from ours. Bold/underlined mark the best/second-best Qwen streaming controllers. QC: question completeness; DPO: direct preference optimization.

Method	ARC-E	ARC-C	SIQA	PIQA	GSM8K	LLaMA-QS	Avg. ↑
Complete-audio baselines							
Qwen2.5-Omni-7B	89.6	81.1	70.9	71.3	26.7	71.0	70.8
Audio Flamingo 3	75.5	59.4	42.2	32.7	10.1	66.3	47.4
Audio Flamingo 3 + AF-Think	47.3	35.0	31.6	21.7	3.0	48.0	30.5
GLM-4-Voice-9B	64.9	48.5	41.2	22.1	6.7	63.7	40.2
Reported streaming baselines from Shih et al.							
Moshi baseline	30.2	21.5	22.8	23.8	8.7	42.8	23.4
Moshi + CoT	77.7	59.8	56.1	56.9	16.1	57.8	56.6
Moshi + CoT w/o streaming ASR	55.8	44.0	50.1	46.3	12.2	59.9	44.8
Moshi QC-SFT	62.8	43.2	45.1	40.7	13.8	56.2	44.4
Moshi QC + length-DPO	65.4	46.0	45.3	46.0	14.7	56.9	46.7
Qwen streaming controllers							
Base controller	87.8	80.8	68.6	63.5	22.8	71.0	67.6
SFT controller	86.3	78.1	68.6	60.5	21.6	**71.7**	66.1
DAPO controller (4 rewards)	88.9	**81.7**	68.4	65.1	24.6	70.3	68.5
DAPO controller (5 rewards)	89.1	**81.7**	69.6	66.4	24.9	71.0	69.2
DAPO controller (6 rewards)	**89.6**	**81.7**	**71.0**	**69.2**	**25.9**	71.0	**70.3**

reasoning, and answer generation. Base and DAPO are essentially tied on this metric; it does not establish cache-native endpoint-to-answer acceleration.

5 Results

5.1 Synthetic spoken SRQA

Table 3 reports per-task accuracy on the six-task synthetic spoken reasoning benchmark and its row-weighted average. Appendix Table 9 separately reports Qwen residual tokens and replay cost. The offline rows establish the strength of modern audio models under complete-audio evaluation. The controller rows address the question targeted by wait-think-answer post-training: with the same backbone and deployment protocol, can policy optimization improve accuracy while reducing post-end reasoning?

SFT learns the action protocol; DAPO then adds task-level trajectory rewards. The six-reward controller raises synthetic accuracy from 67.6% (Base) and 66.1% (SFT) to 70.3%, with gains on ARC-E, ARC-C, SIQA, PIQA, and GSM8K, while reducing post-end residual reasoning from 10.44 to 8.99 tokens. Relative to SFT, this is a 4.2-point gain and an 8.5% residual reduction; seven-set replay-audit RTF falls from 1.353 to 1.134 (Appendix B.1). The intermediate DAPO rows form a cumulative reward-stack ablation: adding the thought-quality term improves accuracy but does not reduce residual length, whereas the full six-term checkpoint improves both quantities. This supports the combined objective at the reported configuration, without isolating single-term causality or establishing optimal reward weights. Appendix D gives the per-task plot and conservative item-level significance bounds.

5.2 Real human-recorded audio

On Real Audio Bench, SFT improves Base by 4.8 points while remaining close in post-end residual. The five-reward DAPO row has the strongest DAPO accuracy; the six-reward row has the shortest residual and is the only learned controller below Base.

Table 4: Real Audio Bench results on 186 human-recorded items. Post-end ↓ is mean residual tokens; Δ columns compare with Base.

Method	Training signal	Acc. ↑	ΔAcc.	Post-end ↓	ΔPost-end
Base controller	–	64.0	–	6.52	–
SFT controller	SFT	**68.8**	**+4.8**	6.64	+0.12
DAPO controller (4 rewards)	$R_a + R_f + R_s + R_u$	65.6	+1.6	7.74	+1.22
DAPO controller (5 rewards)	$+R_t$	67.7	+3.7	7.39	+0.87
DAPO controller (6 rewards)	$+R_t + R_c$	65.1	+1.1	**6.33**	**-0.19**

The 95% bootstrap intervals overlap on this 186-item set (Appendix D.7), so we treat it as evidence of transfer beyond TTS speech, not a fine-grained controller ranking. The primary claim rests on the larger synthetic SRQA evaluation.

5.3 Operating-point controls and interpretation

The same checkpoints can also answer without the controller protocol. In native-direct mode, Base and DAPO answer 6,437 and 6,441 of the same 8,959 synthetic SRQA questions correctly, or 71.85% and 71.89% (paired McNemar $p = .731$). This control detects no loss of answer-only accuracy after post-training; it is not an equivalence test. Native direct remains stronger than the streaming DAPO controller's 70.3% in aggregate, so the controller is an additional reasoning-enabled operating mode, not a replacement for direct answering on every query. The complete-audio reasoning Base scores 70.8%; it receives all audio before the required final-think and is not the matched streaming reference. Appendix Table 9 reports all Qwen operating points together.

The matched streaming Base–DAPO comparison tests whether post-training improves wait/think/answer behavior under one backbone and replay protocol. Training changes both the decisions and their generated content, so the gain cannot be attributed to timing alone. Comparison to native direct instead asks when to use each mode, a selection rule we do not learn here. A gain in the streaming mode does not make explicit reasoning necessary for every query.

6 Limitations

We study input-side control on one backbone with benchmark-supplied endpoints. The evaluation does not cover endpoint detection or the full-duplex output decisions of when to speak, stop, interrupt, and resume. It also does not learn to route between native direct and controller modes. Because direct answering is stronger in aggregate, such routing matters for deployment.

Post-end tokens measure residual reasoning text, not elapsed response time. Replay RTF includes full-prefix computation; its Base–DAPO difference is negligible. Neither metric establishes cache-native endpoint-to-answer acceleration. The Moshi comparison spans different backbones and interfaces and does not replace a matched external-trigger baseline. Real Audio Bench has five speakers and 186 items; training seeds were not replicated. Matched trigger controls, wall-clock timing, broader real speech, and repeated runs would better establish deployment benefits and generality. The answer scorer and reasoning rewards use Qwen-family judges. A second Qwen judge probes judge sensitivity on a diagnostic subset (Appendix A.1), but human agreement over the full benchmark remains untested.

7 Conclusion

We train a wait/think/answer controller on native audio prefixes. In the matched Qwen streaming protocol, six-reward DAPO improves accuracy and reduces post-end residual reasoning, with no detected loss of native-direct accuracy. Human recordings test transfer beyond TTS. The action interface and trajectory reward make the timing and quality of reasoning updates jointly trainable. Cache-native timing, mode selection, and full-duplex interaction remain open.

AI USE STATEMENT

Large language models are methodological components of this work. GPT-4o was used to generate candidate training traces and benchmark items, and Qwen-family models were used for speech synthesis, reward judgments, and answer evaluation, as documented in Sections 3–5 and Appendices A–C. LLM-based assistants also supported methodological and experiment planning, coding and code inspection, result analysis, and manuscript drafting and editing. The authors made the final methodological and experimental decisions, checked the implementations, citations, analyses, and reported results, and take responsibility for the submission.

ETHICS STATEMENT

Real-time reasoning may support accessibility, translation, and tutoring, but also enable monitoring, persuasion, or voice misuse. Real Audio Bench contains scripted recordings from five consenting readers. The recordings can reveal voice identity even without names; their intended use is spoken-answer evaluation rather than voice cloning, identification, or surveillance. Appendix E documents provenance, consent, limitations, and intended use; Appendix F lists upstream asset licenses.

REPRODUCIBILITY STATEMENT

Sections 3 and 4 define the controller, training stages, and evaluation protocols. Appendix A documents task composition and scoring checks; Appendix B provides training settings, reward definitions, and runtime details; Appendix C gives data-generation and judge prompts with their input and output contracts. Additional comparisons and uncertainty analyses are in Appendix D.

REFERENCES

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. PIQA: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

Sara Bögels, Lilla Magyari, and Stephen C. Levinson. Neural signatures of response planning occur midway through an incoming question in conversation. *Scientific Reports*, 5:12881, 2015.

Gregg A. Castellucci, Christopher K. Kovach, Matthew A. Howard III, Jeremy D. W. Greenlee, and Michael A. Long. A speech planning network for interactive language use. *Nature*, 602:117–122, 2022.

Cheng-Han Chiang, Xiaofei Wang, Linjie Li, Chung-Ching Lin, Kevin Lin, Shujie Liu, Zhendong Wang, Zhengyuan Yang, Hung-yi Lee, and Lijuan Wang. SHANKS: Simultaneous hearing and thinking for spoken language models. *arXiv preprint arXiv:2510.06917*, 2025.

Cheng-Han Chiang, Xiaofei Wang, Linjie Li, Chung-Ching Lin, Kevin Lin, Shujie Liu, Zhendong Wang, Zhengyuan Yang, Hung-yi Lee, and Lijuan Wang. STITCH: Simultaneous thinking and talking with chunked reasoning for spoken language models. In *International Conference on Learning Representations*, 2026.

Yunfei Chu, Jin Xu, Xiaohuan Zhou, Qian Yang, Shiliang Zhang, Zhijie Yan, Chang Zhou, and Jingren Zhou. Qwen-Audio: Advancing universal audio understanding via unified large-scale audio-language models. *arXiv preprint arXiv:2311.07919*, 2023.

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

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. *arXiv preprint arXiv:1803.05457*, 2018.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, and others. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Alexandre Defossez, Laurent Mazare, Manu Orsini, Amelie Royer, Patrick Perez, Herve Jegou, Edouard Grave, and Neil Zeghidour. Moshi: A speech-text foundation model for real-time dialogue. *arXiv preprint arXiv:2410.00037*, 2024.

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

Alex Graves, Santiago Fernandez, Faustino Gomez, and Jurgen Schmidhuber. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the International Conference on Machine Learning*, 2006.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Hangrui Hu, Xinfa Zhu, Ting He, Dake Guo, Bin Zhang, Xiong Wang, Zhifang Guo, Ziyue Jiang, Hongkun Hao, Zishan Guo, Xinyu Zhang, Pei Zhang, Baosong Yang, Jin Xu, Jingren Zhou, and Junyang Lin. Qwen3-TTS technical report. *arXiv preprint arXiv:2601.15621*, 2026.

Zhifeng Kong, Arushi Goel, Joao Felipe Santos, Sreyan Ghosh, Rafael Valle, Wei Ping, and Bryan Catanzaro. Audio Flamingo Sound-CoT technical report: Improving chain-of-thought reasoning in sound understanding. *arXiv preprint arXiv:2508.11818*, 2025.

Stephen C. Levinson and Francisco Torreira. Timing in turn-taking and its implications for processing models of language. *Frontiers in Psychology*, 6:731, 2015.

Gang Li, Jizhong Liu, Heinrich Dinkel, Yadong Niu, Junbo Zhang, and Jian Luan. Reinforcement learning outperforms supervised fine-tuning: A case study on audio question answering. *arXiv preprint arXiv:2503.11197*, 2025.

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

Lilla Magyari, Jan Peter de Ruiter, and Stephen C. Levinson. Temporal preparation for speaking in question-answer sequences. *Frontiers in Psychology*, 8:211, 2017.

Eliya Nachmani, Alon Levkovitch, Roy Hirsch, Julian Salazar, Chulayuth Asawaroengchai, Soroosh Mariooryad, Ehud Rivlin, RJ Skerry-Ryan, and Michelle Tadmor Ramanovich. Spoken question answering and speech continuation using spectrogram-powered LLM. *arXiv preprint arXiv:2305.15255*, 2023.

OpenAI. GPT-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

Qwen Team. Qwen3.6-35B-A3B: Agentic coding power, now open to all. Hugging Face model card, 2026. URL `https://huggingface.co/Qwen/Qwen3.6-35B-A3B`.

Andrew Rouditchenko, Saurabhchand Bhati, Edson Araujo, Samuel Thomas, Hilde Kuehne, Rogerio Feris, and James Glass. Omni-R1: Do you really need audio to fine-tune your audio LLM? *arXiv preprint arXiv:2505.09439*, 2025.

Harvey Sacks, Emanuel A. Schegloff, and Gail Jefferson. A simplest systematics for the organization of turn-taking for conversation. *Language*, 50(4):696–735, 1974.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. Social IQa: Commonsense reasoning about social interactions. In *Proceedings of EMNLP-IJCNLP*, 2019.

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. *arXiv preprint arXiv:2402.03300*, 2024.

Yi-Jen Shih, Desh Raj, Chunyang Wu, Wei Zhou, SK Bong, Yashesh Gaur, Jay Mahadeokar, Ozlem Kalinli, and Michael L. Seltzer. Can speech LLMs think while listening? In *International Conference on Learning Representations*, 2026.

Greg J. Stephens, Lauren J. Silbert, and Uri Hasson. Speaker-listener neural coupling underlies successful communication. *Proceedings of the National Academy of Sciences*, 107(32):14425–14430, 2010.

Tanya Stivers, N. J. Enfield, Penelope Brown, Christina Englert, Makoto Hayashi, Trine Heinemann, Gertie Hoymann, Federico Rossano, Jan Peter de Ruiter, Kyung-Eun Yoon, and Stephen C. Levinson. Universals and cultural variation in turn-taking in conversation. *Proceedings of the National Academy of Sciences*, 106(26):10587–10592, 2009.

Yi Su, Jisheng Bai, Qisheng Xu, Kele Xu, and Yong Dou. Audio-language models for audio-centric tasks: A systematic survey. *arXiv preprint arXiv:2501.15177*, 2025.

Fei Tian, Xiangyu Tony Zhang, Yuxin Zhang, Haoyang Zhang, Yuxin Li, Daijiao Liu, Yayue Deng, Donghang Wu, Jun Chen, Liang Zhao, Chengyuan Yao, Hexin Liu, Eng Siong Chng, Xuerui Yang, Xiangyu Zhang, Daxin Jiang, and Gang Yu. Step-Audio-R1 technical report. *arXiv preprint arXiv:2511.15848*, 2025.

Junlong Tong, Yingqi Fan, Anhao Zhao, Yunpu Ma, and Xiaoyu Shen. StreamingThinker: Large language models can think while reading. *arXiv preprint arXiv:2510.17238*, 2025.

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

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, 2022.

Cheng Wen, Tingwei Guo, Shuaijiang Zhao, Wei Zou, and Xiangang Li. SARI: Structured audio reasoning via curriculum-guided reinforcement learning. *arXiv preprint arXiv:2504.15900*, 2025.

Gijs Wijngaard, Elia Formisano, Michele Esposito, and Michel Dumontier. AudSemThinker: Enhancing audio-language models through reasoning over semantics of sound. *arXiv preprint arXiv:2505.14142*, 2025.

Shu Wu, Chenxing Li, Wenfu Wang, Hao Zhang, Hualei Wang, Meng Yu, and Dong Yu. Audio-Thinker: Guiding audio language model when and how to think via reinforcement learning. *arXiv preprint arXiv:2508.08039*, 2025.

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

Zhifei Xie, Mingbao Lin, Zihang Liu, Pengcheng Wu, Shuicheng Yan, and Chunyan Miao. Audio-Reasoner: Improving reasoning capability in large audio language models. *arXiv preprint arXiv:2503.02318*, 2025.

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. *arXiv preprint arXiv:2503.20215*, 2025.

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

Chih-Kai Yang, Neo S. Ho, and Hung-yi Lee. Towards holistic evaluation of large audio-language models: A comprehensive survey. *arXiv preprint arXiv:2505.15957*, 2025.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, and others. DAPO: An open-source LLM reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

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

Yuze Zhao, Jintao Huang, Jinghan Hu, Xingjun Wang, Yunlin Mao, Daoze Zhang, Hong Zhang, Zeyinzi Jiang, Zhikai Wu, Baole Ai, Ang Wang, Wenmeng Zhou, and Yingda Chen. SWIFT: A scalable lightWeight infrastructure for fine-tuning. *arXiv preprint arXiv:2408.05517*, 2024.

Wenhao Zou, Yuwei Miao, Zhanyu Ma, Jun Xu, Jiuchong Gao, Jinghua Hao, Renqing He, and Jingwen Xu. LTS-VoiceAgent: A listen-think-speak framework for efficient streaming voice interaction via semantic triggering and incremental reasoning. *arXiv preprint arXiv:2601.19952*, 2026.

A Benchmark task and data details

The synthetic spoken SRQA benchmark contains six task families. ARC-Easy and ARC-Challenge are multiple-choice science questions. PIQA tests physical commonsense, SocialIQA tests social commonsense, GSM8K tests grade-school math, and LLaMA-QS tests short-answer factual question answering. The row counts in the reported benchmark are: 2376 ARC-Easy, 1172 ARC-Challenge, 1838 PIQA, 1954 SocialIQA, 1319 GSM8K, and 300 LLaMA-QS examples, for 8,959 total rows.

All tasks are converted into spoken prompts. Multiple-choice tasks are rewritten so that the options are naturally spoken. GSM8K questions are kept close to the original wording where possible, with spoken-friendly normalization of notation. LLaMA-QS items are evaluated as short-answer factual questions. The native-audio inference and text-scoring path is defined in Section 4.

A.1 Answer-scoring robustness

To test the robustness of the semantic fallback defined in Section 4, we re-evaluate a diagnostic subset with Qwen2.5-72B-Instruct. Table 5 reports agreement with the primary Qwen3.6-35B-A3B judge. This is a model-to-model robustness check, not a human-validity study or a complete second-model re-evaluation of all 8,959 items.

Table 5: Independent-judge diagnostic. Final decisions include deterministic task-rule acceptances before the semantic fallback.

Decision set	n	Agreement (%)	Cohen's κ
Rule-rejected semantic cases	907	97.13	0.835
Final rule-or-judge decisions	2,916	99.11	0.978

A.2 Human-recorded evaluation examples

Table 6 gives three representative entries from the 186-item Real Audio Bench described in Section 4. They illustrate the intended structure: an initially plausible state is followed by a later constraint, correction, or additional quantity that determines the answer.

Table 6: Representative Real Audio Bench entries. The table shows the spoken content and gold answer; the corresponding audio is a human recording.

Spoken question	Gold answer
I thought the shuttle runs until 10, but the driver just said the last one leaves at 9:30. When is the last shuttle?	9:30
I bought groceries for a hundred dollars, and they said there is a ten percent discount on orders over fifty, but then I found a coupon for an extra five dollars off if the total is still above seventy. What is my final bill?	85
I need to book a training session, and the trainer is available from 1 to 3, but the training room is only free from 2 to 4. When should I schedule it?	2 to 3

B Training, inference, and compute details

We summarize the main training settings for SFT and DAPO. SFT uses MS-Swift (Zhao et al., 2024) LoRA training (Hu et al., 2022); DAPO uses a custom streaming controller trainer.

Table 7: Training hyperparameters for supervised fine-tuning and DAPO controller optimization.

Setting	Supervised fine-tuning	DAPO controller optimization
Base model	Qwen2.5-Omni-7B initialized from base model weights	Qwen2.5-Omni-7B initialized from the SFT LoRA adapter
Training framework	MS-Swift 4.1.2	Custom streaming-controller trainer
Adaptation	LoRA, rank 8, alpha 32, dropout 0.05	LoRA actor, rank 8, alpha 32
Target modules	All linear layers	All linear layers
Precision	bfloat16	bfloat16
Hardware	4 NVIDIA B200 GPUs	4–5 NVIDIA B200 GPUs, depending on reward stack
Maximum length	8192 tokens	8192-token rollout context; 48-token think and answer caps
Learning rate	1×10^{-5}	4×10^{-7} actor learning rate
Batching	Per-device batch 4, gradient accumulation 2; effective batch 32 rows/update	8 rollouts
Schedule	One-epoch SFT with cosine decay and warmup	1000 steps, 50 warmup steps
Optimizer	Fused AdamW, weight decay 0.1, betas 0.9/0.95	AdamW actor update
Frozen modules	Audio encoder and aligner frozen; language model trainable	Same base frozen modules as controller LoRA training
Model selection	Periodic validation using loss and token accuracy	DAPO variants selected by validation behavior
Policy regularization	–	KL coefficient 0.01; asymmetric clipping 0.20 / 0.28

B.1 Replay runtime audit

Table 8 reports an implementation-level runtime audit for the reported full-prefix deployment harness. RTF is total controller wall-clock time divided by source-audio duration. The timer spans all pre-endpoint controller calls and intermediate thoughts, repeated prefix prefills, the final-think turn, and answer generation. These measurements therefore include the repeated prefix replay used in the reported controller implementation, so they complement the final-think token metric and do not replace it. The SFT initialization controller is the supervised checkpoint used to initialize the DAPO run in this audit.

The DAPO controller in this audit is the reported six-reward deployment lane. The main text interprets this audit separately from the residual-token metric.

B.2 Stateful KV-cache controller prototype

To check whether the same wait-think-answer information flow can be implemented cache-natively, we built an experimental stateful controller. Audio chunks are appended into a single persistent

Table 8: Replay-harness RTF for the full-prefix deployment audit. Lower is faster.

Controller lane	LLaMA-QS	Real Audio	ARC-C	ARC-E	SIQA	PIQA	GSM8K	Mean
Base controller	1.317	1.101	1.091	1.112	1.088	1.097	1.155	1.137
SFT controller	1.584	1.179	1.317	1.362	1.316	1.376	1.336	1.353
DAPO controller	1.278	1.136	1.080	1.198	1.035	1.088	1.123	1.134

Qwen2.5-Omni thinker KV cache. Each controller decision is generated from a temporary fork of that persistent state. `<wait/>` actions are not committed back into the persistent text cache; `<think>...</think>` updates are normalized and appended back; the final-think and answer turns are generated from the full cached audio plus all committed thoughts.

The prototype runs at the level of `model.forward(..., past_key_values=..., use_cache=True)` with a manual short-token decode loop, instead of the high-level `generate()` calls used by the benchmark path. Mechanism checks on a four-second real-audio sample confirm that cache lengths match attention lengths under both 2.0s and 0.5s chunking, and that wait actions do not enter the persistent cache. These are wiring-level smoke runs, not benchmark scores; turning the prototype into a score-producing batch runner remains future work.

B.3 SFT AND DAPO TRAINING DETAILS

The audio-only cold-start SFT run shows decreasing training loss and increasing token accuracy during training (Figure 3). DAPO update points are chosen based on validation behavior, since the streaming reward is non-monotonic across policy updates. The decision-grid comparison and causal padding example appear in Section 3, Table 1.

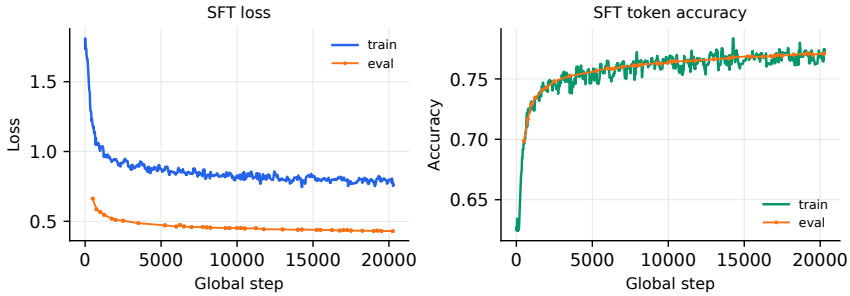

Figure 3: SFT training curves from the audio-only cold-start run. Training and validation metrics improve during supervised fine-tuning, providing the initialization for policy optimization.

For DAPO, each prompt is sampled into a group of G rollouts. We compute group-relative advantages

$$A_i = \frac{R_i - \frac{1}{G}\sum_{j=1}^{G} R_j}{\mathrm{std}(R_1, \ldots, R_G) + \epsilon},$$

and apply token-level clipped updates over the completion-token mask M. With $y_{i,t}$ denoting the t-th generated token in rollout i, $o_{i,t}$ its token-level context, and $r_{i,t}(\theta) = \exp(\log \pi_\theta(y_{i,t} \mid o_{i,t}) - \log \pi_{\mathrm{old}}(y_{i,t} \mid o_{i,t}))$, the objective is

$$\mathcal{L}_{\mathrm{DAPO}} = -\frac{1}{|M|} \sum_{(i,t)\in M} \begin{cases} \min(r_{i,t}A_i, \mathrm{clip}(r_{i,t}, 1-\epsilon_l, 1+\epsilon_h)A_i), & A_i \geq 0, \\ \max(r_{i,t}A_i, \mathrm{clip}(r_{i,t}, 1-\epsilon_l, 1+\epsilon_h)A_i), & A_i < 0. \end{cases} \tag{4}$$

Groups that fail the validity conditions in Section 3 are resampled up to a fixed retry budget; persistent failures are skipped and logged. The following paragraphs specify implementation details beyond the six-term summary in Table 2.

Reward implementation details. The six terms above are the conceptual reward categories; the implementation adds only local guardrails inside these categories. R_a reuses the two-stage correctness decision from Section 4 and adds difficulty-aware effort calibration, so a correct answer is not rewarded equally when the controller uses an implausibly empty or unnecessarily verbose reasoning state.

For R_s, let L_{final} be the token length of the post-endpoint final-think state. The base penalty is

$$P_s = -\min\left(3.0,\ 0.30\max(0, L_{\text{final}} - 6)\right). \tag{5}$$

The runtime token count is used when available; otherwise the reward path uses a character-based token estimate. A $+0.4$ compact-cue bonus is available only when the final answer is correct, the answer type is valid, the final state supports that answer, and $3 \le L_{\text{final}} \le 6$. Empty placeholders, unsupported cues, wrong answers, question-form answers, label-only answers when a semantic answer is expected, and yes/no type mismatches are ineligible. Thus the length reward cannot be maximized by emitting an uninformative short cue.

R_u is deterministic. Let T be the set of target evidence-update ticks from the aligned controller metadata and P the predicted pre-endpoint think ticks. Each prediction is paired with the nearest unmatched target within two 0.5s ticks. Matched pairs are true positives; unmatched predictions and targets are false positives and false negatives. The score combines update precision and recall with a sparsity corridor, penalizing both missed evidence changes and near-every-tick thinking. The symmetric window permits an early match; this timing score is not itself a test of prefix grounding. Grounding is checked separately during trajectory quality control.

The judge-assisted terms use Qwen3.6-35B-A3B (Qwen Team, 2026). R_t receives the question, current-evidence field, earlier visible state, new thought, model answer, and reference answer; R_c instead receives the complete visible thought chain. Both return one value in $\{0, 0.5, 1\}$. Appendix C gives the complete rubrics and input contracts. Out-of-contract outputs are parsed conservatively as zero. These judge calls provide training rewards, using different prompts and scoring criteria from benchmark evaluation; both use the same underlying judge model.

Reward weight selection. The reward weights reported in Section 3.5 were selected through validation-set sweeps during method development. Among the soft-coefficient terms, the update-timing weight λ_u and the residual-length weight λ_s had the largest qualitative effect on the accuracy–residual-work operating point because they directly shape when and how often the controller emits visible reasoning state. The correctness, protocol, thought-quality, and chain-consistency weights ($\lambda_a, \lambda_f, \lambda_t, \lambda_c$) were comparatively less sensitive in the explored range. Protocol validity is enforced by the gate in Eq. (3), rather than by increasing λ_f to outweigh other rewards; λ_t and λ_c weight the judge-scored preferences. These are qualitative observations from validation tuning, not a separate controlled sensitivity study.

C Prompts, alignment, and data construction notes

The canonical synthetic record contains the fields `question_text`, `tts_text`, `tts_instruct`, `transcript_text`, `anchor_words`, `logical_actions`, and `final_answer`. Here `question_text` is the canonical task question, `tts_text` is the spoken surface form passed to the TTS renderer, `tts_instruct` is the delivery/style control prompt, `transcript_text` is the normalized transcript aligned to the waveform, `anchor_words` are lexical anchors used to localize answer-relevant state changes, `logical_actions` is the semantic wait-think-answer trace before serialization, and `final_answer` is the gold answer string. The teacher prompt requires spoken-friendly wording, avoids unnatural written math notation where possible, and asks for thoughts that preserve short semantic state instead of generic summaries, tone descriptions, or pause descriptions. Acoustic style is controlled through `tts_instruct`; it should influence the rendered speech, not become the content of the reasoning state unless it changes the answer.

The generation plan crosses eight topics—travel booking, shopping and recommendation, scheduling, technical support and customer-service troubleshooting, math and logic, science, history and geography, and casual personal-assistant tasks—with two answer regimes and three difficulty levels. The verifiable regime requires a uniquely checkable answer; the open-ended regime permits a concise helpful response. Easy items contain one stable intent and one or two informative updates. Medium

items contain one reversal or adversarial pause and two or three updates. Hard items contain at least two streaming challenges, drawn from reversal, adversarial pause, and long-range condition accumulation, with four or more informative updates. This produces the 48 topic-by-verifiability-by-difficulty strata used in the generation plan.

Filtering and human trajectory audit. Normalization produces 77,184 candidate records from the initial 80,000-item generation plan. Validation and deduplication remove 1,461 records, leaving the 75,723 aligned examples used in the reported experiments. Automatic validators check schema validity, legal action order, anchor coverage, answer consistency, and duplicates. We additionally sample 200 trajectories across topics, answer regimes, and difficulty levels for author review. Each trajectory is checked for whether a thought is grounded in the audio prefix available at its timestamp, whether actions occur in legal order, whether intermediate updates carry task-relevant information, and whether the final state supports the answer. Detected failures are corrected or excluded before the data freeze. This is a quality-control audit rather than a multi-annotator labeling study, so we do not report inter-annotator agreement.

Full spoken inputs are rendered once. We do not synthesize each controller segment independently, because segment-wise synthesis would inject artificial prosodic breaks exactly where the model is supposed to learn natural timing. Forced alignment maps transcript words to audio timestamps, and action anchors are snapped upward to the 0.5s controller grid. The supervised export and the DAPO export are both generated from this aligned representation.

C.1 Benchmark semantic-fallback prompt

After deterministic task scoring rejects a response, the semantic fallback receives the question, choices when present, ground-truth answer, and generated text. Its instruction is:

```
Return exactly one minified JSON object and nothing else:
{"model_final_answer":"...",
 "judge_result":"<correct>|<incorrect>|<no_final_answer>"}

Use only the question, choices, ground-truth answer, and model response.
Ignore politeness and filler. Treat clear spoken paraphrases,
digit/number-word forms, and matching quantity-unit forms as equivalent.
When choices are present, judge equivalence relative to those choices.
If the response has no final answer, use <no_final_answer> for both fields.

Question: {question}
Choices: {choices}
Ground-truth answer: {ground_truth}
Model response: {model_response}
```

The parser accepts only the three declared labels. Invalid output is retried once with the same fields and a JSON-only reminder; a still-invalid output is resolved as incorrect. Numeric and answer-type guardrails can also downgrade a nominally correct judge label when the extracted answer conflicts with the reference quantity or required type.

C.2 Reward-judge prompt contracts

The thought-quality judge R_t receives the following instruction. The placeholder describing answer leakage differs for pre-endpoint and final-think states, but all other fields and scoring rules are fixed.

```
You are evaluating one <think> from a streaming audio reasoning model.

Return exactly one of: 1.0, 0.5, 0.0. Return no explanation.

A good <think> is a compact answer-state update. Pre-endpoint state may be
provisional when grounded. Final-think must preserve the requested answer
type and support the model answer against the reference answer. A concrete,
self-consistent but wrong note must not receive the highest score.
```

```
1.0 = grounded, answer-shaped state supporting the correct/reference answer
      or an equivalent model answer.
0.5 = partially useful or incomplete state that does not clearly contradict
      the reference answer.
0.0 = wrong state, wrong answer type/option/variable, intermediate value used
      as the final answer, unsupported hallucination, placeholder, or meta text.

Yes/no questions require yes/no state. Difference questions require the
difference; total/current-value questions require the requested final value.
Entity/action questions require the selected phrase rather than a question or
bare option label unless a label is the required format. Down-score prosody
commentary, task restatement, repetition, wrong-option justification, and
arithmetic over the wrong variable.

Question: {question}
Evidence span: {segment_span}
Earlier reasoning chain: {earlier_state_chain}
Previous think: {previous_state}
Current local evidence: {current_evidence}
Current think: {current_think}
Model final answer: {model_answer}
Reference answer: {reference_answer}
```

The chain-consistency judge R_c receives:

```
You are evaluating reference-aware short-state reasoning for a streaming
audio reasoning model.

Return exactly one of: 1.0, 0.5, 0.0. Return no explanation.

1.0 = the final stable chain supports the model answer, preserves the
      requested answer type, and is compatible with the reference answer.
0.5 = the chain is partially useful but incomplete, or points toward the
      reference without fully resolving it.
0.0 = the chain is self-consistent but wrong, supports an incorrect answer,
      preserves the wrong variable, selects the wrong option, has an
      answer-type mismatch, or provides no substantive support.

Yes/no questions must resolve to yes/no. Numeric questions must resolve to the
requested number/unit rather than an intermediate amount. Entity/action/option
questions must resolve to the requested phrase, not a question-form response
or a bare label unless the label is the expected format.

Question: {question}
Visible think-chain: {state_chain}
Model final answer: {model_answer}
Reference answer: {reference_answer}
```

Both judges require a parsed substantive state and model answer. Missing state, missing answer, invalid answer type, or output outside the three-value contract receives zero without an additional positive signal.

C.3 SYNTHETIC DATA GENERATION PROMPT

The project-specific synthetic data prompt is reproduced below. The core GPT-4o (OpenAI, 2024) contract is:

```
System:
You create high-quality English training records for a streaming speech
reasoning model. Return one JSON object only.

The spoken script is always user speech only. Do not solve the task or speak as
the assistant inside tts_text or transcript_text. Do not use inline TTS markup
such as <break>. Use punctuation and a separate tts_instruct field to induce
```

```
natural speech style when needed.

transcript_text must be clean English alignment text. logical_actions must use
only <wait/>, <think>...</think>, and <answer>...</answer>. Some anchors may map
to <wait/> because a physical pause or filler does not necessarily require
thinking.

Every <think> should be short, incremental, grounded in the current evidence,
and useful for the answer state. Prefer concrete facts, constraints,
eliminations, quantities, corrections, answer-type constraints, or current
candidate state. Avoid generic meta commentary and avoid long explanations.

Every sample must be unique. Treat the scenario seed only as a starting
situation and vary names, numbers, locations, items, and wording. Do not use
LaTeX, bullet lists, stage directions, or duplex terminology.

User template:
Create one English training example.
sample_id: {sample_id}
task_family: {task_family}
topic: {topic}
verifiability: {verifiability}
task_type: {task_type}
difficulty: {difficulty}
variant_index: {variant_index}
scenario_seed: {scenario_seed}
Target about {pause_count} meaningful anchor boundaries.
{family_guidance}
{difficulty_guidance}

Return JSON fields:
task_type, topic, verifiable, difficulty, question_text, tts_text,
tts_instruct, transcript_text, anchor_words, logical_actions, final_answer,
difficulty_metadata.

Key requirements:
- tts_text and transcript_text contain only user speech.
- anchor_words are copied from transcript_text.
- logical_actions has one key per anchor plus anchor_AUDIO_END.
- non-final anchors use <wait/> or <think>...</think>, never <answer>.
- anchor_AUDIO_END contains one final <think>...</think> followed by
  one <answer>...</answer>.
- the answer text exactly matches final_answer.
```

C.4 Spoken SRQA prompt provenance

For the six synthetic SRQA benchmark families, we follow the spoken-rewrite prompt contract from Shih et al. (2026). The implementation uses their task-specific prompts for ARC, PIQA, SocialIQA, and GSM8K style conversion: preserve all information needed for the answer, verbalize choices inside the spoken question, and return a JSON object containing the converted spoken question and, where applicable, the converted answer. We cite their prompt without reproducing it as a new project-specific prompt.

C.5 Real Audio Bench generation prompt

Real Audio Bench uses a separate GPT-4o candidate-generation prompt because its goal is real human delivery, not TTS rendering. The source generation spec fixes 200 planned items, five speakers, and 40 recordings per speaker, then filters the recorded set to 186 final rows. The universal prompt is:

```
System:
You are writing candidate items for a real-human benchmark for streaming spoken
reasoning.
```

```
This benchmark is NOT ordinary QA. It is for a streaming controller that hears
audio over time and must wait until a late decisive cue arrives.

Hard constraints:
- Write natural spoken English only.
- The utterance must sound like something a person would actually say aloud to
  an assistant.
- Every item must be verifiable from the utterance itself.
- Each item must have exactly one preferred gold answer.
- The spoken utterance must contain an early plausible answer that later
  becomes wrong, incomplete, or premature.
- A late cue must overwrite or finalize the answer.
- Avoid textbook math, school trivia, science, history, and canned benchmark
  style.
- Keep spoken duration within 8 to 18 seconds.
- Keep the answer short and cleanly scoreable.
- Across a batch, diversify the reasoning mechanism.

User template:
Generate {count} candidate items.

Coverage requirements:
- use the requested categories, scenario families, difficulties, and reasoning
  mechanisms as evenly as possible
- all items must be verifiable
- prioritize distinctness from the existing synthetic corpus
- every item must pass a uniqueness test: after reading the full utterance,
  there should be exactly one reasonable gold answer
- when using time windows, thresholds, eligibility, or scheduling conflicts,
  state the deciding rule explicitly

Allowed categories:
mid_way_reversal, adversarial_pause, incremental_accumulation,
noisy_environment, fast_speech_hesitation, real_world_situational

Allowed scenario families:
day_of_travel_logistics, home_landlord_appliance, workplace_school_admin_it,
food_shopping_returns, health_pharmacy_finance_lite

Allowed reasoning mechanisms:
overwrite_final_slot, cumulative_total, tiered_discount_total,
fee_or_threshold_decision, exclusion_choice, bounded_window_selection,
quantity_update, eligibility_decision, refund_or_credit_total,
schedule_window_resolution

Return one JSON object with an items array. Each item has:
item_id, category, difficulty, scenario_family, reasoning_mechanism,
verifiability, question_spoken, gold_answer, answer_check_type,
why_unique_answer, ambiguity_check, why_early_answer_would_be_wrong,
critical_audio_cue, speaker_notes, estimated_duration_sec, acoustic_target.
```

After generation, an LLM verifier and human audit screened the candidate pool for a unique answer, spoken naturalness, reasoning strength, and distinctness. The audit also applied 23 answer-key corrections before the manifest was frozen.

D Additional results and ablations

D.1 Qwen operating points across protocols

Table 9 collects the Qwen controls discussed in Section 5.3. Accuracy and post-end residual tokens use synthetic SRQA; mean RTF is the seven-set replay audit in Appendix B.1.

Table 9: Qwen operating points. Native direct has no required final-think span. Dashes indicate inapplicable or unmeasured quantities, not zero cost.

Protocol and checkpoint	Acc. (%)	Post-end tokens	Mean RTF
Native direct, Base	71.85	–	–
Native direct, DAPO (6 rewards)	71.89	–	–
Complete-audio reasoning, Base	70.8	10.27	–
Streaming Base	67.6	10.44	1.137
Streaming SFT	66.1	9.82	1.353
Streaming DAPO (4 rewards)	68.5	10.87	–
Streaming DAPO (5 rewards)	69.2	10.94	–
Streaming DAPO (6 rewards)	70.3	8.99	1.134

D.2 Per-task synthetic SRQA accuracy

Figure 4 visualizes the matched comparison between the streaming Base controller and the six-reward DAPO controller reported in Table 3. DAPO improves five task families and matches Base on LLaMA-QS; GSM8K uses a separate horizontal scale so that its lower absolute accuracy does not compress the other five comparisons.

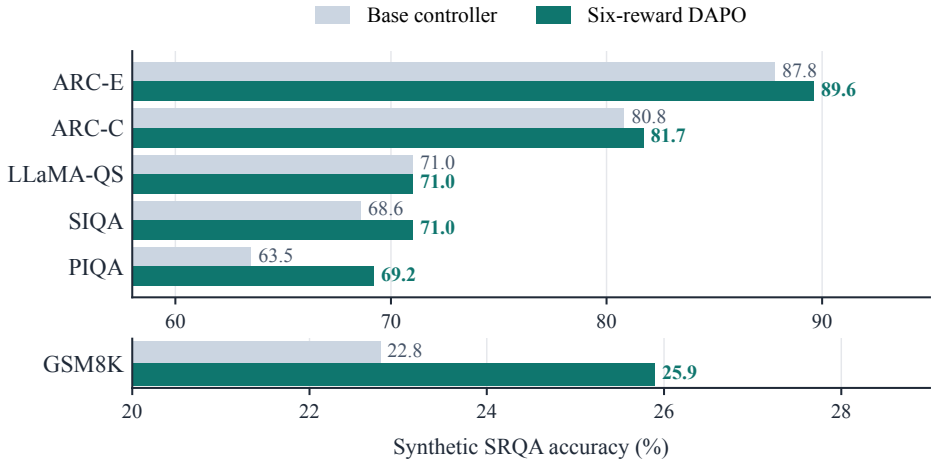

Figure 4: Per-task synthetic SRQA accuracy for the streaming Base controller and the six-reward DAPO controller. GSM8K is shown in a separate panel because its accuracy lies near 25%.

D.3 External model-native residual counts

Table 10 retains the original descriptive counts for the complete-audio external models. Each count uses that model's tokenizer and prompted final-think span; these values are not a cross-model latency ranking. The Qwen protocols and costs appear together in Table 9.

Table 10: Complete-audio external models: final-think length in each model's own text tokens. No conversion to wall-clock time is implied.

Model	Model-native final-think tokens
Audio Flamingo 3	4.97
Audio Flamingo 3 + AF-Think	9.18
GLM-4-Voice-9B	20.88

D.4 PAPER-LEVEL COMPARISON WITH SHIH ET AL.

Shih et al. (2026) use a Moshi text-monologue stream and define latency as the number of tokens between the end of the question and the start of the system response. Our Qwen controller counts text tokens in the residual final-think state after the supplied endpoint. The boundaries are related, but the tokenizers, generation clocks, model families, and controller interfaces are different. Table 11 therefore reports each method's change relative to its own matched starting point; it is not a head-to-head ranking of Moshi and Qwen latency.

Table 11: Within-system accuracy and residual-token changes. Shih et al. values are row-weighted aggregates of their reported per-task results, using the task counts in Appendix A; Qwen values are measured in our harness. Token counts remain in each architecture's native text stream.

Within-system comparison	Accuracy (%)	Native residual counts
Moshi QC-SFT → length-DPO	$44.4 \rightarrow 46.7$	$52.42 \rightarrow 19.31$ (-63.2%)
Qwen streaming Base → DAPO	$67.6 \rightarrow 70.3$	$10.44 \rightarrow 8.99$ (-13.9%)

The two results support the same qualitative conclusion under different architectures: post-training can reduce the measured post-end reasoning output while improving accuracy relative to the matched reasoning-enabled starting point. They do not show that the Qwen controller is faster than the Moshi controller.

D.5 CONTROLLER LAYER RELATIVE TO ADJACENT SYSTEMS

Table 12 separates the layer at which each nearby system decides when to reason. This is an architectural comparison rather than an empirical ranking: the systems use different backbones, datasets, action interfaces, and response clocks.

Table 12: Controller-level positioning relative to adjacent think-while-listening systems.

System	Timing observation and decision	Reasoning state and optimization
Shih et al. (2026)	Moshi text-monologue stream; question completeness selects the CoT onset	CoT generation followed by correctness/length preference optimization
LTS-VoiceAgent (Zou et al., 2026)	Streaming ASR text; a separate semantic trigger selects informative prefixes	An external dual-role orchestrator coordinates incremental Thinker and Speaker state
SHANKS (Chiang et al., 2025)	Fixed-duration speech chunks; unspoken reasoning is generated as each chunk arrives	Continuous chunkwise state supports interruption and tool-use decisions
Ours	Native Qwen audio prefixes plus committed visible states; the model policy chooses WAIT or THINK	Recurrent state updates are initialized by SFT and optimized with a whole-trajectory DAPO reward

Within the Qwen controller family, the SFT-to-DAPO comparison reports the observed checkpoint difference under the complete training recipe: accuracy increases from 66.1% to 70.3%, post-end residual state decreases from 9.82 to 8.99 tokens, and replay-harness RTF decreases from 1.353 to 1.134. These measurements describe the reported post-training run; they do not establish that the resulting controller outperforms LTS-VoiceAgent or SHANKS under a matched system evaluation.

D.6 PRIMARY-BENCHMARK ITEM-LEVEL EVIDENCE

On the 8,959-item synthetic SRQA set, streaming Base, SFT, and six-reward DAPO answer 6,054, 5,919, and 6,299 items correctly. Even the least-favorable paired tables compatible with these exact marginal counts give two-sided exact McNemar bounds of $p \leq .0011$ for DAPO versus Base and $p \leq 5.2 \times 10^{-7}$ for DAPO versus SFT. These tests reject an equal-accuracy null in this paired item-level analysis, conditional on the fixed checkpoints and scorer. The bounds do not measure training-seed variation, which remains unavailable for the reported checkpoints.

D.7 REAL AUDIO BENCH BOOTSTRAP CONFIDENCE INTERVALS

Real Audio Bench contains 186 human-recorded items (Section 4). To quantify the uncertainty introduced by this small evaluation size, Table 13 reports per-lane accuracy with 95% bootstrap

confidence intervals over 10,000 resamples of the 186 items. The lanes match the controller variants reported in Table 4.

Table 13: Real Audio Bench accuracy with 95% bootstrap confidence intervals (10,000 resamples). The lanes match the controller variants reported in Table 4.

Lane	95% bootstrap CI
Base controller	[0.570, 0.710]
SFT controller	[0.618, 0.753]
DAPO controller (4 rewards)	[0.586, 0.726]
DAPO controller (5 rewards)	[0.608, 0.742]
DAPO controller (6 rewards)	[0.581, 0.720]

The corresponding point estimates and their interpretation are reported once in Table 4 and the surrounding main-text discussion.

E Data ethics and intended use

The synthetic training corpus is generated from authored schemas and model outputs rather than recordings of private conversations. Real Audio Bench is constructed from scripted prompts recorded specifically for this project by five internal readers. The public split contains no spontaneous conversations, personal narratives, demographic attributes, or author-provided speaker names; metadata uses reader IDs. Candidate prompts, screening decisions, corrected answers, and source identifiers are retained in the release manifest to make provenance auditable. The dataset is released under CC BY 4.0 for research on spoken-question reasoning. All five readers provided informed consent for the public research distribution and use of their voice recordings; no visual likeness was collected.

Voice recordings can nevertheless act as biometric identifiers. The intended use is evaluation of answer content under human-recorded speech, not speaker identification, voice cloning, biometric inference, or surveillance. The set is small, English-only, and collected from five readers, so it should not be used to draw demographic conclusions or to represent general conversational speech.

F Asset licenses

Table 14 summarizes upstream asset licenses and terms, checked against official model or dataset cards, repositories, or source documentation where available at the time of submission.

Table 14: Public licenses or terms of use for upstream assets used in this work.

Asset	License or terms
Qwen2.5-Omni-7B (Xu et al., 2025a)	Apache 2.0
Qwen3-TTS (Hu et al., 2026)	Apache 2.0
Qwen3.6-35B-A3B (Qwen Team, 2026)	Apache 2.0
GLM-4-Voice-9B (Zeng et al., 2024)	GLM-4-Voice License; code under Apache 2.0
Audio Flamingo 3 (Goel et al., 2025)	NVIDIA OneWay Noncommercial License
GPT-4o (OpenAI, 2024)	OpenAI API terms of service
MS-Swift (Zhao et al., 2024)	Apache 2.0
ARC (Clark et al., 2018)	CC-BY-SA 4.0
PIQA (Bisk et al., 2020)	Academic Free License 3.0
SocialIQA (Sap et al., 2019)	CC-BY 4.0
GSM8K (Cobbe et al., 2021)	MIT
LLaMA-Questions/Spectron (Nachmani et al., 2023)	Source paper / archived GitHub terms; license not specified