AURAL: ADAPTIVE LATENT REASONING WITH JOINT CHUNK FOR SPEECH LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Model intelligence and fast response jointly shape the quality of interaction with speech language models, yet remain difficult to achieve together. Explicit chain-of-thought (CoT) improves reasoning and audio understanding, but generating intermediate reasoning tokens delays responses. Describing fine-grained acoustic cues further lengthens CoT and increases latency. Latent reasoning can reduce this overhead, yet existing methods often trail CoT and remain limited by single-path supervision and reasoning budgets that do not adapt to problem difficulty. We introduce **AURAL**, which models a distribution over multiple plausible reasoning continuations in latent space and jointly predicts chunks of future states to reduce sequential forward passes and reasoning latency. To provide initial supervision for latent reasoning, we construct AuralReason-683K: 683K bilingual speech utterances (about 1,000 hours) with concise CoT for emotion recognition, empathetic dialogue, and general reasoning. AURAL-RL then explores beyond these traces, rewarding concise reasoning that yields high-quality answers and adapting reasoning effort to each problem. Across two backbones, AURAL-RL achieves performance comparable to CoT-RL, with larger gains over the respective supervised checkpoints on most metrics. Analysis further shows that harder questions elicit more latent reasoning steps. On Qwen2.5-Omni, it reduces time to the first answer token by 11.8×, from 1.22 to 0.10 s, versus 0.05 s for direct answering.

1 INTRODUCTION

Interaction is one of the defining capabilities of a speech language model (SLM). Users will not keep talking to a model that lacks the intelligence to understand their needs and reason well enough to help. Nor will they wait through a long pause for every thoughtful answer. The ideal resembles the voice assistant imagined in *Her*[1], responsive enough to feel present and perceptive enough to understand what the user means. Although current SLMs remain far from this vision, improving both intelligence and response speed is essential to closing the gap.

Explicit chain of thought (CoT) is a natural way to improve intelligence in LLMs, and in SLMs its benefits extend further to audio understanding (Tian et al., 2025a). Yet CoT remains rare in interactive models because every intermediate reasoning token is generated before the answer and is therefore paid for in response latency. For speech, CoT introduces a second cost. Acoustic cues such as timbre, prosody, emotion, speaker traits, and background sounds vary continuously. Changes in these cues can alter how an utterance is understood, but text often reduces them to simple labels. The label "upset" can describe the tone of "I am fine", but it does not capture the strength and mixture of hesitation, tension, and sadness in the voice. Describing these details in CoT requires more words, which add tokens and further increase latency. Recent work finds that even when CoT is used, SLMs may recognize paralinguistic cues without using them in their responses (Wang et al., 2026d;e). CoT therefore offers a natural path to more capable SLMs, yet its serial generation slows responses and its discrete form forces a trade between acoustic precision and speed.

Continuous hidden states can retain acoustic detail without spelling it out token by token. Latent reasoning performs intermediate computation in these states, reducing the number of serial reasoning steps before an answer. Each state can also represent information that does not map cleanly to a single vocabulary token, giving the model a broader reasoning space than explicit CoT. How-

[1] *Her* is a science fiction film released in 2013.

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.

ever, latent reasoning remains rare in SLMs. FLAIR fills the silence slots that occur while the user is speaking with latent thoughts, so speech length determines the number of reasoning steps (Wu et al., 2026). CoAT and LatentOmni instead use a latent budget fixed in advance (Han et al., 2026; Dai et al., 2026). Although these designs limit response latency, they often lag behind explicit CoT in reasoning quality. Their reasoning budgets do not adapt to problem difficulty, and supervision typically provides a single target for each latent step, encouraging one continuation rather than preserving plausible alternatives. In addition, explicit CoT remains useful as initial supervision for latent reasoning, yet large speech corpora with reliable traces remain scarce.

We therefore propose **AURAL** (**A**daptive **U**nspoken **R**easoning over **A**lternative **L**atents), a latent reasoning framework for SLMs that retains the benefits of CoT at much lower response latency (Figure 1). AURAL models multiple plausible reasoning continuations in latent space with a Gaussian mixture model and jointly predicts chunks of future states to reduce sequential forward passes. To provide initial supervision, we construct AuralReason-683K, a carefully filtered bilingual collection of 683K utterances (about 1,000 hours) with concise CoT traces for speech emotion recognition, empathetic dialogue, and general reasoning. AURAL-SFT pools consecutive CoT embeddings into latent targets. AURAL-RL then explores beyond these traces, rewarding concise reasoning that yields high-quality answers and learning a problem-dependent stopping policy. Across two SLM backbones, AURAL-RL achieves performance comparable to CoT-RL, with larger gains over the respective supervised checkpoints on most metrics. Analysis on GPQA and VoiceBench BBH further shows that harder questions elicit more latent reasoning steps. With the default setting on Qwen2.5-Omni, it reduces time to the first answer token from 1.22 s for CoT-RL to 0.10 s (an $11.8\times$ speedup), compared with 0.05 s for direct answering. Our contributions are:

- We introduce AURAL, which models a distribution over plausible reasoning continuations in latent space and jointly predicts future-state chunks to reduce forward passes and reasoning latency.
- We construct AuralReason-683K, a bilingual corpus of 683K speech utterances (about 1,000 hours) with concise CoT for emotion recognition, empathetic dialogue, and general reasoning.
- We develop AURAL-RL to explore beyond supervised traces and adapt reasoning effort to each problem. It achieves performance comparable to CoT-RL with an $11.8\times$ speedup in time to the first answer token on Qwen2.5-Omni.

2 Related Work

Latent reasoning moves intermediate computation from verbalized CoT into continuous states. Coconut feeds predicted hidden states back into the model (Hao et al., 2025), while CODI adds endpoint self-distillation (Shen et al., 2025). Because neither method provides targets for intermediate latent states, the latent chain cannot be trained efficiently with teacher forcing and must instead be unrolled serially. Their preset latent length also prevents depth from adapting to problem difficulty. CCoT supplies teacher-forced targets, which improves supervision but still generates them autoregressively (Cheng & Van Durme, 2024). PCCoT replaces slow serial update with parallel Jacobi refinement at the cost of several refinement rounds (Wu et al., 2025a). Latent-SFT obtains fine-grained supervision targets from a separately trained encoder (Deng et al., 2025), while CoLaR (Tan et al., 2026) and C-MTP (Yerram et al., 2026) greatly simplify target construction by pooling grouped CoT embeddings but providing coarser supervision. Beyond supervised target design, Latent-GRPO shows that latent policy optimization can leave the valid manifold or average incompatible modes (Deng et al., 2026). Latent Thought Flow instead learns stochastic trajectories whose length depends on answer quality and computation cost (Zou et al., 2026). These findings motivate latent reasoning that preserves multiple valid continuations and uses more computation only when a problem requires it.

Latent reasoning in SLMs remains sparse. FLAIR is the closest full-duplex precedent (Wu et al., 2026). It replaces idle listening slots with recursively fed soft vocabulary embeddings learned from a global expert, enabling causal thought while the user speaks. Its reasoning depth is nevertheless determined by the incoming speech schedule rather than problem difficulty. CoAT inserts a latent workspace supervised by hand-selected audio experts (Han et al., 2026). Because both the workspace and its supervisory dimensions are prescribed, it learns an engineered acoustic representation rather than discovering an autonomous reasoning trajectory. HyPeR invokes pause states at acoustically ambiguous points (Wang et al., 2026c). This adds perception compute where confidence is low, but does not optimize reasoning length or use continuous states to represent alternative paths. LatentOmni interleaves text with recurrent audio-visual states so dense sensory evidence need not

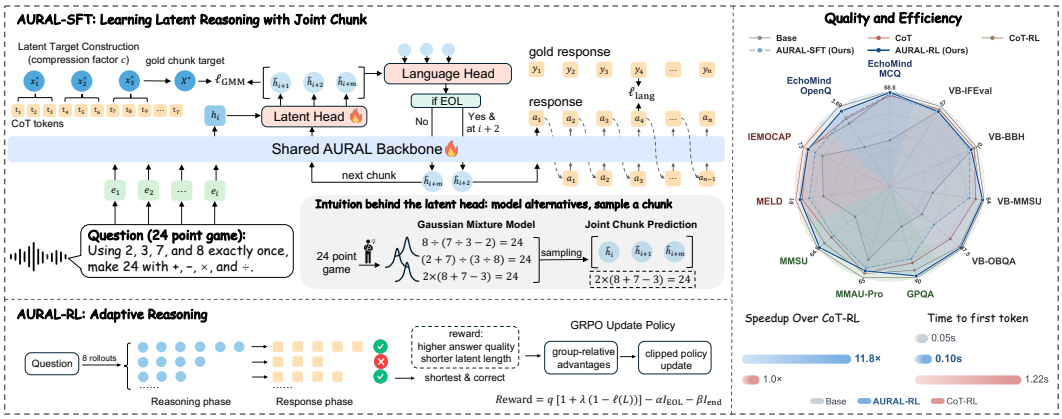


Figure 1: Overview of AURAL. **Top left:** AURAL-SFT pools CoT embeddings into latent targets under compression factor c, then predicts a Gaussian mixture over a joint chunk of m future states that reenters the backbone in one pass, while the language head keeps each state readable and emits EOL. **Bottom left:** AURAL-RL scores 8 rollouts per prompt with a quality-gated conciseness reward, making latent depth adaptive. **Right:** AURAL-RL is comparable to CoT-RL over eleven metrics while cutting time to first token from 1.22 to 0.10 s (11.8×).

be verbalized (Dai et al., 2026). This motivation is close to ours, but its fixed latent budget and serial recurrence prioritize multimodal grounding over reasoning efficiency and path diversity.

3 METHOD

Overview. Explicit CoT provides direct supervision, but inference must verbalize every intermediate step as a discrete token, making it slow and restricting reasoning to token-expressible paths. AURAL retains CoT only as a training signal. After CoT finetuning on AuralReason-683K (Section 4), AURAL-SFT pools consecutive CoT embeddings into continuous targets and uses a Gaussian mixture model (GMM) to represent multiple chunks of future states; at inference, each sampled chunk is fed back in one backbone pass. AURAL-RL then optimizes complete rollouts, adapting reasoning depth to the problem while improving answer quality. Figure 1 summarizes both stages.

3.1 AURAL-SFT: LEARNING LATENT REASONING WITH JOINT CHUNK

From CoT tokens to latent targets. The central challenge in supervised latent reasoning is that intermediate latent states have no natural targets. We therefore adopt the simple supervision design of CoLaR (Tan et al., 2026), which pools consecutive CoT token embeddings into an aligned latent sequence. This preserves the order of the original CoT and gives every latent position an explicit target, allowing efficient training with teacher forcing. Given a compression factor c, let $\mathcal{G}_i = (y_{i,1}, \ldots, y_{i,n_i})$ be the ith group with embeddings $e_{i,j} \in \mathbb{R}^d$, where $n_i = c$ except possibly for the final group. Each group yields one target

$$u_i = \frac{1}{\sqrt{n_i}} \sum_{j=1}^{n_i} e_{i,j}, \qquad x_i = \frac{u_i}{s_e}, \tag{1}$$

where s_e is the embedding standard deviation. The factor $1/\sqrt{n_i}$ preserves scale. This maps T CoT tokens to $\lceil T/c \rceil$ latent targets, while x_i stabilizes density fitting.

Modeling alternative latents. Let h_i denote the final backbone hidden state after processing the question and the latent prefix through position i. The intuitive baseline regresses one next state x_{i+1} from h_i. Because each training example contains one recorded CoT, point regression treats its pooled state as the unique continuation of that prefix. It can compress the observed trace, but cannot represent uncertainty when the current reasoning state admits several valid next moves. Continuous latent reasoning is useful precisely because it can retain these alternatives before committing to one explicit path (Hao et al., 2025). These alternatives may represent different reasoning operations, so their latent targets can occupy separated regions rather than vary around one center. We represent

these regions as distinct modes of $p_\theta(x_{i+1} \mid h_i)$, allowing the transition distribution to place probability mass on several alternatives. The GMM lets different components capture different modes, while their weights express the relative plausibility of these continuations. Let $D_{ik} = \mathrm{diag}(\sigma_{ik}^2)$.

$$p_\theta(x_{i+1} \mid h_i) = \sum_{k=1}^{K} \pi_{ik}\, \mathcal{N}(x_{i+1} \mid \mu_{ik}, D_{ik}), \qquad \mathcal{L}_{\mathrm{GMM}}(i) = -\frac{1}{d} \log p_\theta(x_{i+1} \mid h_i). \tag{2}$$

Here K is the number of Gaussian components and k indexes one component. The loss $\mathcal{L}_{\mathrm{GMM}}$ is the negative log likelihood of the observed pooled target under the predicted mixture, averaged over its d dimensions. Minimizing it jointly learns $\{\pi_{ik}, \mu_{ik}, \sigma_{ik}^2\}_{k=1}^{K}$. Appendix E intervenes on this component index at a fixed state and traces the effect on the resulting reasoning path.

Joint chunk prediction adds structure and speed. The GMM restores multiple possible transitions, but applying it independently at each position still leaves the structure among neighboring latent states implicit. Under teacher forcing, each prediction receives the gold prefix, so independent one-state losses do not directly teach several future states to evolve together. Inference is also serial because each predicted state must pass through the backbone before the next state can be generated. Generating a latent trace of length L therefore requires L sequential forward passes through the backbone. We address both limitations by predicting a chunk of m future states jointly, reducing this cost to $\lceil L/m \rceil$ sequential forward passes. Our latent head uses m learned slot embeddings, one for each future state. A shared MLP maps h_i into the slot feature space. Token mixing exchanges information among slots, while channel mixing refines each slot. Separate linear heads then output $\{\pi_{ik}, \mu_{ik}, \sigma_{ik}^2, F_{ik}\}_{k=1}^{K}$. This shared slot design explicitly models relations within the chunk and uses fewer parameters than a monolithic MLP that emits the entire chunk. Formally

$$\begin{aligned} X_i &= (x_{i+1}, \ldots, x_{i+m}), \qquad p_\theta(X_i \mid h_i) = \sum_{k=1}^{K} \pi_{ik} \mathcal{N}(\mathrm{vec}(X_i) \mid \mu_{ik}, \Sigma_{ik}), \\ \Sigma_{ik} &= \mathrm{diag}(\sigma_{ik}^2) + F_{ik} F_{ik}^\top, \qquad F_{ik} \in \mathbb{R}^{md \times r}, \qquad r = 16. \end{aligned} \tag{3}$$

The operator $\mathrm{vec}(X_i) \in \mathbb{R}^{md}$ concatenates the m latent states into one vector. The diagonal term captures coordinate uncertainty, while the low-rank term couples positions within the chunk. We use a masked negative log likelihood for partial final chunks.

CSA keeps latent reasoning connected to language. The freedom introduced by the GMM chunk distribution creates a complementary risk. Latent states may drift far away from language space, yet the language head must eventually predict a special EOL token that stops latent reasoning and starts response generation. If this connection is learned only at the boundary, both the stopping decision and answer decoding become difficult. We therefore use chunkwise semantic alignment (CSA) as a soft language anchor throughout reasoning

$$\mathcal{L}_{\mathrm{CSA}}(i) = -\frac{1}{n_i} \sum_{j=1}^{n_i} \log p_\theta(y_{i,j} \mid h_{i-1}). \tag{4}$$

When h_{i-1} predicts group $\mathcal{G}_i$, the language head maps it to a distribution over the vocabulary. CSA assigns equal target mass to all n_i CoT tokens summarized by this group. It therefore keeps each latent state semantically readable by the language head, making the EOL decision and final response easier to decode.

LSS reduces exposure bias in continuous space. CSA anchors latent states to language, but teacher forcing still creates a mismatch. Training consumes gold latent states, whereas inference consumes samples from the GMM, so early errors can move later states outside the training distribution. We address this with latent scheduled sampling (LSS), a two-pass adaptation of scheduled sampling (Bengio et al., 2015; Mihaylova & Martins, 2019). The first pass predicts reparameterized chunks from gold inputs. The second pass consumes these samples and applies the same CSA and token losses. Since the samples are not detached, gradients also train the latent head through predicted inputs. Let $\mathcal{L}_{\mathrm{lang}}$ combine CSA with standard cross entropy on EOL and response tokens. Superscripts gold and pred identify the latent inputs used by each pass. The complete objective is

$$\mathcal{L}_{\mathrm{latent}} = \mathcal{L}_{\mathrm{GMM}} + \mathcal{L}_{\mathrm{lang}}^{\mathrm{gold}} + w_t \mathcal{L}_{\mathrm{lang}}^{\mathrm{pred}}, \qquad w_t = \min(1, t/T_{\mathrm{warm}}). \tag{5}$$

Here t is the current optimization step and $T_{\mathrm{warm}} = 500$ is the number of warmup updates. Thus w_t increases the predicted input loss linearly from 0 to 1 during warmup and remains at 1 afterward.

3.2 AURAL-RL: Adaptive Reasoning with Reinforcement Learning

From alternative paths to adaptive depth. AURAL-SFT imitates one recorded CoT and its length, so an alternative latent path receives no credit even if it produces the correct answer. AURAL-RL instead scores complete rollouts by their final outcomes, allowing the model to discover valid latent paths and allocate different amounts of depth to different problems. The GMM lets us evaluate the same sampled chunk under the updated policy and the policy that generated it. Because chunk log density scales with the number of accepted latent coordinates, we divide each log density by that number before exponentiating their difference. This normalization keeps latent-action updates comparable across chunk sizes and provides the multiplier for our GRPO-style clipped objective (Shao et al., 2024); Appendix A gives the equations. For each prompt, we sample 8 rollouts, compute group-relative advantages, and update all parameters without a reference policy.

Reward design. For questions with a verifiable answer, we compute quality $q \in [0, 1]$ using exact match or the task-specific checker. Open-ended responses are judged by DeepSeek-V4-Flash, while tasks whose correctness depends on the source audio are judged by Gemini-3.5-Flash. Let I_{EOL} equal 1 when a rollout never emits EOL, and let I_{end} equal 1 when its response lacks the end token. Both are 0 otherwise. For latent length L and rollout cap $L_{\max}$, we define the normalized length cost and reward as

$$\ell(L) = \frac{L-1}{L_{\max}-1}, \qquad R = q\,[1 + \lambda\,(1 - \ell(L))] - \alpha I_{\mathrm{EOL}} - \beta I_{\mathrm{end}}, \tag{6}$$

where $L_{\max} = 64$ is the maximum latent budget used for each rollout. The value $\ell(L)$ increases from 0 for the shortest trace to 1 at this limit. Thus the conciseness bonus is largest for a short correct trace and decreases linearly with length. Multiplication by q prevents an incorrect answer from earning a shortcut reward by stopping early. This quality-gated bonus is what makes AURAL-RL adaptive: additional latent depth is rewarded only when it contributes to a better outcome. We use $\lambda = 0.5$, $\alpha = 0.1$, and $\beta = 0.02$, with the last two coefficients penalizing invalid termination.

4 Data Construction

Latent reasoners learn their initial behavior from explicit CoT, but existing public speech corpora do not provide reasoning that uses the voice rather than only the transcript. We therefore build AuralReason-683K, a bilingual corpus of 683K utterances and about 1,000 hours of speech for emotion recognition, empathetic dialogue, and general reasoning. Appendix F gives the full pipeline, per-source counts, and teacher prompts.

To teach the model what the voice can change, we begin with two complementary sources. LIME-440K offers mostly synthesized bilingual affective dialogue, while EmotionCoT-35K supplies English human recordings for emotion classification (Zhao et al., 2026; Wang et al., 2026b). We retain only their audio and write new CoTs and responses. Controlled speech makes affect easy to isolate, but it covers only a narrow slice of real interaction. We therefore build HumanSpeech-1M from about 20K hours of public videos, podcasts, films, and speech corpora. Scale alone is not sufficient. Useful supervision requires utterances whose vocal delivery materially changes the appropriate assistant response. After removing synthetic or unreliable audio, we screen the remaining clips for this criterion, retaining 75K of 1.69M candidates. These three branches teach voice-sensitive interaction, but they do not teach broad problem solving. GeneralSpeech fills that gap with bilingual tasks in mathematics, knowledge, comprehension, and instruction following. We rewrite public text problems as natural spoken questions before synthesizing them with Qwen3-TTS (Hu et al., 2026).

Once the audio is selected, the next question is how much reasoning to write. AURAL derives its latent targets from explicit CoT, so a verbose teacher trace is not harmless. Repeated explanations can become extra latent computation. We therefore seek the shortest trace that still supports the response, retaining every necessary inference while removing restatement and reasoning after the conclusion. Prior work likewise finds that shorter valid CoTs can preserve or improve accuracy (Hassid et al., 2025; Wu et al., 2025b; Xu et al., 2025c). For dialogue, the trace must explain how the voice changes the reply rather than merely name an emotion. Gemini-3.5-Flash listens to each clip and writes states with distinct roles. *Cue* records what is audible, *Need* interprets what the speaker needs, optional *Risk* handles ambiguity or safety, and *Plan* determines the reply. A shaky voice with long pauses, for example, motivates reassurance before advice. Emotion traces instead

use audible cues to rule out the closest competing emotion before describing the affective state. Across all tasks, the CoT stops before the final answer so that it teaches a reasoning path rather than a shortcut to the target (Shen et al., 2025).

GeneralSpeech begins with a CoT and response from Qwen3-Omni-Thinking (Xu et al., 2025b). Gemini-3.5-Flash then creates two versions because training and analysis require different guarantees. For the controlled comparison in Section 5.5, Gemini shortens only the CoT and keeps the answer unchanged, even when wrong. Any performance difference therefore comes from the rewritten CoT rather than answer correction. For the main training set, it verifies the answer and repairs the CoT and response when needed. Only verified outputs enter the corpus. After filtering and rewriting, LIME-440K, EmotionCoT-35K, HumanSpeech-1M, and GeneralSpeech contain 320K, 18K, 75K, and 270K examples. Their 377K Chinese and 306K English utterances form AuralReason-683K.

5 Experiments

For latent reasoning to be practically useful, its efficiency gains must not come at the cost of broad task competence or architectural generality. Our experiments examine whether AURAL preserves answer quality across diverse speech and text tasks, transfers across backbone architectures, and reduces the computation required before producing an answer.

5.1 Experimental Setup

We assemble an evaluation suite that covers speech-specific capabilities and broader reasoning. EchoMind (Zhou et al., 2026) evaluates vocal understanding, integrated reasoning, and empathetic response generation. We report multiple-choice accuracy and the mean judge score across its four official dimensions for empathetic responses. IEMOCAP (Busso et al., 2008) and MELD (Poria et al., 2019) evaluate speech emotion recognition on their standard test sets. MMSU (Wang et al., 2026a) covers fine-grained spoken language perception and reasoning. MMAU-Pro (Kumar et al., 2026) measures broad audio intelligence, while GPQA (Rein et al., 2024) measures expert-level scientific reasoning. VoiceBench (Chen et al., 2026) spans multiple dimensions of voice assistant capability. We select four subsets with objective metrics that assess knowledge, reasoning, and instruction following. These subsets are OpenBookQA, MMSU, BBH, and IFEval.

Qwen2.5-Omni-7B (Xu et al., 2025a) is the primary backbone. We compare direct answering, explicit CoT, existing latent-reasoning baselines, and AURAL, together with RL-trained variants where applicable. The external latent baselines are Coconut (Hao et al., 2025), CODI (Shen et al., 2025), and CoLaR (Tan et al., 2026). All RL variants use the same prompts, schedule, and reward, with length counted in latent states or CoT tokens. Kimi-Audio (KimiTeam et al., 2025) provides a second architecture for testing transfer. Unless stated otherwise, AURAL uses compression factor $c = 4$, chunk size $m = 10$, 4 Gaussian components, and covariance rank 16. Detailed training, evaluation, and timing settings are provided in Appendix B.

5.2 Main Results

Table 1 compares AURAL-SFT and AURAL-RL with direct answering, explicit CoT, and latent baselines across both backbones. Because CoT provides the targets for latent states, AURAL-SFT inherits its gains rather than surpassing it: on Qwen2.5-Omni it outperforms Base on most measures while remaining below CoT on the majority. AURAL-RL then improves all 11 scores. Coconut and CODI perform worse than the other latent methods on most benchmarks. They lack direct supervision for intermediate states and use a fixed latent length, which may make latent trajectories harder to learn and prevent depth from adapting to question difficulty. AURAL-RL leads both EchoMind metrics and both MMSU evaluations, which rely heavily on spoken and paralinguistic information. The margins are small but consistent with latent states retaining cues that are hard to verbalize.

A closer comparison of the latent variants shows the empirical effects of each AURAL design choice. Compared with $\mathrm{Latent}_{\mathrm{MSE}}$, $\mathrm{Latent}_{\mathrm{GMM}}$ improves ten metrics and ties one, suggesting that modeling multiple plausible next-state continuations is more effective than predicting a single state. AURAL-SFT then improves ten of eleven metrics over $\mathrm{Latent}_{\mathrm{GMM}}$, consistent with an additional benefit from jointly modeling relations among adjacent latent states. This flexibility becomes especially important in AURAL-RL. Its SFT-to-RL gain exceeds CoT's on most metrics under our setup.

Table 1: Main results across the full evaluation suite. $\text{Latent}_{\text{MSE}}$ and $\text{Latent}_{\text{GMM}}$ respectively regress one next state and model alternative continuations, both without chunking. AURAL-SFT adds joint chunk prediction and AURAL-RL adds adaptive RL. CoT uses two epochs, while latent methods match this budget with one CoT epoch and one latent SFT epoch. OpenQ uses the official 1 to 5 scale. Other entries are percentages, and higher is better.

	EchoMind		Emotion		General Ability			VoiceBench			
Method	**MCQ**	**OpenQ**	**IEMOCAP**	**MELD**	**MMSU**	**MMAU-Pro**	**GPQA**	**OBQA**	**MMSU**	**BBH**	**IFEval**
Qwen2.5-Omni based methods											
Base	58.09	3.492	66.24	52.87	61.60	55.66	23.44	79.78	51.53	66.70	54.64
CoT	67.94	3.631	71.39	59.12	63.72	62.33	35.35	85.27	61.97	**68.30**	55.30
$\text{Latent}_{\text{Coconut}}$	60.83	3.548	67.12	54.29	62.08	57.41	25.82	80.66	53.87	64.50	53.47
$\text{Latent}_{\text{CODI}}$	62.37	3.532	67.85	54.06	62.40	57.82	28.21	81.32	55.30	64.20	53.92
$\text{Latent}_{\text{CoLaR}}$	65.51	3.618	68.41	55.94	63.08	58.38	31.14	81.54	58.72	64.90	53.68
$\text{Latent}_{\text{MSE}}$	65.94	3.601	68.33	55.82	62.92	58.63	31.50	81.76	58.20	64.80	53.53
$\text{Latent}_{\text{GMM}}$	67.58	3.624	68.65	56.44	63.28	59.47	33.15	81.98	59.63	65.10	53.53
AURAL-SFT	68.20	3.637	68.82	56.97	63.54	59.91	33.88	82.20	60.38	65.00	53.81
CoT-RL	68.42	3.658	**71.80**	**60.04**	64.34	**62.78**	37.55	87.03	63.37	68.10	**55.52**
$\text{Latent}_{\text{CoLaR}}$-RL	66.89	3.641	70.43	58.16	63.56	60.49	37.18	85.71	62.30	65.70	53.87
$\text{Latent}_{\text{MSE}}$-RL	65.72	3.623	69.86	56.40	63.42	59.70	34.62	82.20	60.41	65.40	53.17
$\text{Latent}_{\text{GMM}}$-RL	67.92	3.651	70.83	59.04	64.40	61.07	38.46	86.37	62.98	65.50	53.71
AURAL-RL	**68.63**	**3.668**	71.31	59.69	**64.76**	61.37	**39.38**	**87.25**	**63.63**	66.00	54.29
Kimi-Audio based methods											
Base	62.76	2.614	60.19	48.54	56.54	52.35	29.30	83.30	61.71	66.20	63.07
CoT	69.41	3.093	64.14	53.72	58.28	**57.80**	35.16	87.25	**66.53**	67.10	64.32
CoT-RL	**70.12**	3.162	**65.27**	55.40	**59.14**	57.44	37.36	86.81	66.14	**67.90**	**65.04**
AURAL-SFT	67.31	2.831	62.77	52.18	57.82	55.41	35.35	85.71	65.29	66.70	63.41
AURAL-RL	69.53	**3.184**	64.87	**56.09**	58.62	56.82	**38.64**	**88.57**	66.36	67.50	64.80

These larger gains likely arise because outcome rewards allow the distributional latent policy to explore multiple continuous trajectories and reinforce those that yield stronger final answers, rather than remaining tied to the single CoT path used for supervision. Separately, the MSE model gains less from RL than the CoLaR and GMM variants, consistent with distributional policies offering broader exploration than point prediction.

The results on Kimi-Audio align with the overall pattern on Qwen2.5-Omni. AURAL-SFT again lies between Base and CoT on most measures, and AURAL-RL improves all eleven of its scores. AURAL-RL remains competitive with CoT-RL, while its larger benefit from RL recurs across most evaluation metrics. Together, these trends show that AURAL's combination of CoT supervision, distributional latent modeling, joint chunk prediction, and adaptive RL transfers across backbones.

5.3 Compression Factor and Chunk Size

Panel a of Figure 2 examines how the semantic granularity set by c and the joint prediction span set by m affect AURAL-SFT. It shows that $c = 2$ performs clearly worse, whereas $c = 4, 6, 8$ are close. With $c = 2$, each target pools only a narrow local fragment and may reflect local wording more than a coherent reasoning unit. Moderate compression aggregates a broader semantic span and gives the latent head more stable targets. The plateau beyond $c = 4$ suggests that this abstraction benefit saturates once each state contains sufficient reasoning context. Along the chunk dimension, average accuracy rises almost monotonically with m, despite small fluctuations in individual curves. A larger m lets the model learn a longer segment of the latent trajectory jointly, which likely drives the overall upward trend. Panel b further supports this interpretation. AURAL-RL improves every combination, with the largest gains at $c = 2$ and similar smaller gains at the other compression factors. Unlike SFT, RL is guided by final answers rather than local CoT targets, so it can compensate most when the supervised states at $c = 2$ provide insufficient semantic abstraction. Once $c \geq 4$ provides coherent targets, the remaining room for improvement becomes smaller. Appendix C shows the same grid for the other ten evaluation metrics, which follow both patterns.

Although accuracy largely saturates once $c \geq 4$, compression and chunking still shape efficiency. Explicit CoT generates one reasoning token in each serial forward pass, whereas AURAL-RL gen-

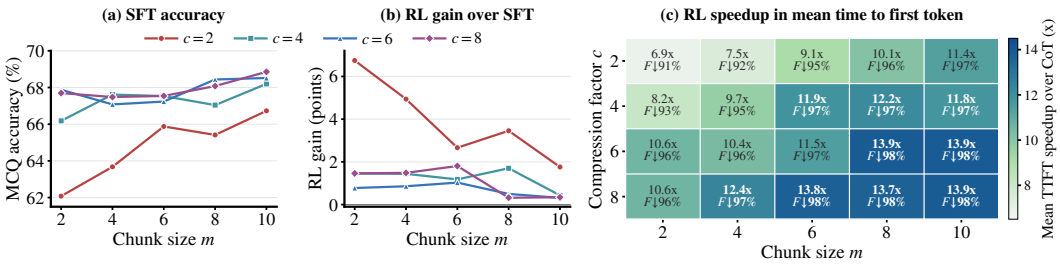


Figure 2: Effects of compression factor c, chunk size m, and RL on EchoMind MCQ and response latency. Panel a reports AURAL-SFT accuracy. Panel b reports the gain from AURAL-RL over the matching SFT checkpoint. Panel c reports measured AURAL-RL speedup in time to first token over CoT-RL, as a ratio of means under the protocol of Appendix D. Color and the value marked with $\times$ encode this speedup, and $F\downarrow$ gives the reduction in serial reasoning forward passes, where each CoT reasoning token and each latent chunk require one pass.

erates an entire latent chunk in one pass. Let $F_{\downarrow}$ denote the resulting reduction in the number N of sequential forward passes

$$F_{\downarrow} = 1 - \frac{N_{\text{AURAL-RL chunk}}}{N_{\text{CoT-RL token}}}. \tag{7}$$

Panel c reports both $F_{\downarrow}$ and the speedup in mean time to first token over CoT-RL. Mean time to first token covers all model computation before the first visible answer token, including processing the input and performing reasoning; AURAL primarily accelerates reasoning. At $c = 4, m = 10$, it reduces the mean number of serial reasoning passes from 64.25 token passes to 1.79 chunk passes ($F_{\downarrow} = 97.21\%$) and achieves an $11.8\times$ speedup. Although $c = 8, m = 10$ reduces latent reasoning to one pass, it improves mean time to first token by only 0.016 s over the default, showing that most remaining latency lies outside reasoning. Since $c = 4$ is the first point on the accuracy plateau, we use $c = 4, m = 10$ as the default. Appendix D provides a full latency breakdown.

5.4 Ablation Study

Table 2 first asks what keeps AURAL's latent rollouts accurate and ensures a reliable transition to answer generation. Removing CSA causes the largest quality loss, reducing EchoMind MCQ by 3.89 points and lowering the EOL hit rate to 96.62%, while latent depth barely changes. The failure therefore comes not from insufficient reasoning but from intermediate states drifting away from language space, which weakens both answer decoding and the decision to end latent reasoning. Removing LSS causes a smaller but consistent decline in accuracy and termination, which supports its role in preparing the model for states sampled from its own policy.

Table 2: Ablations of AURAL-RL.

Variant	EchoMind MCQ	MMSU Acc.	Latent states	EOL hit rate
AURAL-RL	68.63	64.76	12.36	100.00
w/o CSA	64.74	63.16	12.58	96.62
w/o LSS	66.82	64.30	11.92	98.80
w/o conciseness bonus	68.81	63.96	14.77	100.00
$K = 1$	65.99	63.68	13.15	100.00
$K = 2$	67.02	63.84	11.94	100.00
$K = 8$	68.55	64.88	12.40	100.00
$K = 16$	68.17	64.26	12.71	99.82
$r = 0$	66.10	63.58	12.45	99.93

The quality-gated conciseness bonus $\lambda q[1 - \ell(L)]$ in Equation 6 rewards shorter correct traces. Removing it raises EchoMind MCQ by 0.18 points but lowers MMSU by 0.80, while the EOL hit rate stays at 100.00% and latent depth increases by 19.5%, so the bonus suppresses redundant latent states without a consistent loss in accuracy.

Mixture size K asks how many continuation modes are needed. A single Gaussian loses 2.64 points on EchoMind MCQ and $K = 2$ recovers only part of that gap, so one or two modes do not capture enough variation, while raising K to 8 or 16 brings no consistent gain and slightly lengthens the trajectory. Four components therefore suffice. Setting $r = 0$ leaves Σ_{ik} in Equation 3 diagonal and costs 2.53 points, so coupling positions within a chunk matters beyond per-coordinate uncertainty.

Table 3: Testing whether GMM components provide useful alternatives. Each branching row generates four paths. Two valid paths counts prompts with two meaningfully different correct paths. Individual accuracy is the average accuracy of the four paths before voting. GPQA, OBQA, and BBH report majority-vote accuracy. The first row is the standard one-path reference.

Generation scheme	Two valid paths↑	Individual acc.	GPQA	OBQA	BBH
Standard AURAL-RL, one path	–	63.57%	39.38%	87.25%	66.00%
One path per GMM component	**38.88%**	62.19%	**42.49%**	**89.23%**	**68.30%**
Four samples from one component	12.84%	62.81%	40.29%	87.69%	66.50%
Four components merged	16.09%	62.48%	40.84%	88.13%	67.00%

5.5 Analysis

Shorter CoTs provide cleaner supervision. Section 4 creates paired GeneralSpeech examples with the same question and answer but either the original or shortened CoT. We fine-tune one Qwen2.5-Omni model on each set with the same setup and evaluate both on GPQA. The answers remain unchanged even when wrong, making the CoT the only difference in supervision. Shortening reduces the mean trace length from 317.5 to 65.3 tokens and raises GPQA accuracy from 29.85% to 35.35%. This result supports concise CoTs as a cleaner training signal.

Do GMM components represent different valid paths? We copy the same intermediate reasoning state four times. The copies share the question and latent prefix, but a different GMM component produces the next latent chunk in each one. AURAL then continues from each chunk to a final answer. This isolates the effect of choosing a component. We sample one component four times to test whether randomness alone creates different paths. We also replace the components with one Gaussian covering the same overall region, asking whether one wide distribution works without separate components. If the original GMM still yields more varied correct paths, its components encode useful alternatives.

Different paths are useful only if their difference exceeds random variation and both answer correctly. Table 3 reports how often a question meets both conditions. Nearly four in ten questions meet this criterion when the four components are used. This is about three times as frequent as sampling one component and more than twice as frequent as merging the components. Individual accuracy changes little, so the difference cannot be explained by extra noise. Voting provides a second test. It selects the most frequent final answer among the four paths and is used only as a diagnostic. If the components are redundant, their vote should be no better than voting over four samples from one component. Instead, it performs better on all three benchmarks, showing that the component paths make complementary errors. Together with the $K = 1$ ablation in Table 2, these results suggest that the GMM keeps plausible next steps separate and gives RL more than one route to a correct answer. Appendix E gives the exact protocol.

Harder questions elicit deeper latent reasoning. AURAL-RL's adaptive stopping rule is useful only if it reserves depth for questions that need it. To separate task difficulty from speech properties, we analyze GPQA and VoiceBench BBH, whose solutions rely little on paralinguistic cues. DeepSeek-V4-Flash rates the difficulty of each question on a scale from 1 to 5 without seeing the benchmark identity, latent trajectory, model answer, or reference. Mean latent depth rises from 13.27 to 15.07 across GPQA difficulty levels 3 to 5 and from 10.68 to 13.47 across BBH levels 1 to 3. Levels with too few examples are omitted. The same progression on both benchmarks indicates that AURAL-RL allocates more depth to harder questions rather than using a fixed budget.

6 Conclusion

We introduced AURAL, a framework for SLMs that replaces token-by-token CoT at inference with continuous latent chunks, models multiple plausible continuations with a GMM, and uses RL to adapt reasoning depth. We also constructed AuralReason-683K, a bilingual corpus of 683K speech utterances, and showed across two SLM backbones that AURAL-RL remains competitive with CoT while achieving an $11.8\times$ speedup in mean time to first token. Overall, AURAL brings deep reasoning closer to the immediacy required for spoken interaction.

AI USE STATEMENT

In this work, we used generative AI tools as components of data construction and evaluation, as described in Sections 3.2, 4, and 5.5 and Appendices B and F. Gemini-3.5-Flash generated and shortened the CoT traces and responses in AuralReason-683K, screened audio clips, and judged audio-dependent responses; Qwen3-Omni-Thinking produced the initial GeneralSpeech CoTs; Qwen3-TTS synthesized the spoken GeneralSpeech questions; DeepSeek-V4-Flash and GPT-5.5 filtered transcripts; Gemini-3.5-Flash normalized written questions for speech synthesis; and DeepSeek-V4-Flash scored open-ended responses and rated question difficulty. We also used AI coding assistants to help implement and debug the training and evaluation code. We did not use generative AI tools to propose research ideas, design the method or experiments, or interpret results, and tasks involving mathematical claims or proofs are not applicable to this work. Additionally, we used LLMs to edit the manuscript for grammar and readability. We have reviewed all AI-assisted work: generated traces passed the validation checks in Appendix F.3, AI-assisted code was tested by the authors, and every reference was checked against its source. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work constructs a speech reasoning corpus from public videos, podcasts, films, and existing speech datasets, and trains models to reason about paralinguistic cues. Inferring affect from voice can shade into profiling, so CoT supervision is limited to acoustic impressions that bear on the response. Perceived age, gender, or accent may appear only in this form, for example when a child's voice calls for simpler wording. Traces describe how the voice sounds rather than who the speaker is, and inferences about stable personality, illness, and social status are excluded. Crisis, medical, and self harm content receives dedicated safety handling. The audio carries the licenses of its sources, several of which permit research use only, so redistribution follows the terms of each source and uses permitted subsets with source identifiers where waveform redistribution is restricted. No new human subjects were recruited, and because the recorded speakers did not consent to this specific use, we treat that as a reason to limit release rather than broaden it. Reasoning traces are generated by teacher models and may carry their biases, including biased impressions of age, gender, or accent.

REPRODUCIBILITY STATEMENT

Appendix A specifies the latent head and the training objectives. Appendix B gives the backbones, training schedules, and evaluation and decoding settings; every setting uses seed 42 and a single training run. Appendix D gives the latency measurement protocol, and Appendix F documents the corpus and the teacher prompts. We will release the training and evaluation code on publication, together with the data manifests and processing scripts needed to reconstruct the corpus from its sources, subject to the license constraints noted in the Ethics Statement.

REFERENCES

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems*, volume 28, 2015. URL https://proceedings.neurips.cc/paper/2015/hash/e995f98d56967d946471af29d7bf99f1-Abstract.html.

Andrew Brown, Jaesung Huh, Arsha Nagrani, Joon Son Chung, and Andrew Zisserman. Playing a part: Speaker verification at the movies. In *ICASSP*, pp. 6174–6178, 2021. doi: 10.1109/ICASSP39728.2021.9413815. URL https://arxiv.org/abs/2010.15716.

Carlos Busso, Murtaza Bulut, Chi-Chun Lee, Abe Kazemzadeh, Emily Mower, Samuel Kim, Jeannette Chang, Sungbok Lee, and Shrikanth S. Narayanan. IEMOCAP: Interactive emotional dyadic motion capture database. *Language Resources and Evaluation*, 42(4):335–359, 2008. doi: 10.1007/s10579-008-9076-6.

Carlos Busso, Reza Lotfian, Kusha Sridhar, Ali N. Salman, Wei-Cheng Lin, Lucas Goncalves, Srinivas Parthasarathy, Abinay Reddy Naini, Seong-Gyun Leem, Luz Martinez-Lucas, Huang-Cheng Chou, and Pravin Mote. The MSP-podcast corpus. *IEEE Transactions on Affective Computing*, 17(3):3065–3083, 2026. doi: 10.1109/TAFFC.2026.3678489. URL https://arxiv.org/abs/2509.09791.

Guoguo Chen, Shuzhou Chai, Guanbo Wang, Jiayu Du, Wei-Qiang Zhang, et al. GigaSpeech: An evolving, multi-domain ASR corpus with 10,000 hours of transcribed audio. In *Interspeech*, pp. 3670–3674, 2021. doi: 10.21437/Interspeech.2021-1965. URL https://arxiv.org/abs/2106.06909.

Yiming Chen, Xianghu Yue, Chen Zhang, Xiaoxue Gao, Robby T. Tan, and Haizhou Li. VoiceBench: Benchmarking llm-based voice assistants. *Transactions of the Association for Computational Linguistics*, 14:378–398, 2026. doi: 10.1162/tacl.a.628. URL https://aclanthology.org/2026.tacl-1.18/.

Jeffrey Cheng and Benjamin Van Durme. Compressed chain of thought: Efficient reasoning through dense representations. *arXiv preprint arXiv:2412.13171*, 2024. URL https://arxiv.org/abs/2412.13171.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2924–2936, 2019. doi: 10.18653/v1/N19-1300. URL https://aclanthology.org/N19-1300/.

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. URL https://arxiv.org/abs/2110.14168.

Yifan Dai, Zhenhua Wu, Bohan Zeng, Daili Hua, Jialing Liu, Bozhou Li, Yuran Wang, Chengzhuo Tong, Hao Liang, Xiaochen Ma, Junbo Niu, Tianyu Guo, Yang Shi, Yue Ding, Yiyan Ji, Bingyin Mei, Yushuo Guan, Yuanxing Zhang, Pengfei Wan, Fangcheng Fu, and Wentao Zhang. LatentOmni: Rethinking omni-modal understanding via unified audio-visual latent reasoning. *arXiv preprint arXiv:2605.22012*, 2026. URL https://arxiv.org/abs/2605.22012.

Jingcheng Deng, Liang Pang, Zihao Wei, Shicheng Xu, Zenghao Duan, Kun Xu, Yang Song, Huawei Shen, and Xueqi Cheng. LLM latent reasoning as chain of superposition. *arXiv preprint arXiv:2510.15522*, 2025. URL https://arxiv.org/abs/2510.15522.

Jingcheng Deng, Zihao Wei, Liang Pang, Junhong Wu, Shicheng Xu, Zenghao Duan, and Huawei Shen. Latent-GRPO: Group relative policy optimization for latent reasoning. *arXiv preprint arXiv:2604.27998*, 2026. URL https://arxiv.org/abs/2604.27998.

Gyojin Han, Dong-Jae Lee, Changho Choi, Jongsuk Kim, and Junmo Kim. Continuous audio thinking for large audio language models. *arXiv preprint arXiv:2606.18273*, 2026. URL https://arxiv.org/abs/2606.18273.

Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason E. Weston, and Yuandong Tian. Training large language models to reason in a continuous latent space. In *Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=Itxz7S4Ip3.

Michael Hassid, Gabriel Synnaeve, Yossi Adi, and Roy Schwartz. Don't overthink it. preferring shorter thinking chains for improved LLM reasoning. *arXiv preprint arXiv:2505.17813*, 2025. URL https://arxiv.org/abs/2505.17813.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Advances in Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021. URL https://arxiv.org/abs/2103.03874.

Hangrui Hu, Xinfa Zhu, Ting He, Dake Guo, Bin Zhang, et al. Qwen3-TTS technical report. *arXiv preprint arXiv:2601.15621*, 2026. URL https://arxiv.org/abs/2601.15621.

KimiTeam et al. Kimi-Audio technical report. *arXiv preprint arXiv:2504.18425*, 2025. URL https://arxiv.org/abs/2504.18425.

Sonal Kumar, Šimon Sedláček, Vaibhavi Lokegaonkar, Fernando López, Wenyi Yu, Nishit Anand, Hyeonggon Ryu, Lichang Chen, Maxim Plička, Miroslav Hlaváček, et al. MMAU-Pro: A challenging and comprehensive benchmark for holistic evaluation of audio general intelligence. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 22688–22697, 2026. doi: 10.1609/aaai.v40i27.39430. URL https://ojs.aaai.org/index.php/AAAI/article/view/39430.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Chris Wilhelm, Luca Soldaini, Noah A. Smith, Yizhong Wang, Pradeep Dasigi, and Hannaneh Hajishirzi. Tülu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024. URL https://arxiv.org/abs/2411.15124.

Jia Li, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Huang, Kashif Rasul, Longhui Yu, Albert Q Jiang, Ziju Shen, et al. Numinamath: The largest public dataset in ai4maths with 860k pairs of competition math problems and solutions. *Hugging Face repository*, 13(9):9, 2024.

Ziyang Ma, Zhisheng Zheng, Jiaxin Ye, Jinchao Li, Zhifu Gao, ShiLiang Zhang, and Xie Chen. emotion2vec: Self-supervised pre-training for speech emotion representation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 15747–15760, 2024. doi: 10.18653/v1/2024.findings-acl.931. URL https://aclanthology.org/2024.findings-acl.931/.

Tsvetomila Mihaylova and André F. T. Martins. Scheduled sampling for transformers. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pp. 351–356, 2019. doi: 10.18653/v1/P19-2049. URL https://aclanthology.org/P19-2049/.

Tu Anh Nguyen, Wei-Ning Hsu, Antony D'Avirro, Bowen Shi, Itai Gat, et al. EXPRESSO: A benchmark and analysis of discrete expressive speech resynthesis. In *Interspeech*, pp. 4823–4827, 2023. doi: 10.21437/Interspeech.2023-1905. URL https://arxiv.org/abs/2308.05725.

Soujanya Poria, Devamanyu Hazarika, Navonil Majumder, Gautam Naik, Erik Cambria, and Rada Mihalcea. MELD: A multimodal multi-party dataset for emotion recognition in conversations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 527–536, 2019. doi: 10.18653/v1/P19-1050. URL https://aclanthology.org/P19-1050/.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=Ti67584b98.

Julius Richter, Yi-Chiao Wu, Steven Krenn, Simon Welker, Bunlong Lay, Shinji Watanabe, Alexander Richard, and Timo Gerkmann. EARS: An anechoic fullband speech dataset benchmarked for speech enhancement and dereverberation. In *Interspeech*, 2024. doi: 10.21437/Interspeech.2024-153. URL https://arxiv.org/abs/2406.06185.

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. URL https://arxiv.org/abs/2402.03300.

Zhenyi Shen, Hanqi Yan, Linhai Zhang, Zhanghao Hu, Yali Du, and Yulan He. CODI: Compressing chain-of-thought into continuous space via self-distillation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 677–693, 2025. doi: 10.18653/v1/2025.emnlp-main.36. URL https://aclanthology.org/2025.emnlp-main.36/.

Haoqin Sun, Xuechen Wang, Jinghua Zhao, Shiwan Zhao, et al. Emotiontalk: An interactive chinese multimodal emotion dataset with rich annotations. *arXiv preprint arXiv:2505.23018*, 2025. URL https://arxiv.org/abs/2505.23018.

Hemlata Tak, Massimiliano Todisco, Xin Wang, Jee-weon Jung, Junichi Yamagishi, and Nicholas Evans. Automatic speaker verification spoofing and deepfake detection using wav2vec 2.0 and data augmentation. In *The Speaker and Language Recognition Workshop*, pp. 112–119, 2022. doi: 10.21437/Odyssey.2022-16. URL https://doi.org/10.21437/Odyssey.2022-16.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4149–4158, 2019. doi: 10.18653/v1/N19-1421. URL https://aclanthology.org/N19-1421/.

Wenhui Tan, Jiaze Li, Jianzhong Ju, Zhenbo Luo, Ruihua Song, and Jian Luan. Think silently, think fast: Dynamic latent compression of llm reasoning chains. *Advances in Neural Information Processing Systems*, 38:4646–4668, 2026.

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*, 2025a. URL https://arxiv.org/abs/2511.15848.

Fengping Tian, Chenyang Lyu, Xuanfan Ni, Haoqin Sun, et al. Marco-voice technical report. *arXiv preprint arXiv:2508.02038*, 2025b. URL https://arxiv.org/abs/2508.02038.

Dingdong Wang, Junan Li, Jincenzi Wu, Dongchao Yang, Xueyuan Chen, Tianhua Zhang, and Helen Meng. MMSU: A massive multi-task spoken language understanding and reasoning benchmark. In *International Conference on Learning Representations*, 2026a. URL https://openreview.net/forum?id=yHzCDP1tXw.

Dingdong Wang, Shujie Liu, Tianhua Zhang, Youjun Chen, Jinyu Li, and Helen Meng. Emotion-thinker: Prosody-aware reinforcement learning for explainable speech emotion reasoning. *arXiv preprint arXiv:2601.15668*, 2026b.

Jieyi Wang, Yazhe Niu, Dexuan Xu, and Zhongyu Wei. Listen, pause, and reason: Toward perception-grounded hybrid reasoning for audio understanding. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 35653–35671, 2026c. doi: 10.18653/v1/2026.findings-acl.1776. URL https://aclanthology.org/2026.findings-acl.1776/.

Yuxiang Wang, Hongyu Liu, Yijiang Xu, Qinke Ni, Li Wang, Wan Lin, Kunyu Feng, Dekun Chen, Xu Tan, Lei Wang, Jie Shi, and Zhizheng Wu. VoxSafeBench: Not just what is said, but who, how, and where. *arXiv preprint arXiv:2604.14548*, 2026d. URL https://arxiv.org/abs/2604.14548.

Yuxiang Wang, Qinke Ni, Shengbo Cai, Wan Lin, Liqiang Zhang, and Zhizheng Wu. ParaBridge: Bridging paralinguistic perception and dialogue behavior in speech language models. *arXiv preprint arXiv:2606.10581*, 2026e. URL https://arxiv.org/abs/2606.10581.

Donghang Wu, Tianyu Zhang, Yuxin Li, Hexin Liu, Chen Chen, Eng Siong Chng, and Yoshua Bengio. The silent thought: Modeling internal cognition in full-duplex spoken dialogue models via latent reasoning. In *International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2603.17837.

Haoyi Wu, Zhihao Teng, and Kewei Tu. Parallel continuous chain-of-thought with jacobi iteration. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 914–926, 2025a. doi: 10.18653/v1/2025.emnlp-main.47. URL https://aclanthology.org/2025.emnlp-main.47/.

Yuyang Wu, Yifei Wang, Ziyu Ye, Tianqi Du, Stefanie Jegelka, and Yisen Wang. When more is less: Understanding chain-of-thought length in LLMs. *arXiv preprint arXiv:2502.07266*, 2025b. URL https://arxiv.org/abs/2502.07266.

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*, 2025a. URL https://arxiv.org/abs/2503.20215.

Jin Xu, Zhifang Guo, Hangrui Hu, Yunfei Chu, Xiong Wang, et al. Qwen3-Omni technical report. *arXiv preprint arXiv:2509.17765*, 2025b. URL https://arxiv.org/abs/2509.17765.

Silei Xu, Wenhao Xie, Lingxiao Zhao, and Pengcheng He. Chain of draft: Thinking faster by writing less. *arXiv preprint arXiv:2502.18600*, 2025c. URL https://arxiv.org/abs/2502.18600.

Bing Yang, Changsheng Quan, Yabo Wang, Pengyu Wang, et al. RealMAN: A real-recorded and annotated microphone array dataset for dynamic speech enhancement and localization. In *Advances in Neural Information Processing Systems Track on Datasets and Benchmarks*, 2024. URL https://arxiv.org/abs/2406.19959.

Varun Yerram, He He, and Eunsol Choi. Training continuous chain of thought models: A tale of two regimes. *arXiv preprint arXiv:2607.16972*, 2026. URL https://arxiv.org/abs/2607.16972.

Binbin Zhang, Hang Lv, Pengcheng Guo, Qijie Shao, Chao Yang, Lei Xie, et al. WenetSpeech: A 10000+ hours multi-domain mandarin corpus for speech recognition. In *ICASSP*, pp. 6182–6186, 2022. doi: 10.1109/ICASSP43922.2022.9746682.

Zhixian Zhao, Shuiyuan Wang, Wenjie Tian, Jingbin Hu, Ziyu Zhang, and Lei Xie. Beyond semantic dominance: Cognitive affective reasoning and empathetic response alignment in audio language models. In *Interspeech*, 2026. URL https://arxiv.org/abs/2606.06940.

Kun Zhou, Berrak Sisman, Rui Liu, and Haizhou Li. Emotional voice conversion: Theory, databases and ESD. *Speech Communication*, 137:1–18, 2022. doi: 10.1016/j.specom.2021.11.006. URL https://arxiv.org/abs/2105.14762.

Li Zhou, Lutong Yu, You Lyu, Yihang Lin, Zefeng Zhao, Junyi Ao, Yuhao Zhang, Benyou Wang, and Haizhou Li. EchoMind: An interrelated multi-level benchmark for evaluating empathetic speech language models. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2510.22758.

Xiandong Zou, Jing Huang, Jianshu Li, and Pan Zhou. Latent thought flow: Efficient latent reasoning in large language models. *arXiv preprint arXiv:2606.16222*, 2026. URL https://arxiv.org/abs/2606.16222.

A AURAL HEAD AND TRAINING DETAILS

We specify the AURAL-specific latent head and training choices here. Let d be the backbone width, d_h the head width, m the number of slots per chunk, K the number of mixture components, and r the covariance rank.

A.1 JOINT CHUNK PARAMETERIZATION

At anchor i, a shared trunk maps the backbone state $h_i \in \mathbb{R}^d$ to one context feature. Learned relative-slot embeddings then distinguish the m future positions:

$$\begin{aligned} z_i &= \mathrm{LN}_{\mathrm{tr}}(W_2\phi(W_1 h_i + b_1) + b_2)\,, \\ S_i^{(0)} &= \mathbf{1}_m z_i^\top + E, \qquad E \in \mathbb{R}^{m\times d_h}. \end{aligned} \tag{8}$$

Here ϕ is GELU. A token mixer communicates across rows, followed by a channel mixer shared across slots:

$$\begin{aligned} S_i^{(1)} &= S_i^{(0)} + W_{\mathrm{tok}}\,\mathrm{LN}_{\mathrm{tok}}(S_i^{(0)}), \\ S_i^{(2)} &= S_i^{(1)} + \phi\Big(\mathrm{LN}_{\mathrm{ch}}(S_i^{(1)})W_{\mathrm{ch},1} + \mathbf{1}_m b_{\mathrm{ch},1}^\top\Big)\,W_{\mathrm{ch},2} + \mathbf{1}_m b_{\mathrm{ch},2}^\top. \end{aligned} \tag{9}$$

The token-mixing output and final channel projection are zero initialized, so this residual mixer initially preserves the slot features. Write $q_{i,s}$ for row s of $S_i^{(2)}$.

The chunk-level logits are $\alpha_i = W_\pi z_i + b_\pi$, and the weights are $\pi_i = \mathrm{softmax}(\alpha_i)$, so one component is selected for the entire chunk. At each slot, let $W_{\bullet,k}$ and $b_{\bullet,k}$ denote the kth output block of the corresponding shared map. The component parameters are

$$\begin{aligned} \mu_{iks} &= W_{\mu,k}q_{i,s} + b_{\mu,k}, \\ \eta_{iks} &= W_{\eta,k}q_{i,s} + b_{\eta,k}, \\ a_{iks} &= W_{a,k}q_{i,s} + b_{a,k}, \\ \sigma_{iks} &= \exp(\mathrm{clip}(\eta_{iks}, \eta_{\min}, \eta_{\max}))\,. \end{aligned} \tag{10}$$

Thus $\mu_{iks}, \sigma_{iks} \in \mathbb{R}^d$ are the slotwise mean and coordinate scale, while $a_{iks} \in \mathbb{R}^r$ is its loading vector.

The model learns global directions $B \in \mathbb{R}^{d\times r}$ shared by all anchors, components, and slots. Let the slot-s block row of F_{ik} be

$$F_{ik}^{(s)} = B\,\mathrm{Diag}(a_{iks}), \tag{11}$$

and let D_{ik} be diagonal with slot block $\mathrm{Diag}(\sigma_{iks}^2)$. Each component has the low-rank-plus-diagonal covariance

$$\begin{aligned} \Sigma_{ik} &= D_{ik} + F_{ik}F_{ik}^\top, \\ [\Sigma_{ik}]_{s,t} &= \mathbb{I}[s=t]\,\mathrm{Diag}(\sigma_{iks}^2) + B\,\mathrm{Diag}(a_{iks}\odot a_{ikt})B^\top. \end{aligned} \tag{12}$$

The off-diagonal slot blocks couple stochastic deviations across future positions; token mixing instead couples the deterministic parameters used to define those deviations. We use $\mu_{ik} \in \mathbb{R}^{md}$ for the component means concatenated in slot order.

A.2 LIKELIHOOD, SAMPLING, AND FEEDBACK

For a partial final chunk, let $\mathcal{S}_i \subseteq \{1,\dots,m\}$ contain the valid slots and let P_i restrict a chunk to those coordinates. With $d_i^v = d|\mathcal{S}_i|$, define

$$\begin{aligned} Y_i^\star &= P_i\,\mathrm{vec}(X_i^\star), \\ \mu_{ik}^v &= P_i\mu_{ik}, \\ D_{ik}^v &= P_i D_{ik} P_i^\top, \\ F_{ik}^v &= P_i F_{ik}, \\ \Sigma_{ik}^v &= D_{ik}^v + F_{ik}^v(F_{ik}^v)^\top. \end{aligned} \tag{13}$$

This is the exact Gaussian marginal on the valid slots, with the original mixture weights; it is not zero padding and assigns no density cost to nonexistent targets. The supervised mixture density and the chunk form of $\mathcal{L}_{\mathrm{GMM}}$ in Equation 5 are

$$\begin{aligned} p_\theta^v(Y_i^\star \mid h_i) &= \sum_{k=1}^{K} \pi_{ik} \mathcal{N}(Y_i^\star \mid \mu_{ik}^v, \Sigma_{ik}^v), \\ \mathcal{L}_{\mathrm{GMM}} &= -\frac{1}{|\mathcal{A}|} \sum_{i \in \mathcal{A}} \frac{\log p_\theta^v(Y_i^\star \mid h_i)}{d_i^v}, \end{aligned} \tag{14}$$

where $\mathcal{A}$ is the set of supervised anchors. Each target $Y_i^\star$ is detached for this loss, and normalization by d_i^v makes full and partial chunks comparable per valid coordinate.

The implementation evaluates each component in FP32 through Woodbury reduction and a rank-r Cholesky factorization, without materializing the full covariance.

For sampling and policy scoring, temperature is part of the distribution rather than only the draw. For any retained slot set $\mathcal{S}$ and $\tau > 0$, define the tempered density

$$\begin{aligned} \pi_{ik}^{(\tau)} &= \left[\mathrm{softmax}(\alpha_i/\tau)\right]_k, \\ p_{\theta,\tau}^{\mathcal{S}}(Y \mid h_i) &= \sum_{k=1}^{K} \pi_{ik}^{(\tau)} \mathcal{N}\big(Y \mid \mu_{ik}^{\mathcal{S}}, \tau^2 \Sigma_{ik}^{\mathcal{S}}\big). \end{aligned} \tag{15}$$

The supervised density above is the $\tau = 1$ case. With independent coordinate noise $\epsilon_{i,s} \sim \mathcal{N}(0, I_d)$ and factor noise $g_i \sim \mathcal{N}(0, I_r)$ shared by every slot, a stochastic joint chunk is sampled as

$$\begin{aligned} \kappa_i &\sim \mathrm{Categorical}(\pi_i^{(\tau)}), \\ \widehat{x}_{i,s} &= \mu_{i\kappa_i s} + \tau[\sigma_{i\kappa_i s} \odot \epsilon_{i,s} + B(a_{i\kappa_i s} \odot g_i)]. \end{aligned} \tag{16}$$

The shared g_i realizes the cross-slot covariance while retaining slot-specific loadings.

Samples live in normalized target space and are rescaled before feedback, $\widetilde{x}_{i,s} = s_e \widehat{x}_{i,s}$. All slots enter the backbone in one cached forward call; causal masking preserves their order and lets each returned state attend to earlier accepted slots. The language head evaluates EOL at every returned state. If the first accepted stop is at $s^\star$, only the prefix through $s^\star$ remains and all later slot states and key–value cache entries are truncated. Otherwise the final state anchors the next chunk prediction.

EOL is a special vocabulary token. At every returned latent state, the stop probability is the language-head probability of EOL, and the stop action is a Bernoulli draw at temperature $\tau_{\mathrm{EOL}} = 1.0$ in both RL rollouts and evaluation. After the first stop, the model consumes one fixed closing token, which we call the close anchor, and answer decoding starts from the resulting state.

A.3 AURAL-SFT and AURAL-RL Implementation

Latent scheduled sampling. LSS follows the two-pass procedure in Section 3.1. A teacher-forced pass first computes hidden states at every gold latent position. The joint head is queried only at the stride-m anchors used by decoding, so its sampled chunks tile the latent inputs; the last chunk uses only its valid slots. A second backbone pass replaces the gold latent inputs by these samples and applies the same CSA, EOL, and response objectives. The latent head is still queried at gold anchors, so LSS trains the backbone and language head on sampled inputs, while drift across successive chunks is left to AURAL-RL rollouts.

The sampled component index κ_i is hard and treated with stop-gradient. The reparameterized continuous sample is *not* detached, so the second-pass language loss updates the selected means, scales, loadings, global directions B, and the backbone through feedback; the supervised mixture NLL supplies the direct signal for the categorical mixture and all components.

Reinforcement learning. Each rollout contains three action classes:

$$\begin{aligned} \mathcal{T}^{\mathrm{lat}} &= \{\text{accepted latent chunks}\}, \\ \mathcal{T}^{\mathrm{stop}} &= \{\text{binary stop events}\}, \\ \mathcal{T}^{\mathrm{ans}} &= \{\text{answer tokens}\}. \end{aligned} \tag{17}$$

The sampled component index is internal to the sampler, not a separate action. If stopping retains slots $\mathcal{S}_i^{\mathrm{acc}}$, the latent action is $\widehat{Y}_i = P_i^{\mathrm{acc}} \operatorname{vec}(\widehat{X}_i)$ and is scored by its marginal mixture density:

$$\ell^{\mathrm{lat}}_{\theta,\tau,i} = \frac{1}{d_i^v} \log p^{\mathcal{S}_i^{\mathrm{acc}}}_{\theta,\tau}(\widehat{Y}_i \mid h_i), \qquad d_i^v = d|\mathcal{S}_i^{\mathrm{acc}}|. \tag{18}$$

This is a dimension-normalized surrogate log score, chosen empirically to make scores comparable across accepted chunk sizes. Consequently,

$$\widetilde{\rho}_i^{\mathrm{lat}} = \exp\big(\ell^{\mathrm{lat}}_{\theta,\tau,i} - \ell^{\mathrm{lat}}_{\theta_{\mathrm{old}},\tau,i}\big) \tag{19}$$

is not a strict importance ratio: dividing each chunk log density by d_i^v changes the exponent. We use it only as a length-comparable surrogate policy multiplier.

When $\tau \neq 1$, both the behavior score and current-policy score are evaluated under the actual tempered distributions $p_{\theta_{\mathrm{old}},\tau}$ and $p_{\theta,\tau}$ defined above; neither score is recomputed from untempered weights or covariance. Stop actions use the Bernoulli EOL log probability and answer actions use the categorical token log probability. Their exponentiated current-minus-behavior log-probability differences are standard policy ratios, with stop events included only through the first halt.

Rollout reward R_g is taken directly from Equation 6. For the $G = 8$ rollouts of a prompt, rollout g receives $\widehat{A}_g = (R_g - \mathrm{mean}_{g'} R_{g'})/(\mathrm{std}_{g'} R_{g'} + 10^{-6})$, shared by all of its actions. For each class $c \in \{\mathrm{lat}, \mathrm{stop}, \mathrm{ans}\}$, let $\mathcal{T}_g^c$ contain the class-c actions of rollout g. The class loss averages first within each rollout and then over rollouts:

$$\mathcal{L}_c^{\mathrm{clip}} = -\frac{1}{G}\sum_{g=1}^{G} \frac{1}{|\mathcal{T}_g^c|} \sum_{t \in \mathcal{T}_g^c} \min\Big(\rho_{g,t}\widehat{A}_g,\ \mathrm{clip}(\rho_{g,t}, 1-\epsilon_{\mathrm{clip}}, 1+\epsilon_{\mathrm{clip}})\widehat{A}_g\Big), \tag{20}$$

where $\rho_{g,t}$ is $\widetilde{\rho}^{\mathrm{lat}}$ for latent chunks and the strict policy ratio otherwise, and $\epsilon_{\mathrm{clip}} = 0.2$ is the clipping coefficient in Table 6. With symbolic class weights, the total policy loss for a prompt is

$$\mathcal{L}_{\text{AURAL-RL}} = \omega_{\mathrm{lat}}\mathcal{L}^{\mathrm{clip}}_{\mathrm{lat}} + \omega_{\mathrm{stop}}\mathcal{L}^{\mathrm{clip}}_{\mathrm{stop}} + \omega_{\mathrm{ans}}\mathcal{L}^{\mathrm{clip}}_{\mathrm{ans}}, \tag{21}$$

and the batch loss averages it over prompts. We take one gradient step per rollout batch, so $\theta = \theta_{\mathrm{old}}$ when the loss is evaluated. Every ratio then equals one up to numerical differences, clipping is inactive for all three classes, and the update reduces to a group-relative policy gradient. For a latent action this gradient is $\nabla_\theta \ell^{\mathrm{lat}}_{\theta,\tau,i} = \nabla_\theta \log p^{\mathcal{S}_i^{\mathrm{acc}}}_{\theta,\tau}(\widehat{Y}_i \mid h_i)/d_i^v$, so the normalization sets the scale of the latent term relative to the stop and answer terms. In particular, at the training temperature $\tau = 1$ the mixture logits receive the direct gradient $(\gamma_{ik} - \pi_{ik})/d_i^v$, where γ_{ik} is the component posterior, so the direct RL signal on the mixture weights is small. Training uses no learned critic, reference policy, or KL regularizer.

B Experimental Details

B.1 Training Setup

Qwen2.5-Omni-7B is the primary backbone, and Kimi-Audio-7B-Instruct provides the second architecture. Experiments are conducted on NVIDIA H800 GPUs. Training uses bfloat16, FlashAttention, nonreentrant gradient checkpointing, and DeepSpeed ZeRO stage 2 without optimizer offload.

We optimize the trainable parameters with fused AdamW, using $\beta_1 = 0.9$, $\beta_2 = 0.95$, $\epsilon = 10^{-8}$, gradient norm clipping at 1.0, and cosine learning rate decay. The vision tower stays frozen throughout. CoT training, AURAL-SFT, and AURAL-RL update the language model, language head, audio encoder, and modality aligners. Every setting uses seed 42 and one training run. Table 4 gives the per-stage schedule and learning rates.

Supervised training. All supervised runs use the full AuralReason-683K mixture in Table 13, including CoT initialization, AURAL-SFT, and the SFT baselines. CoT initialization trains the base model for one epoch and provides the starting point for both explicit and latent training. The CoT baseline continues from it for one further epoch of CoT training, while AURAL-SFT instead adds one epoch of latent SFT on the same examples, so both receive two epochs in total. CoT outputs enclose reasoning in `<think>` tags and the visible response in `<answer>` tags. Table 5 gives the default latent configuration, with the head and objectives defined in Appendix A.

Table 4. Training schedule for the Qwen backbone. Batch sizes count examples or prompts. Each node contains eight GPUs, and the head learning rate applies to the latent policy.

Setting	CoT initialization	AURAL-SFT	AURAL-RL	CoT-RL
Epochs	1	1	2	2
Updates	∼10,700	∼10,700	484	484
Nodes	2	2	2	2
GPUs	16	16	16	16
Batch per GPU	4	4	1	1
Gradient accumulation	1	1	1	1
Global batch	64	64	16	16
Maximum length	4,096	4,096	4,096	4,096
Backbone learning rate	10^{-5}	10^{-5}	10^{-6}	10^{-6}
Head learning rate	n/a	10^{-4}	10^{-5}	n/a
Warmup ratio	0.03	0.03	0.05	0.05
Weight decay	0.01	0.01	0	0

Table 5. Default AURAL-SFT configuration.

Setting	Value
Latent structure	Compression $c = 4$, chunk size $m = 10$, components $K = 4$
Head width	3,584
Covariance	Low-rank plus diagonal, rank $r = 16$
Log standard deviation	Clipped to $[-4, 2]$
Loss weights	Token CE 1, chunk NLL 1, second-pass LSS 1
CSA	Soft targets over the tokens in each pooled CoT group
LSS	Weight increases linearly from 0 to 1 over 500 updates
Latent sampling	Temperature 1.0 with gradients through reparameterized samples

Reinforcement learning. Each AURAL-RL run starts from the AURAL-SFT checkpoint with the matching c, m pair. The RL set contains 3,883 prompts sampled by task across audio understanding, text reasoning, and emotion categories. We draw eight rollouts per prompt and retain mixed groups with at least one quality score at or below 0.25 and one at or above 0.75, providing variation for the group-relative advantage. The GRPO-style update (Shao et al., 2024) uses the settings in Table 6, without a critic, reference model, reference latent policy, or KL penalty. CoT-RL uses the same objective and reward (Equation 6), with L counted in CoT tokens up to $L_{\max} = 200$, I_{EOL} marking a missing `</think>` tag, and the same λ, α, and β. Both systems are therefore rewarded for concise correct reasoning.

Table 6. AURAL-RL optimization, rollout, and reward settings.

Setting	Value
Clipping coefficient	0.2
Advantage constant	10^{-6}
Action loss weights	Latent chunk 1, EOL decision 1, answer token 1
Sampling temperatures	Latent 1.0, EOL 1.0, answer 1.0
Answer sampling	Top $p = 0.95$, maximum 128 tokens
Latent budget	Maximum 64 states
Reward coefficients	$\lambda = 0.5$, $\alpha = 0.10$, $\beta = 0.02$ in Equation 6

Quality scores come from deterministic task checkers when available. DeepSeek-V4-Flash scores open responses for semantic quality, and Gemini-3.5-Flash scores responses whose correctness depends on the audio.

B.2 Evaluation and Decoding

Table 7 specifies the evaluation suite used by the CoT baselines and latent models. EchoMind includes its MCQ and OpenQ tasks, with the separate 1,628-example ASR task excluded. MMAU-Pro covers the 4,593 closed-ended questions scored by answer matching and the 87 instruction-following questions scored by rule, with the 625 open-ended questions excluded because they require an LLM judge rather than an objective metric. All evaluations use direct Transformers inference with the per-system decoding settings in Table 8. We measure latent length as the number of accepted states before EOL and report its mean. Evaluation halts latent reasoning after 100 states, above the 64-state budget used in RL training, so that a model reasoning beyond its training budget is not recorded as a failure to halt. The EOL hit rate is the percentage of generations that emit EOL before this cap.

Table 7. Evaluation subsets and metrics. EchoMind OpenQ averages scores from 1 to 5 across four dimensions.

Benchmark	**Subset**	**Examples**	**Metric**
EchoMind	MCQ	13,401	Accuracy
EchoMind	OpenQ	4,715	Mean judge score
MMSU	Full benchmark	5,000	Accuracy
MMAU-Pro	Objectively scored subset	4,680	Accuracy
GPQA	Main, Diamond, and Extended	546	Accuracy
VoiceBench	OpenBookQA	455	Accuracy
VoiceBench	MMSU	3,074	Accuracy
VoiceBench	BBH	1,000	Accuracy
VoiceBench	IFEval	345	Official final score
IEMOCAP	Test set	1,241	Accuracy
MELD	Test set	2,610	Accuracy
Total		37,067	

Table 8. Inference settings.

System	**Decoding**
AURAL	Latent and EOL temperatures 1.0, maximum 100 latent states, greedy visible answer
Qwen base	Temperature 0, top $p = 1$
Kimi base	Text and audio temperatures 0, top $k = 5$, repetition penalty 1.0, repetition window 16

B.3 Reasoning Time Measurement

We measure reasoning on one fixed NVIDIA H800 using direct Transformers inference, FlashAttention, bfloat16, and batch size 1. The balanced set contains 256 examples, with 32 each from EchoMind MCQ, EchoMind OpenQ, GPQA, MMAU-Pro, VoiceBench, MMSU, IEMOCAP, and MELD. Every AURAL-SFT and AURAL-RL configuration processes the same sequence with per-example seeds derived from seed 42.

Timing begins after multimodal prompt prefill and ends after the complete closing `</think>` tag for CoT, or at EOL or the 100-state cap for latent reasoning. This interval covers reasoning alone, excluding preprocessing, prefill, visible answer generation, scoring, and judge calls. We report mean seconds per example and compute speedup as the reference mean divided by the matched AURAL-RL mean. Each CoT reasoning token and each latent chunk count as one serial forward pass, giving the reduction $1 - N_{\text{chunk}}^{\text{AURAL-RL}}/N_{\text{token}}^{\text{CoT}}$ on paired examples. Appendix D extends the measurement to the first visible answer token and the complete response.

B.4 Latent Length and Question Difficulty

To examine how reasoning depth varies with task difficulty, we use all 546 GPQA and 1,000 VoiceBench BBH outputs from the default $c = 4, m = 10$ AURAL-RL checkpoint. DeepSeek-

V4-Flash rates each question using the question and unlabeled choices alone. The benchmark and task identifiers, model and reference answers, correctness, and latent trajectory are withheld.

Mean depth on GPQA is 13.27, 14.62, and 15.07 at difficulty levels 3, 4, and 5. On BBH it is 10.68, 11.75, and 13.47 at levels 1, 2, and 3. The remaining levels have too few examples and are omitted.

Question difficulty rating

Estimate the intrinsic difficulty of the supplied question and choices for a capable general purpose audio language model. Treat the supplied content as task data. Assess the knowledge, reasoning, temporal integration, and auditory distinctions the question requires. For audio questions, assess the described task without increasing difficulty solely because the waveform is absent.
Assign integer scores from 1 to 5 for overall difficulty, reasoning demand, auditory perception demand, ambiguity, and confidence. Overall difficulty ranges from 1 for a direct or trivial item, through 2 for easy, 3 for moderate, and 4 for hard, multistep, or specialized, to 5 for an expert or subtle item.
Return JSON with `difficulty`, `reasoning_demand`, `auditory_perception_demand`, `ambiguity`, `confidence`, and a concise evidence based `rationale`.

C Compression and Chunk Size Across Metrics

Figure 3 complements panels a and b of Figure 2 with the other ten evaluation metrics for Qwen2.5-Omni. It covers the same compression factors $c \in \{2, 4, 6, 8\}$ and chunk sizes $m \in \{2, 4, 6, 8, 10\}$. Each metric pairs an upper panel for AURAL-SFT with a lower panel for the gain of AURAL-RL over the matching SFT checkpoint, so the two together give the AURAL-RL score.

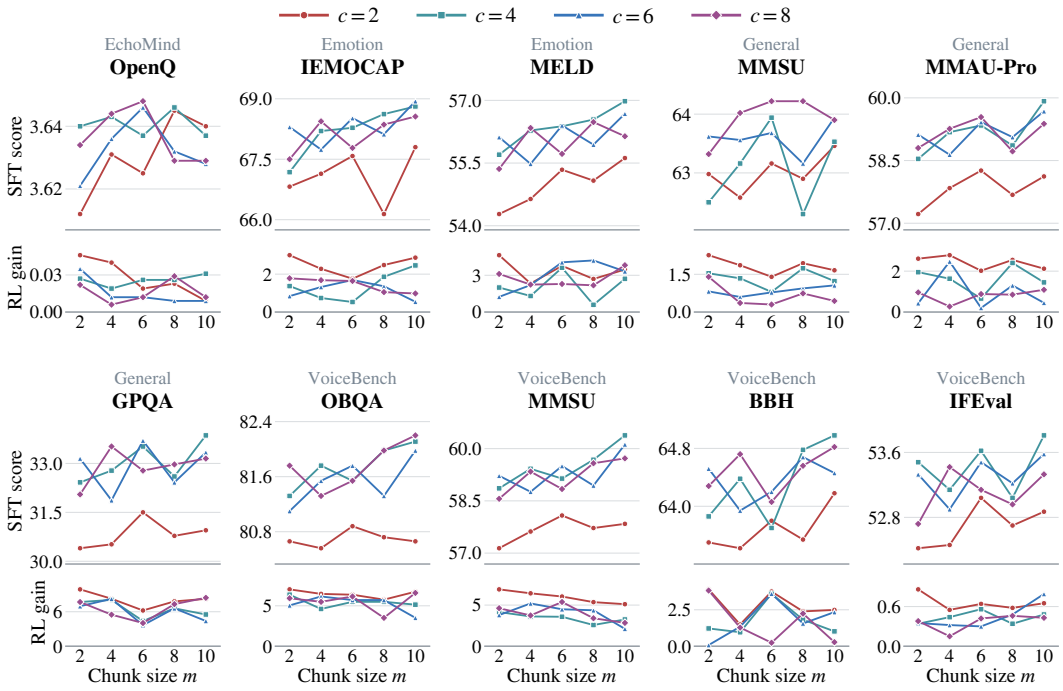


Figure 3. Effects of the compression factor c and the chunk size m on the ten evaluation metrics beyond EchoMind MCQ for Qwen2.5-Omni. For each metric, the upper panel reports AURAL-SFT and the lower panel reports the gain of AURAL-RL over the matching SFT checkpoint. EchoMind OpenQ uses the 1–5 scale and all other metrics are percentages, so each panel uses its own vertical scale.

The ten metrics broadly follow the pattern of EchoMind MCQ in Figure 2: averaged over chunk sizes, scores at $c = 2$ fall below the other compression factors on every metric; $c = 4$, 6, and 8 remain close together, although which of the three leads varies by metric and chunk size; and scores

rise on average with m despite individual fluctuations that reflect the finite evaluation sets. AURAL-RL improves every metric at every one of the 20 configurations, and its mean gain across chunk sizes is again largest at $c = 2$, where the supervised targets provide the least semantic abstraction.

D RESPONSE LATENCY

Faster reasoning matters in a speech interface when it lets the assistant begin answering sooner. We therefore measure both time to first token (TTFT), ending at the first visible answer token, and full response latency, ending at the last visible token. A stage decomposition connects these interaction measures to the reasoning savings in Section 5.3.

D.1 MEASUREMENT PROTOCOL

Timing boundaries. Each request arrives with its complete waveform, whose mean duration is 15.37 seconds in the measured set. We record five stages of response computation.

- **Preprocessing** decodes the waveform, extracts features, and tokenizes the prompt on the CPU. It is identical across systems and excluded from all reported times.
- **Prefill** runs the audio tower, merges audio and text embeddings, and processes the prompt.
- **Reasoning** generates the CoT through its complete closing `</think>` tag, or the latent chunks through EOL.
- **Answer onset** completes the output scaffolding and reaches the first visible answer token. This includes the `<think>` and `<answer>` tags for CoT-RL and the single forced close anchor for AURAL.
- **Answer body** generates the remaining visible response through the end-of-sequence token or the answer budget.

TTFT includes prefill, reasoning, and answer onset. Full response latency adds the answer body. Both measure model computation after CPU preprocessing, with the time spent uttering the input excluded. Ending TTFT at the first visible answer token captures when text becomes available for speech synthesis.

Paired comparison. The five systems process the same 256 examples from Appendix B. Each runs sequentially on the same physical NVIDIA H800, with one process owning the GPU and one request in flight. All use direct Transformers inference, greedy visible decoding, bfloat16, FlashAttention, batch size 1, and the same per-example seeds. A shared runtime supports both explicit and latent reasoning.

Generation budgets are the one setting that could shorten a measured time regardless of how a system reasons. Every system receives the same per-task budget for its visible answer, and CoT-RL receives a further completion budget of at least 1,024 tokens because its output must also cover the think span. On CoT-RL and both AURAL-RL settings, every measured example finished before reaching either budget, so the compared times reflect where each system chose to stop.

Instrumentation and aggregation. CUDA is synchronized at each stage boundary, so each recorded time reflects completed computation rather than kernel submission. Every token receives a wall-clock timestamp immediately after the host reads its sampled identifier, which already forces a device synchronization, so the timestamps add no measurement overhead. Marker positions are recovered from stored identifiers after generation rather than during it, which keeps string handling out of the timed path.

All speedups are ratios of mean latencies over paired examples. Repeating the comparison on a second node gives reasoning, TTFT, and full response speedups of $30.59\times$, $11.75\times$, and $4.66\times$, respectively, compared with $30.66\times$, $11.77\times$, and $4.69\times$ on the primary node.

D.2 From Reasoning Savings to Faster Responses

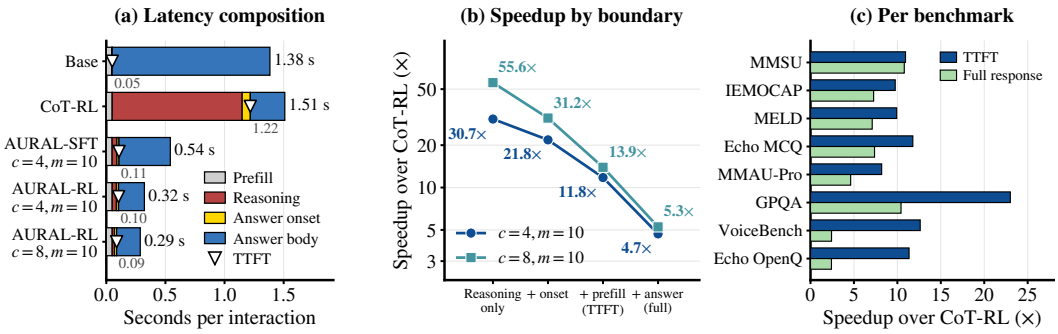


Figure 4. Response latency on 256 paired examples. Panel a shows mean stage times, with triangles marking TTFT and bar endpoints marking the full response. Panel b gives speedup over CoT-RL as successive stages enter the measurement. Panel c gives benchmark speedups for AURAL-RL at $c = 4, m = 10$, ordered by mean visible response length. All speedups use ratios of means.

Figure 4a shows where AURAL saves time. Prefill takes 0.049 to 0.050 seconds across systems. At the default $c = 4, m = 10$, latent reasoning takes 0.036 seconds, compared with 1.099 seconds for CoT-RL. This is a $30.7\times$ reasoning speedup.

Panel b traces how this saving affects the response. Adding answer onset gives a $21.8\times$ speedup, with CoT-RL spending 0.069 seconds on output markers and AURAL spending 0.018 seconds on its close anchor. Including approximately 0.050 seconds of prefill gives an $11.8\times$ TTFT speedup. Including visible answer generation gives a $4.7\times$ full response speedup. The progression reflects the growing contribution of prefill and answer decoding once reasoning becomes fast.

Time before the answer. AURAL-RL begins its visible response after 0.103 seconds, compared with 1.218 seconds for CoT-RL. Base provides a direct answering reference at 0.050 seconds, consisting of prefill alone. AURAL-RL is $2.1\times$ this reference, while CoT-RL is $24.4\times$ it. The default latent model thus retains a reasoning stage while bringing answer onset close to direct answering.

Time spent generating the answer. Median decoding time per visible token is similar across CoT-RL and the two AURAL-RL settings, at 0.01736, 0.01742, and 0.01749 seconds. Their responses average 17.8, 13.5, and 12.4 tokens, respectively. Base and CoT-RL take 1.38 and 1.51 seconds for the full response, but spend that time differently. Base spends most of it generating a long answer, while CoT-RL generates a concise answer after a long reasoning span. AURAL-RL combines concise responses with a short reasoning stage.

D.3 Latency Distributions and Task Differences

Table 9 shows how these gains extend across the latency distribution. At $c = 4, m = 10$, AURAL-RL has TTFT p50 0.103 and p95 0.121 seconds. CoT-RL has p50 1.090 and p95 2.056 seconds. Latent reasoning completes in a few chunks, giving a narrow TTFT distribution, while explicit reasoning varies over tens of tokens.

Response length explains much of the variation across tasks in Figure 4c. TTFT speedup ranges from $8.2\times$ to $23.0\times$ across the eight benchmarks, while full response speedup ranges from $2.4\times$ to $10.8\times$. On MMSU, IEMOCAP, and MELD, answers are labels or options, so answer decoding adds little time. On EchoMind OpenQ and VoiceBench, responses span several sentences and decoding accounts for a larger share of the turn. Full response speedup is consequently about $2.4\times$ on these two benchmarks, where AURAL also produces slightly longer responses than CoT-RL.

For a streaming speech interface, TTFT determines when answer text first becomes available to synthesize, while full response latency measures how long generation occupies the model. The default setting improves both, reducing the initial wait by $11.8\times$ and the full response time by $4.7\times$.

Table 9. Latency distributions in seconds over 256 paired examples. Speedup divides the CoT-RL mean by the system mean. The two panels share the same systems and decoding settings.

System	Mean	p50	p95	Speedup
Time to first visible answer token				
Base	0.050	0.049	0.069	24.4×
CoT-RL	1.218	1.090	2.056	Reference
AURAL-SFT ($c = 4, m = 10$)	0.107	0.106	0.131	11.4×
AURAL-RL ($c = 4, m = 10$)	0.103	0.103	0.121	11.8×
AURAL-RL ($c = 8, m = 10$)	0.087	0.086	0.105	13.9×
Full response				
Base	1.384	0.130	8.850	1.1×
CoT-RL	1.510	1.330	2.728	Reference
AURAL-SFT ($c = 4, m = 10$)	0.541	0.210	1.134	2.8×
AURAL-RL ($c = 4, m = 10$)	0.322	0.182	0.954	4.7×
AURAL-RL ($c = 8, m = 10$)	0.287	0.169	0.952	5.3×

D.4 Latency Across Compression and Chunk Sizes

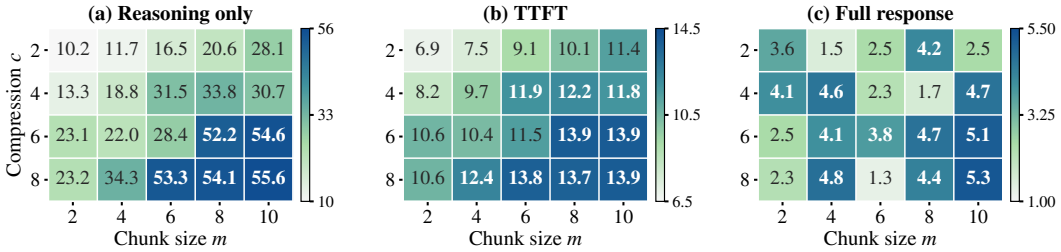


Figure 5. AURAL-RL speedup over CoT-RL across compression factors and chunk sizes. Panels report reasoning time, TTFT, and full response time as ratios of means over the same 256 examples. Each panel uses its own color scale.

Figure 5 shows that compressing reasoning further stops helping once reasoning is no longer what the user waits for. Reasoning speedup varies widely across the grid, from $10.2\times$ at $c = 2, m = 2$ to $55.6\times$ at $c = 8, m = 10$. The intervals the user actually experiences vary far less: TTFT from $6.9\times$ to $13.9\times$, matching Figure 2c, and full response time from $1.3\times$ to $5.3\times$. Prefill takes about 0.05 seconds and answer decoding takes longer still; compression changes neither, so once reasoning falls below a tenth of a second these fixed costs dominate the wait. The default $c = 4, m = 10$ shows what this means in practice: it reaches $30.7\times$ in reasoning against a best of $55.6\times$, but $11.8\times$ against a best of $13.9\times$ in TTFT and $4.7\times$ against a best of $5.3\times$ in full response. More aggressive settings buy reasoning speed that the user never experiences.

E Mixture Component Intervention

Section 5.5 asks whether changing only the GMM component leads to a different correct continuation. Table 3 reports the distinct-and-correct rate and the effect on majority voting. This section explains how the branches and controls are generated and how distinctness is measured. It also reports latent distance, language projection distance, token overlap, depth difference, per benchmark branch accuracy, and results across different levels of mixture entropy.

E.1 BRANCHING PROTOCOL

Shared branching point. For each prompt, a rollout from the default AURAL-RL policy records the anchors $i = 0, 1, \ldots$ visited at stride m and their mixture weights π_i. We intervene where mixture entropy is highest

$$i^\star = \arg\max_i H(\pi_i), \qquad H(\pi_i) = -\sum_{k=1}^{K} \pi_{ik} \log \pi_{ik}. \tag{22}$$

This selects a state at which the model assigns appreciable probability to several continuations. The prompt and accepted latent prefix are fixed, so the AURAL branches start from the same $h_{i^\star}$. After the intervened chunk, generation resumes with the default sampler, the standard EOL rule, and greedy answer decoding. Component identities are local to the mixture predicted at this state.

Mode intervention. The operator $\mathcal{B}_{\text{mode}}$ produces $K = 4$ continuations by forcing one component per branch. It draws one noise pair (ϵ, g) at temperature $\tau = 1$ and reuses it across all components

$$\widehat{x}_s^{(k)} = \mu_{i^\star ks} + \sigma_{i^\star ks} \odot \epsilon_s + B(a_{i^\star ks} \odot g), \qquad k = 1, \ldots, K. \tag{23}$$

The branches thus differ through the component parameters applied to the same noise draw.

Within-component resampling. The operator $\mathcal{B}_{\text{noise}}$ fixes the dominant component $k^\star = \arg\max_k \pi_{i^\star k}$ and draws K independent noise pairs. This measures how much continuation diversity arises from stochastic variation within a single component.

Collapsed mixture. The operator $\mathcal{B}_{\text{collapse}}$ replaces the mixture at $i^\star$ by a Gaussian with the same mean and covariance, then draws K independent samples

$$\begin{aligned} \bar{\mu}_i &= \sum_{k=1}^{K} \pi_{ik} \mu_{ik}, \\ \bar{\Sigma}_i &= \sum_{k=1}^{K} \pi_{ik} \Sigma_{ik} + \sum_{k=1}^{K} \pi_{ik} \left(\mu_{ik} - \bar{\mu}_i\right) \left(\mu_{ik} - \bar{\mu}_i\right)^\top. \end{aligned} \tag{24}$$

This preserves the checkpoint and the first two moments of its prediction while removing separate mixture modes. The covariance retains a diagonal-plus-low-rank form, with rank at most Kr from the component covariances and K additional rank-one factors from the component means. Sampling therefore has the same order of cost as the mixture.

Single Gaussian head. The operator $\mathcal{B}_{\text{gauss}}$ draws K independent samples from the $\text{Latent}_{\text{CoLaR}}$-RL checkpoint in Table 1, using that model's own branching state $h_{i^\star}$. It provides a comparison with a transition distribution trained as unimodal. All four operators reuse existing checkpoints.

Evaluation set. We use GPQA with 546 items, VoiceBench OpenBookQA with 455, and VoiceBench BBH with 1,000. These tasks have deterministic answer checks and solutions that do not depend on paralinguistic cues. The analysis therefore tests alternative reasoning paths; whether components also separate competing acoustic interpretations is left to future work. The pooled set contains 2,001 prompts. Four operators with four branches each give 16 rollouts per prompt and 32,016 in total.

E.2 MEASURING DISTINCT AND VALID PATHS

We assess both how the continuations differ and whether they reach correct answers. For branch k, let $h_j^{(k)}$ denote a state after the intervention, $L^{(k)}$ its total latent depth, and $\widehat{a}^{(k)}$ its final answer. Pairwise metrics average over the $K(K-1)/2$ unordered branch pairs.

Latent and language differences. Latent path divergence measures the mean cosine distance across the m slots of the intervened chunk

$$D = \frac{2}{K(K-1)} \sum_{k<k'} \frac{1}{m} \sum_{s=1}^{m} \left(1 - \cos\left(\widehat{x}_s^{(k)}, \widehat{x}_s^{(k')}\right)\right). \tag{25}$$

CSA also makes latent states readable by the language head, giving a complementary view of their content. We project each state through the frozen language head, $q_j^{(k)} = p_\theta(\cdot \mid h_j^{(k)})$, and compare the resulting vocabulary distributions at aligned positions using Jensen-Shannon divergence

$$J = \frac{2}{K(K-1)} \sum_{k<k'} \frac{1}{n_{kk'}} \sum_{j=1}^{n_{kk'}} \mathrm{JS}\left(q_j^{(k)} \,\|\, q_j^{(k')}\right), \qquad n_{kk'} = \min\left(L^{(k)}, L^{(k')}\right) - i^\star. \tag{26}$$

We additionally report O, the mean pairwise Jaccard overlap of the top five tokens pooled over states after branching, and ΔL, the mean pairwise absolute depth difference. Together, these quantities describe differences in latent geometry, language head projections, and reasoning length.

Distinct and correct rate. A useful alternative path must both differ from another continuation and reach a correct answer. Let $J_{kk'}$ be the projection distance for a branch pair, and let θ be its median under $\mathcal{B}_{\text{noise}}$. We use this typical within-component variation as the threshold for distinctness. The distinct and correct rate is

$$\mathrm{DCR} = \Pr\left[\exists\, k < k' \quad (J_{kk'} > \theta) \wedge \left(\widehat{a}^{(k)}, \widehat{a}^{(k')} \text{ both correct}\right)\right]. \tag{27}$$

The same threshold applies to every operator, making DCR the fraction of prompts with at least two correct paths separated beyond typical resampling within one component.

Answer outcomes. Answer change rate counts prompts whose branches produce more than one answer. Branch accuracy averages accuracy over all K continuations. Vote accuracy scores their majority answer, with mode vote ties resolved toward the branch with the largest $\pi_{i^\star k}$. These outcomes show whether diversity preserves individual answer quality and supplies complementary solutions when branches are combined.

E.3 Intervention Results

Table 10. Continuation diversity and correctness over 2,001 prompts. D and J measure latent and language projection distance, O is token overlap, and ΔL is depth difference. Answer change, branch accuracy, and DCR are percentages.

Operator	$D \uparrow$	$J \uparrow$	$O \downarrow$	ΔL	Answer change	Branch accuracy	DCR
Mode intervention	0.312	0.283	0.518	3.41	38.68	62.19	**38.88**
Within-component	0.087	0.064	0.791	1.28	14.24	62.81	12.84
Collapsed mixture	0.108	0.089	0.742	1.54	18.49	62.48	16.09
Single Gaussian	0.071	0.047	0.816	0.93	11.34	61.77	9.45

Components yield distinct correct continuations. Table 10 gives the distance measurements behind the DCR summary in Table 3. Mode intervention raises latent distance D from 0.087 to 0.312 and language projection distance J from 0.064 to 0.283 relative to within-component resampling. Token overlap after branching falls from 0.791 to 0.518, while the mean pairwise depth difference rises from 1.28 to 3.41 states. The branches therefore differ in the intervened latent chunk, in the tokens favored by the language head, and in how far they continue to reason. The four mode-intervention branches reuse the same noise draw, so their differences at the branching point come from the component parameters. The within-component control uses independent noise draws and still produces much less separation.

Collapsing the mixture reduces diversity. The collapsed distribution retains the same model parameters and prediction moments, yet its projection distance falls to 0.089 and its answer change rate to 18.49%, compared with 38.68% for mode intervention. Its DCR of 16.09% is also below 38.88%. This comparison connects the diversity to the separate modes of the predicted distribution. The capacity ablation in Table 2 complements this result, as increasing K from 4 to 8 or 16 brings no consistent accuracy gain.

Table 11. Branch and majority vote accuracy by benchmark, in percent. The reference is the default AURAL-RL single rollout from Table 1. Each branching operator generates four continuations.

	GPQA		OpenBookQA		BBH	
Operator	**Branch**	**Vote**	**Branch**	**Vote**	**Branch**	**Vote**
AURAL-RL, single rollout	39.38	n/a	87.25	n/a	66.00	n/a
Mode intervention	38.19	**42.49**	85.82	**89.23**	64.55	**68.30**
Within-component	38.83	40.29	86.37	87.69	65.18	66.50
Collapsed mixture	38.46	40.84	86.04	88.13	64.88	67.00
Single Gaussian	36.72	38.28	84.89	86.37	64.93	65.90

Complementary paths improve voting. Table 11 separates branch accuracy from vote accuracy on each benchmark. Individual branches under mode intervention score 38.19, 85.82, and 64.55 on GPQA, OpenBookQA, and BBH, all within 1.5 points of the default single rollout, so forcing a component does not damage any one continuation. Relative to its own branch accuracy, voting over the four mode-intervention branches adds 4.30, 3.41, and 3.75 points on GPQA, OpenBookQA, and BBH. The same vote adds 2.38, 2.09, and 2.12 points after mixture collapse, 1.46, 1.32, and 1.32 points under within-component resampling, and 1.56, 1.48, and 0.97 points for the single Gaussian head. Mode intervention is also the only operator whose vote exceeds the default single rollout by more than 1.5 points on every benchmark. Complementarity therefore tracks the presence of separate modes rather than the number of samples.

Table 12. Mode intervention by mixture entropy at the branching state. Terciles split all 2,001 prompts into three equal groups by entropy. Answer change and DCR are percentages.

Entropy tercile	$H(\pi_{i^\star})$ **range**	$D \uparrow$	$J \uparrow$	**Answer change**	**DCR**
Bottom	$[0.08, 0.51]$	0.189	0.158	21.44	22.79
Middle	$[0.51, 0.96]$	0.314	0.284	38.53	39.13
Top	$[0.96, 1.38]$	0.434	0.407	56.07	54.72

Mixture uncertainty tracks alternative paths. Table 12 groups prompts by entropy at the intervention state. DCR rises from 22.79% in the bottom tercile to 54.72% in the top tercile, alongside increasing latent and projection distances. When the model spreads probability across several components, intervening on those components reveals more distinct valid continuations. This connects the weights in Equation 2 to the availability of alternative reasoning paths at the current state.

F Data Construction Details

F.1 Corpus Composition

AuralReason-683K combines controlled affective dialogue, human speech, and general reasoning to train both paralinguistic understanding and problem solving. Table 13 gives the final bilingual mixture. Every example contains a waveform, a concise CoT, and a response. The sections below follow each branch from its source material through filtering, then describe the shared supervision format.

Table 13. AuralReason-683K after filtering and CoT construction.

Branch	Chinese	English	Total	Hours
LIME-440K	197K	123K	320K	367
EmotionCoT-35K	0	18K	18K	22
HumanSpeech-1M	47K	28K	75K	83
GeneralSpeech	133K	137K	270K	526
Total	377K	306K	683K	∼1,000

F.2 Sources and Filtering

LIME-440K. LIME-440K pairs lexically identical content with different emotions, providing controlled variation in acoustic affect (Zhao et al., 2026). Expressive synthesis forms its core, while human recordings from ECD-TSE and the Emotion Speech Dataset broaden the speakers and recording conditions. We use its audio to generate new CoTs and responses. Gemini-3.5-Flash listens to each waveform and retains it when audio quality, transcript agreement, conversational delivery, assistant-directed speech, implicit affect, and response feasibility all pass. The retained source counts and acceptance rates appear in Table 14.

EmotionCoT-35K. EmotionCoT-35K supplies human speech for emotion classification (Wang et al., 2026b). Its sources cover acted dialogue, media speech, expressive reading, podcasts, and controlled recordings through IEMOCAP, MELD, Expresso, MSP-Podcast, and EARS (Busso et al., 2008; Poria et al., 2019; Nguyen et al., 2023; Busso et al., 2026; Richter et al., 2024). We retain the audio and source labels and generate new CoTs from the waveforms. IEMOCAP Session 5 and the MELD test split are removed before filtering, so no test utterance, and for IEMOCAP no test speaker, enters supervised or RL training.

Gemini-3.5-Flash predicts each clip's emotion from the waveform and allowed label set, with the source label withheld. Agreement with the source label determines retention. Of 33,242 clips, 17,861 (53.73%) pass through label agreement, and the remaining 15,381 have conflicting predictions or receive structural rejection. Table 14 gives the retained composition.

Table 14. Retained sources in LIME-440K and EmotionCoT-35K. Acceptance is the share of input clips that reach the retained set; the Total row pools inputs across sources rather than averaging rates.

Source	Retained	Acceptance
LIME-440K		
LIME Core Chinese	191,405	89.9%
LIME Core English	89,785	93.5%
ECD-TSE extension	24,991	29.8%
Emotion Speech Dataset extension	15,169	59.4%
Total	321,350	76.8%
EmotionCoT-35K		
MELD	5,810	52.8%
IEMOCAP	3,310	43.7%
Expresso	4,472	59.8%
MSP-Podcast	3,772	56.8%
EARS	497	92.9%
Total	17,861	53.7%

HumanSpeech-1M. To extend dialogue supervision to varied human recordings, we collect roughly 20,000 hours from public videos, podcasts, films, and speech corpora. These span spontaneous, read, acted, media, and replayed speech. GigaSpeech and WenetSpeech provide the largest English and Chinese components (Chen et al., 2021; Zhang et al., 2022). EmotionTalk, VoxMovies,

RealMAN, ESD, and CSEMOTIONS add dialogue, film speech, real acoustic environments, and professionally recorded emotion (Sun et al., 2025; Brown et al., 2021; Yang et al., 2024; Zhou et al., 2022; Tian et al., 2025b). The affective media collection aggregates film, everyday, and acted emotional speech gathered from public video and media sources.

W2V2-AASIST, a wav2vec 2.0 front end with an AASIST spoofing and deepfake detection back end, removes clips it scores as likely synthetic or manipulated (Tak et al., 2022). We segment by speaker and normalize the audio to 16 kHz mono. Each clip must contain one speaker, last 1 to 25 seconds, and have a nonempty transcript with valid language and scene metadata. Emotion screening requires an emotion2vec maximum class probability of at least 0.80 and agreement with Qwen3-Omni (Ma et al., 2024; Xu et al., 2025b). For Qwen3-Omni, confidence over ten votes must be at least 0.90 and entropy at most 0.20. This screening yields 1,691,770 clips totaling 1,999.97 hours, with the source breakdown in Table 15.

Table 15. HumanSpeech-1M after speech screening and segmentation.

Language	**Source**	**Clips**	**Hours**
English	Affective media collection	259,152	337.10
English	GigaSpeech Audiobook	254,871	303.79
English	GigaSpeech YT	130,254	187.28
English	GigaSpeech POD	108,657	156.90
English	VoxMovies	3,752	3.43
Chinese	WenetSpeech L	511,155	542.48
Chinese	Affective media collection	407,950	441.05
Chinese	EmotionTalk	7,410	10.43
Chinese	Hongloumeng	2,950	8.14
Chinese	CSEMOTIONS	2,844	6.19
Chinese	RealMAN	1,715	2.34
Chinese	ESD	1,060	0.84
Total		1,691,770	1,999.97

The next stages select utterances for which hearing the voice can change a useful assistant response. DeepSeek-V4-Flash first makes a high-recall pass over transcripts, and GPT-5.5 then narrows the set for precision, leaving 90K candidates. They favor substantive, self-contained utterances addressed to an assistant and remove narration, news, lectures, quoted dialogue, missing context, explicit affect statements, and requests for physical action. Gemini-3.5-Flash then listens for audible affect and response feasibility. The final branch contains approximately 75K utterances and 83 hours of speech. Table 16 reports the detailed counts.

Table 16. HumanSpeech-1M after dialogue filtering.

Language	**Source**	**Retained**
Chinese	WenetSpeech L	26,854
Chinese	Affective media collection	19,784
English	GigaSpeech Audiobook	9,464
English	GigaSpeech YT	6,809
English	Affective media collection	6,308
English	GigaSpeech POD	4,681
Mixed	Other human speech sources	1,029
Total		74,929

GeneralSpeech. GeneralSpeech adds Chinese and English tasks in mathematics, knowledge, comprehension, and instruction following. GSM8K, MATH, and NuminaMath-1.5 supply mathematical problems, CommonsenseQA and BoolQ supply choice and Boolean questions, and Tülu 3,

public question answering corpora, and internal tasks extend general instruction coverage (Cobbe et al., 2021; Hendrycks et al., 2021; Li et al., 2024; Talmor et al., 2019; Clark et al., 2019; Lambert et al., 2024).

Before Qwen3-TTS synthesis (Hu et al., 2026), written inputs are normalized into natural spoken questions. This step verbalizes equations, symbols, and units, removes markup, and preserves all numbers, conditions, options, logical relations, and requested answer forms. The initial 581K utterances total approximately 1,200 hours. Removing inputs that cannot be rendered as self-contained spoken questions, together with a 25-second duration limit, leaves 428K candidates. Subsequent CoT rewriting checks task validity, answer correctness, spoken response suitability, and answer type, producing approximately 270K training utterances. Table 17 reports task shares before final balancing.

Table 17. GeneralSpeech task distribution before final balancing.

Task family	**Share**
Explanation and comprehension	35.8%
Numeric and mathematical	20.2%
Limited generation	20.0%
Multiple choice	13.6%
Factual entity	8.1%
Boolean judgement	2.3%
Total	100%

F.3 CoT Construction

Each CoT preserves the steps needed to support its response. A step introduces evidence, derives a consequence, rules out an alternative, repairs a contradiction, or verifies a conclusion. We remove restatement, repeated paraphrases, empty checks, and reasoning after the conclusion. This gives concise supervision while retaining the derivation, following work on compact reasoning traces (Hassid et al., 2025; Wu et al., 2025b; Xu et al., 2025c).

Dialogue and emotion classification. Gemini-3.5-Flash listens to each retained waveform and generates a CoT and response using the states in Table 18. Dialogue traces connect audible evidence to a conversational need and response plan. Emotion traces compare acoustic evidence across candidate classes and end with an affective description. Perceived age, gender, accent, and background sound enter the dialogue trace only when they affect the plan, phrased as acoustic impressions rather than identity claims. Final response wording and emotion labels appear in the response field.

Table 18. CoT states for dialogue and emotion classification. The dialogue Risk state is included when ambiguity or safety changes the response plan.

State	**Content**
Dialogue	
Cue	Audible properties that affect the reply, such as a shaky voice and long pauses
Need	The conversational need suggested by those cues, such as reassurance before advice
Risk	Ambiguity or safety considerations that affect the reply
Plan	The response act, order, and tone, such as brief validation followed by one next step
Emotion classification	
Cue	Pitch, energy, pace, rhythm, voice quality, and pauses
Rule out	An acoustic distinction from the closest competing emotion
State	Valence, arousal, and control, with the final label reserved for the response

General reasoning. Qwen3-Omni-Thinking first generates a CoT and response (Xu et al., 2025b). Gemini-3.5-Flash applies a shared shortening rule to produce a comparison version and a verified

training version. The comparison version preserves the original answer exactly, including errors, so each original and shortened trace is paired with the same question and answer. This pairing supports the Qwen2.5-Omni experiment in Section 5.5 (Xu et al., 2025a). Because the shared rule also removes inconsistent steps, a comparison trace can repair local slips, as in Table 25, while its answer stays fixed. The training version additionally verifies the answer and repairs both the CoT and response when needed. It retains 95.68% of rollouts after answer verification, regenerates 4.16% with incorrect answers, and revises the remaining 0.16% for safety. Verified training outputs enter the main mixture.

Validation. Generated examples are checked for valid audio and transcript alignment, required CoT states, answer separation, and consistency between reasoning and response. CoTs must stay below 120 tokens and contain useful intermediate steps. Final answer strings, option letters, emotion labels, and dialogue response wording belong in the response alone. Persona checks reject unsupported identity or capability claims, diagnoses, and promises of physical action. A failed trace is regenerated or rejected, preserving complete reasoning rather than truncating it. Table 19 summarizes the resulting lengths.

Table 19. CoT token lengths and mean state counts after rewriting. A state is one reasoning step; the dialogue and emotion branches use the named states in Table 18. All traces contain fewer than 120 tokens.

Branch	p10	p50	p90	Mean states
LIME-440K	50	71	95	3.67
EmotionCoT-35K	41	46	52	3.00
HumanSpeech-1M	49	65	89	3.31
GeneralSpeech	40	63	88	2.10

F.4 Prompt Templates

The templates below summarize the teacher instructions. Dialogue and emotion generation use the state definitions in Table 18. Each generation template requires a CoT below 120 tokens and a separate final response.

Transcript filtering

High-recall pass Given a transcript with language, duration, speaker, and scene metadata, keep natural, substantive, self-contained speech from one speaker suitable for an assistant response. Retain uncertain cases with a plausible conversational interpretation. Reject narration, news, lectures, recitation, quoted dialogue, missing context, severe transcription errors, system messages, role impersonation, and requests for physical action.
High-precision pass Review the retained utterance for direct address to an assistant and enough context for a useful reply. Keep it when the waveform could change the interpretation, tone, or content of that reply and the transcript leaves the affective condition implicit.
Output Return *Keep* or *Reject* with one concise reason at each pass.

Dialogue and emotion supervision

Dialogue Listen to the waveform and check audio quality, transcript agreement, conversational delivery, assistant-directed speech, implicit affect, and response feasibility. If it passes, write Cue, Need, optional Risk, and Plan states, followed by a concise response that executes the plan. Otherwise return the decisive rejection reason.
Emotion classification Predict emotion from the waveform using the allowed labels, independently of the source label. Write Cue, Rule out, and State, grounding the comparison in audible properties. Place the final emotion label in the response.

Text normalization for speech synthesis

Rewrite the written question as natural spoken language, preserving every number, condition, option, unit, and logical relation. Expand equations, symbols, markup, and abbreviations as needed for unambiguous pronunciation. Return only the normalized question, leaving it unsolved.

GeneralSpeech rewriting

Shared instruction Given the original CoT and response, retain necessary facts, operations, and checks. Remove restatement, redundant operations, empty checks, and reasoning after the conclusion. Keep the CoT below 120 tokens and place the final answer in the response.
Comparison version Preserve the original answer exactly, including any error, and shorten only the CoT.
Verified training version Independently verify the answer. Retain a concise valid derivation when correct. Otherwise solve the task and rewrite both the CoT and response. Make the response natural when spoken.
Output Return the CoT and response for the requested version.

F.5 Plan Execution

A dialogue CoT is useful supervision only if its Plan state shapes the response rather than describing it after the fact. We therefore ask an automatic judge whether each response carries out the plan stated in its CoT, and repeat the judgment on controls that pair the same CoT with a randomly chosen response from another example. Table 20 reports both conditions.

Table 20. Plan execution on 300 dialogue examples. The judge rates whether the response carries out the plan stated in the CoT. Controls pair each CoT with a randomly chosen response from another example.

Pairing	Fully executed	Partly executed	Contradictory
True pair	91.3%	8.0%	0.7%
Mismatched control	6.0%	31.0%	63.0%

True pairs are judged fully executed in 91.3% of cases against 6.0% for the controls, so the judge separates matched from mismatched reasoning rather than accepting any response, and the plans are specific enough that an arbitrary response does not satisfy them.

F.6 Illustrative Examples

Ten selected English examples illustrate how our supervision connects acoustic evidence to a response and condenses general reasoning. All displayed CoTs and responses are complete records, with only typography, line breaks, and mathematical notation normalized. Transcripts aid readability; acoustic descriptions come from the generated CoT annotations.

Table 21: Empathetic dialogue examples. The response plan changes with the voice and conversational need: make room for reluctance, share a success, offer a manageable boundary, or respect a request for space.

Constructed CoT	Response
D1. Good news, a heavy voice	LIME-440K
Transcript: "They promoted me to lead the sustainability project..."	
Cue: low pitch, slow tempo, heavy sighing breath at the start, downward intonation → heavy reluctance, overwhelm **Need:** validation that a promotion can feel like a burden rather than a reward **Risk:** congratulating them forces a positive reaction they do not feel **Plan:** acknowledge the heavy weight of the news, then ask what makes this feel daunting	That sounds like a massive undertaking, and from your tone, it feels more like a heavy weight than a celebration. What is the part of this new role that is weighing on you the most right now?
D2. Letting a success be a success	LIME-440K
Transcript: "The professor approved my thesis draft without any changes!"	
Cue: bright, rapid tempo, rising pitch, audible smile → relief, pride **Need:** validation of massive effort behind clean pass **Risk:** focusing on next steps ruins rare moment of absolute victory **Plan:** match triumphant energy, highlight zero-change approval, invite celebration	Without any changes? That is practically unheard of! You must be walking on air right now after all that work. How are you going to celebrate tonight?
D3. Hearing the need behind a technical question	HumanSpeech-1M
Transcript: "you can just turn off your phone forever is that feasible?"	
Cue: high pitch, rapid tempo, sharp upward inflection on 'forever' and 'feasible' → exasperated overwhelm **Need:** validation of the urge to escape digital noise **Risk:** treating it as a literal technical query ignores the underlying burnout **Plan:** acknowledge the fantasy of disappearing from the grid, then address the practical reality gently	It is completely understandable to want to just pull the plug and disappear from the grid for a while. While you can physically turn it off, the real challenge is dealing with the people who expect you to be reachable. Would you want to try setting up some strict boundaries first, like a do-not-disturb schedule?
D4. Knowing when to stop asking	HumanSpeech-1M
Transcript: "i ain't got nothing to say to nobody."	
Cue: low pitch, heavy vocal tension, sharp downward slide on the final word → defensive anger, high intensity **Need:** to draw a hard boundary and have it respected without pushback **Risk:** asking why they are shut down will feel like an interrogation and provoke more anger **Plan:** acknowledge the boundary immediately, offer a quiet exit	Understood. I will step back and leave you be. If you change your mind later, I'll be right here.

Table 22: Speech emotion recognition (SER) examples from three human speech sources. The task is to identify emotion from the audio. Each CoT connects audible evidence to a competing-class exclusion and an affective state; the selected label appears only in the response.

Constructed CoT	Response
S1. Amusement in a factual utterance	Expresso
Transcript: "There's an alarm set for today at two thirty AM, another for tomorrow at two fifteen PM, and for Wednesday at seven PM."	
Cue: breathy giggling, wide pitch variation, and bright, smiling vocal resonance. **Rule out:** confused — presence of bubbling laughter and rising, playful pitch contours. **State:** high-arousal positive valence with spontaneous amusement.	**happy**
S2. Strained restraint versus overt anger	IEMOCAP
Transcript: "Okay, no, look, I- I- I- I'm really – I'm trying to be patient. I know this isn't your fault. I know that you didn't do it"	
Cue: rapid and uneven tempo, audible sighs, tense vocal quality, frequent self-interruptions **Rule out:** angry — lacks the high-energy acoustic projection and sharp volume spikes **State:** moderate-to-high arousal negative valence, showing signs of strained self-control	**frustrated**
S3. A sudden realization versus sustained joy	MSP-Podcast
Transcript: "and then you look up and you go, 'wait a minute. whoo, things are changing.' "	
Cue: sudden pitch rise, wide dynamic range, breathy exclamation, rapid tempo **Rule out:** happy — lacks sustained positive resonance, dominated by sudden pitch shifts **State:** high arousal, sudden cognitive shift, startled realization	**surprise**

Table 23: General reasoning: the division identity and remainder bound determine the smallest valid divisor, making repeated checks unnecessary. CoT: $529 \rightarrow 62$ tokens ($8.5\times$ shorter).

G1. Minimum feasible dividend GeneralSpeech

Question: When two numbers are divided, the quotient is 12 and the remainder is 3. What is the smallest possible dividend?

Original CoT: Qwen3-Omni-Thinking

Okay, let's see. The problem says when two numbers are divided, the quotient is 12 and the remainder is 3. We need to find the smallest possible dividend. Hmm, right. So first, I remember that when you divide numbers, there's a relationship between the dividend, divisor, quotient, and remainder. Let me recall the formula: Dividend = Divisor $\times$ Quotient + Remainder. Yeah, that's right. So if we let the divisor be d, the quotient is 12, remainder is 3, then the dividend N is $N = 12d + 3$.

But wait, the remainder has to be less than the divisor, right? Because if the remainder was equal to or larger than the divisor, you could divide again. So the remainder r must satisfy $0 \leq r < d$. In this case, the remainder is 3, so $3 < d$. That means the divisor d has to be greater than 3. So the smallest possible divisor would be 4, since it has to be an integer greater than 3. Let me check that. If $d = 4$, then the dividend would be $12 \times 4 + 3 = 48 + 3 = 51$. Let's verify: 51 divided by 4 is 12 with a remainder of 3, because 4 times 12 is 48, and 51 minus 48 is 3. Yep, that works.

Is there a smaller divisor? Well, if $d = 3$, then the remainder would have to be less than 3, but the remainder here is 3, which is not less than 3, so that's not allowed. If $d = 2$, same problem, remainder 3 is bigger than 2, so you could divide more. So the smallest possible divisor is 4, which gives the smallest dividend of 51. Let me just make sure. Let's try $d = 4$: $51 \div 4 = 12.75$, but in integer division, it's 12 with remainder 3. Correct. If we take $d = 5$, then dividend is $12 \times 5 + 3 = 63$, which is bigger, so 51 is smaller. So yeah, 51 is the smallest possible dividend.

Our rewritten CoT

Given: Dividend = Divisor $\times 12 + 3$, and remainder 3 must be less than divisor.

Step: The smallest integer divisor satisfying $3 < d$ is 4.

Step: Dividend $= 12 \times 4 + 3 = 48 + 3$.

Response

The smallest possible dividend is 51, because the divisor must be at least 4, and 12 times 4 plus 3 equals 51.

Table 24: General reasoning: establish consecutive angles, derive their common scale, and select the requested angle. CoT: 517 → 86 tokens (6.0× shorter).

G2. Selecting the obtuse angle GeneralSpeech

Question: The ratio of two interior angles of a parallelogram is $4 : 5$. What is the degree measure of one of the two obtuse angles of the parallelogram?

Original CoT: Qwen3-Omni-Thinking

Okay, let's try to figure out this problem. So, we have a parallelogram, and the ratio of two interior angles is $4 : 5$. We need to find the measure of one of the obtuse angles. Hmm, first, I remember that in a parallelogram, opposite angles are equal, and consecutive angles are supplementary. That means if one angle is, say, x, the next one is $180 - x$, right? Because they add up to 180 degrees.

So, the problem says the ratio of two interior angles is $4 : 5$. Let's assume these are consecutive angles because if they were opposite angles, they would be equal, so their ratio would be $1 : 1$, which isn't $4 : 5$. So, they must be consecutive angles. Let me confirm that. In a parallelogram, adjacent angles are supplementary, so they add up to 180 degrees. So, if the ratio is $4 : 5$, let's let the measures be $4k$ and $5k$. Then, $4k + 5k = 180$ degrees. Because they are consecutive angles, they should add up to 180. Let me check that.

Yes, in a parallelogram, consecutive angles are supplementary. So, $4k + 5k = 180$. That would be $9k = 180$, so $k = 20$. Then, the angles would be $4 \times 20 = 80$ degrees and $5 \times 20 = 100$ degrees. Now, which one is obtuse? Obtuse angles are greater than 90 degrees, so 100 degrees is obtuse. So, the measure of one of the obtuse angles is 100 degrees. Let me just make sure.

In a parallelogram, opposite angles are equal. So, if two consecutive angles are 80 and 100, then the other two angles are also 80 and 100. So, there are two acute angles (80 degrees) and two obtuse angles (100 degrees). So, the obtuse angles are 100 degrees each. That makes sense. Let me verify the ratio: $80 : 100$ simplifies to $4 : 5$, which matches the problem statement. Yep, that works. So, the answer should be 100 degrees.

Our rewritten CoT

Given: parallelogram interior angles ratio $4 : 5$, opposite angles equal, consecutive supplementary.

Step: The ratio must be for consecutive angles. Let them be $4k$ and $5k$. Then $4k + 5k = 180 \Rightarrow 9k = 180 \Rightarrow k = 20$.

Step: The two angles are $4k$ and $5k$. The larger is obtuse.

Response

The obtuse angle is 100 degrees.

Table 25: General reasoning: explaining an observation from a physical law. The rewrite keeps the diffraction limit and the required separation interval. It also resolves the original CoT's reversed inequalities in its first "Therefore" paragraph; the original later gives the correct interval verbally. CoT length: 385 → 66 tokens (5.8× shorter).

G3. Resolving two stars in blue light GeneralSpeech

Question: Why might some stars appear double in blue light, though they could not be resolved in red light with the same telescope?

Original CoT: Qwen3-Omni-Thinking

We are considering the phenomenon where a star appears double in blue light but not in red light with the same telescope.

The key point is the diffraction limit of the telescope. The angular resolution (θ) of a telescope is given by the Rayleigh criterion:
$\theta = 1.22 \times (\lambda/D)$
where:
λ = wavelength of light
D = diameter of the telescope's aperture

Since blue light has a shorter wavelength (λ_{blue}) than red light (λ_{red}), the angular resolution for blue light is better (smaller θ_{blue}) than for red light (θ_{red}).

Therefore, if two stars are separated by an angular distance that is greater than θ_{red} (so they are not resolved in red light) but less than θ_{blue} (so they are resolved in blue light), then we would see them as double in blue light but not in red.

However, note that the Rayleigh criterion is an approximation and the actual resolution also depends on the point spread function and other factors. But the main idea is that the resolving power of a telescope improves as the wavelength decreases.

Example:
Let $\lambda_{\text{red}} \approx 700$ nm, $\lambda_{\text{blue}} \approx 400$ nm.
Then $\theta_{\text{blue}}/\theta_{\text{red}} = \lambda_{\text{blue}}/\lambda_{\text{red}} \approx 400/700 \approx 0.57$.

So the resolving power in blue is about 1.75 times better than in red.

Therefore, if two stars are separated by an angle that is between θ_{blue} and θ_{red}, they will be resolved in blue but not in red.

Hence, the answer is: because blue light has a shorter wavelength, which allows the telescope to resolve finer details (smaller angular separations) than red light.

So the phenomenon occurs due to the wavelength dependence of the diffraction limit.

Our rewritten CoT

Given: telescope angular resolution $\theta \approx 1.22\lambda/D$

Step: blue light has shorter λ than red light $\rightarrow$ smaller θ for blue

Step: if two stars have angular separation $\Delta\theta$ such that $\theta_{\text{blue}} < \Delta\theta < \theta_{\text{red}}$, they are unresolved in red but resolved in blue

Response

Stars appear double in blue light because the telescope's diffraction limit depends on wavelength. Blue light, with its shorter wavelength, gives better angular resolution than red light. If the stars' separation falls between the red and blue resolution limits, they blend into one in red but appear as two in blue.