Is Text All You Need? Text as a Universal Information Bottleneck for Speech LLMs

Ming-Hao Hsu[1,†] **Yuxuan Hu**[2] **Shujie Liu**[3,*] **Jinyu Li**[2] **Zhizheng Wu**[1,4,5,*]

Abstract

Large language models (LLMs) provide a powerful reasoning backbone for speech understanding, but integrating continuous acoustic signals into a pretrained LLM remains challenging. A central design question is how to tie speech representations to the LLM's text input space while preserving the continuous variation needed for different speech tasks. In this work, we propose Convex Gate (C-Gate), a speech–LLM bridge that represents each frame as a convex combination of frozen token embeddings. This architectural convex-hull constraint fixes the output codebook while allowing continuous mixtures learned from task supervision. With a Whisper-large-v3 encoder, a partially adapted Qwen2.5-7B-Instruct decoder, and joint training on speech recognition, emotion recognition, and spoken reasoning, C-Gate performs accurate transcription and emotion recognition, our representative semantic and acoustic tasks, through one shared decoder. On text-only benchmarks, the adapted decoder loses instruction-following and MMLU accuracy and matches or exceeds the original model on GSM8K and HellaSwag. We further design analyses and interventions on the trained model to examine where the bridge sequence carries task information and how the decoder reads it. These findings show how fixed text embeddings can support both semantic and acoustic readouts through continuous speech-conditioned mixtures.

1 Introduction

Reusing pretrained large language models as the reasoning backbone for speech understanding is increasingly necessary for scalable multimodal systems. Instruction-tuned LLMs such as Qwen2.5-7B-Instruct [Yang et al., 2024] carry world knowledge, calibrated answer behavior, and tool-use priors whose retraining from speech-aligned data alone would be prohibitively expensive. A standard design pattern attaches a speech encoder to a pretrained LLM via a learned bridge [Li et al., 2023, Rubenstein et al., 2023, Zhang et al., 2023, Tang et al., 2024, Chu et al., 2023, 2024], allowing a single decoder to serve transcription, paralinguistic, and spoken-reasoning workloads. Under this paradigm, the central design question is what representation form should be used to feed speech information into the LLM context. One answer is continuous prompting, which conditions pretrained LLMs through learned prefix representations [Li and Liang, 2021, Lester et al., 2021]. Another, increasingly popular, answer constructs speech representations from the LLM's own input embedding table [Ma et al., 2025, Yang et al., 2025].

Existing interfaces make different choices about vocabulary alignment and continuous expressivity. Some methods align speech representations with discrete tokens using objectives such as CTC or text alignment [Ma et al., 2025, Fan et al., 2025, Peng et al., 2026, Wang et al., 2026a]. These objectives emphasize semantic content, while emotion recognition also requires acoustic cues such as prosody. Other approaches learn continuous tokens or latent embeddings without restricting them to mixtures

[1]School of Data Science, The Chinese University of Hong Kong, Shenzhen, China. [2]Microsoft CoreAI, Redmond, WA, USA. [3]Microsoft Research Asia, Hong Kong, China. [4]City University of Macau. [5]Amphion Technology Co., Ltd. *Correspondence to: Shujie Liu <shujliu@microsoft.com> and Zhizheng Wu <wuzhizheng@cuhk.edu.cn>. †Work done during an internship at Microsoft.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

of vocabulary embeddings [Tang et al., 2024, Chu et al., 2023, 2024]. This gives the bridge flexibility to choose its output representations. Together, these designs motivate separating two choices: the embedding vectors available to the bridge and the supervision used to combine them. We ask whether a fixed text embedding table can support both semantic and acoustic targets when speech selects continuous mixtures through task supervision. We take speech recognition and emotion recognition as the representative semantic and acoustic tasks.

To study this question, we propose the Convex Gate (C-Gate), a speech–LLM bridge with an architectural convex-hull constraint. We represent each time step as a convex combination of LLM input embeddings, fixing the output codebook while preserving continuous expressivity. Concretely, C-Gate scores the full vocabulary using downsampled speech features, selects the top-16 entries, and renormalizes their weights to combine rows of the frozen embedding table $\mathbf{E}$. The bridge uses the raw embedding values, with no value projection or post-codebook multilayer perceptron (MLP). This construction separates a fixed output geometry from the choice of routing supervision. Within this construction, task losses train only the selection and the mixture weights, allowing the support to carry information useful for transcription or emotion recognition. The shared embedding coordinates also make the selected tokens, mixture weights, and decoder attention directly inspectable.

Our experiments show that C-Gate supports both transcription and emotion recognition with a shared decoder. Beyond task performance, we find that selected token identities carry emotion information even when their mixture weights are nearly uniform. Interventions on trained models further show that decoding depends on temporal order and the learned key–value correspondence. These findings distinguish the information recoverable from the selected supports from how the decoder uses the full bridge sequence.

We summarize our contributions as follows:

- **Geometry-Constrained Speech–LLM Interface:** We introduce C-Gate, which represents each speech frame as a sparse convex mixture of 16 rows of the frozen LLM embedding table, selected by scoring the full vocabulary. The bridge applies no projection after the embedding table, so its outputs stay inside the convex hull of the text embeddings, and within the bridge the task losses train only which rows are selected and how they are weighted.
- **Semantic and Acoustic Tasks from One Frozen Codebook:** We show that convex mixtures of a frozen text embedding table support transcription as a semantic task and emotion recognition as an acoustic task with a shared, partially adapted decoder. In our experiments, a single checkpoint trained jointly on speech recognition, emotion recognition, and spoken reasoning reaches 3.98% and 8.42% word error rate (WER) on LibriSpeech test-clean and test-other and 84.3% accuracy on emotion test recordings from eight corpora.
- **Representation and Readout Analysis:** We show that the mixture weights are nearly uniform and that the identities of the selected tokens alone support a linear emotion readout at least as accurate as that of the pooled encoder states. Interventions on the trained model show that decoding depends on frame order and on the learned key–value correspondence, while setting the weights to exactly uniform has a much smaller effect.

2 Related Work

Speech-to-LLM Adapters. Speech–LLM systems require an adapter that converts acoustic encoder features into a sequence of embeddings the language-model decoder can consume. Adapter designs differ in how they aggregate acoustic features over time and construct the representations passed to the decoder [Tang et al., 2024, Ma et al., 2024a, Lee and Chun, 2026]. One line of designs uses query-based aggregation, which has a precedent in vision–language modeling [Alayrac et al., 2022], where BLIP-2 introduced the Q-Former to connect a frozen image encoder and a frozen language model [Li et al., 2023]. SALMONN extends this design to speech and audio through a window-level Q-Former that produces a variable-length sequence of audio embeddings [Li et al., 2023, Tang et al., 2024]. Qwen-Audio and Qwen2-Audio, in turn, connect audio encoders to LLMs through audio–text pretraining across multiple tasks [Chu et al., 2023, 2024]. However, these adapters leave the bridge a free output space, so where its outputs settle relative to the text embeddings is an outcome of training [Hsu et al., 2026a] rather than a property of the design. We instead fix that geometry by construction and ask what a bridge confined to the text embedding hull can still carry.

Vocabulary Mixtures and Alignment Objectives. Beyond the adapter architecture, a speech–LLM interface is shaped by both its output representation and the supervision used to learn the mapping [Wang et al., 2023, Deng et al., 2025, Ma et al., 2025]. The output representation can also be discrete, and codec-based tokenizers such as SpeechTokenizer show that discrete speech units can retain speaker and prosodic information alongside content [Zhang et al., 2024, Mousavi et al., 2024, Hsu et al., 2025]. Alignment can guide temporal aggregation, as in SSR's segmentation of speech features and AlignFormer's use of CTC alignments to define dynamic Q-Former windows [Tan et al., 2025, Fan et al., 2025]. CTC posteriors can also provide a common interface for speech and text [Ma et al., 2025], which TASU exploits by simulating these posteriors from text-only data during alignment training [Peng et al., 2026]. Behavioral supervision offers another route [Cuervo et al., 2025]: BLSP learns from text continuations, while TARS uses reinforcement learning to align speech- and text-conditioned hidden representations and generated outputs [Wang et al., 2023, 2026a]. Yet in each case the tie between speech and text enters through the training signal, and a transcript or a text-conditioned behavior defines what the speech representation should match. By contrast, C-Gate places the tie in the architecture and trains the bridge with the task losses alone, so the targets decide what the mixture keeps.

A direct connection to the LLM vocabulary can also be built into the output representation by constructing mixtures of token embeddings [Ma et al., 2025, Yang et al., 2025]. LegoSLM uses CTC posteriors as mixture weights over LLM embeddings, with an additional blank representation, to reconstruct speech embeddings [Ma et al., 2025]. For automatic speech recognition (ASR), Yang et al. [2025] softly discretize audio using a codebook initialized from the LLM embedding table, studying sparse mixtures with both updated and fixed codebooks. These methods establish vocabulary mixtures as a speech–LLM interface for transcription, where the target is itself a token sequence and a CTC posterior or an ASR-trained discretization supplies the weights [Ma et al., 2025, Yang et al., 2025]. They leave open, however, whether the original frozen table can also carry an acoustic target that no transcript token names, with weights learned from the task losses alone. C-Gate takes up this question: each bridge output is a normalized mixture over selected rows of the original, frozen embedding table, and we learn the mixture weights from task losses without CTC supervision.

Evaluation beyond Transcription. Transcription accuracy alone does not reveal how well a speech–LLM interface supports paralinguistic understanding and spoken reasoning [Ma et al., 2024b, Wang et al., 2024a, 2026b,c, Hsu et al., 2026b]. VoiceBench evaluates responses to spoken instructions [Chen et al., 2026], and SpeechMMLU probes knowledge-based question answering from speech [Tan et al., 2025]. MMAU extends evaluation to understanding and reasoning across speech, environmental sounds, and music [Sakshi et al., 2025]. Interpreting these scores also requires distinguishing correct answers from answers that depend on the audio input [Rouditchenko et al., 2025, Zang et al., 2025, He et al., 2026]. Work on audio contribution examines this distinction directly [He et al., 2026], and studies of multiple-choice evaluation show that scores can vary with the ordering and wording of answer choices [López et al., 2025]. These controls ask whether a benchmark score depends on the audio. Beyond adopting them for the reasoning benchmarks, we extend the same logic to the bridge and intervene on the waveform, the key–value mapping, and the frame order of a trained checkpoint to locate what the decoder reads for transcription and emotion recognition.

3 Convex Gate

C-Gate converts acoustic features into weighted averages of the LLM's input embeddings (Fig. 1). The acoustic scorer determines which entries to combine, while the frozen embedding table determines the values passed to the decoder.

Geometric Constraint. Let $\mathbf{E} \in \mathbb{R}^{V \times d_{\text{llm}}}$ be the frozen embedding table, whose row $\mathbf{E}_v$ is the embedding of token v. The permitted outputs belong to

$$\text{conv}(\mathbf{E}) = \left\{ \sum_{v=1}^{V} a_v \mathbf{E}_v : a_v \geq 0,\ \sum_{v=1}^{V} a_v = 1 \right\}. \tag{1}$$

This construction fixes the output codebook while allowing continuous mixtures between its rows. The resulting sequence can carry speech-dependent variation through both the selected embeddings and their mixture weights.

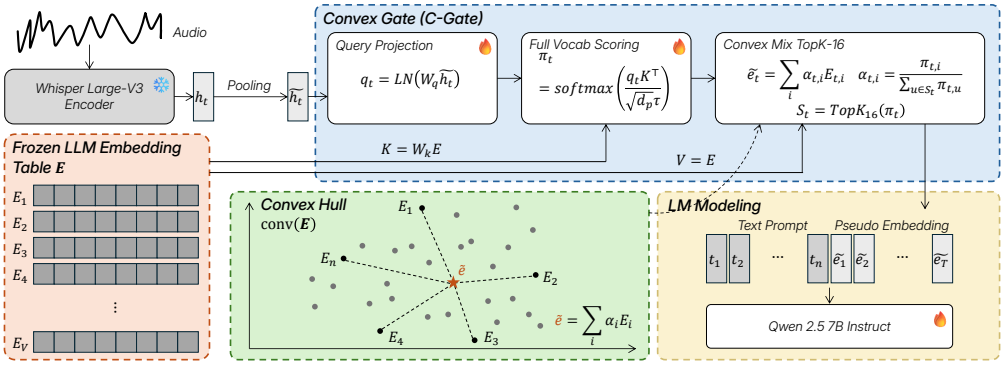

Figure 1: **C-Gate Architecture.** A sparse convex mixture of frozen token embeddings connects the speech encoder to the partially adapted decoder. Snowflakes mark frozen modules, and flames mark components with trainable parameters. Decoder adaptation is restricted to the attention projections specified in Section 4.1.

Acoustic Queries and Vocabulary Keys. The frozen encoder produces $\mathbf{h}_{1:T} \in \mathbb{R}^{T \times d_{\text{enc}}}$. Mean pooling with stride r gives $T' = \lceil T/r \rceil$ pooled states $\tilde{\mathbf{h}}_{1:T'}$. Using row vectors throughout, we form queries and keys as

$$\mathbf{q}_t = \mathrm{LN}(f_q(\tilde{\mathbf{h}}_t)) \in \mathbb{R}^{1 \times d_p}, \qquad \mathbf{K} = f_k(\mathbf{E}) \in \mathbb{R}^{V \times d_p}, \tag{2}$$

where f_q and f_k are learned projection maps with weight matrices $\mathbf{W}_q \in \mathbb{R}^{d_p \times d_{\text{enc}}}$ and $\mathbf{W}_k \in \mathbb{R}^{d_p \times d_{\text{llm}}}$, applied to row vectors by right multiplication with their transposes. Layer normalization acts on the query, and the key projection enters only the scores, so it changes which entries are selected while leaving the values unchanged. With a positive learned temperature τ, we score the vocabulary and select the largest k entries:

$$\mathbf{z}_t = \frac{\mathbf{q}_t \mathbf{K}^\top}{\sqrt{d_p}\,\tau}, \qquad \boldsymbol{\pi}_t = \mathrm{softmax}(\mathbf{z}_t), \qquad S_t = \mathrm{TopK}_k(\boldsymbol{\pi}_t), \tag{3}$$

where $S_t \subset \{1, \ldots, V\}$ indexes the k largest entries of $\boldsymbol{\pi}_t$.

Raw Embedding Mixtures. The selected probabilities are renormalized before constructing each bridge output:

$$\alpha_{t,v} = \begin{cases} \dfrac{\pi_{t,v}}{\sum_{u \in S_t} \pi_{t,u}}, & v \in S_t, \\ 0, & v \notin S_t, \end{cases} \qquad \tilde{\mathbf{e}}_t = \sum_{v \in S_t} \alpha_{t,v} \mathbf{E}_v \in \mathrm{conv}(\mathbf{E}). \tag{4}$$

Nonnegative coefficients summing to one establish the inclusion in Eq. 1 directly. The mixture is passed to the decoder as is, with no learned value projection or post-mixture MLP. For training, support selection is discrete, while the selected weights are differentiable wherever the support remains unchanged. The selection receives no supervision of its own, such as a transcript-token target, and is shaped only by the task losses below.

Decoder Interface. The decoder conditions on the task instruction $\mathbf{x}$ and the bridge sequence $\tilde{\mathbf{e}}_{1:T'}$ and generates the target tokens $y_{1:N}$ autoregressively:

$$p_{\mathcal{M}}(y_{1:N} \mid \mathbf{x}, \tilde{\mathbf{e}}_{1:T'}) = \prod_{i=1}^{N} p_{\mathcal{M}}\big(y_i \mid \mathcal{C}(\mathbf{x}, \tilde{\mathbf{e}}_{1:T'}), y_{<i}\big), \tag{5}$$

where $\mathcal{C}$ denotes the prompt and audio context rendered with the decoder's chat template. The text tokens of the prompt use the same frozen embedding table. At inference, the same model generates transcripts, emotion completions, or reasoning answers. It uses no separate emotion classifier or task-specific decoder.

Training Objective. We train the acoustic scorer and selected decoder attention parameters by minimizing the expected next-token cross-entropy for each task:

$$\mathcal{L}_{\mathrm{CE}} = \mathbb{E}_{\mathcal{D}} \left[-\log p_{\mathcal{M}}\big(y_i \mid \mathcal{C}(\mathbf{x}, \tilde{\mathbf{e}}_{1:T'}), y_{<i}\big) \right]. \tag{6}$$

Here $\mathcal{D}$ denotes the distribution of target-token prediction examples for the task, and $y_{<i}$ is the ground-truth target prefix. When several tasks are trained jointly, we sum their losses with dynamic weights that follow each task's loss relative to its initial value (Appendix A). Throughout training, the speech encoder and the embedding values remain frozen. Section 4 specifies the model configuration and adaptation scope.

4 Experimental Setup

This section describes the model configuration, the training data and optimization, and the evaluation benchmarks. More details can be found in Appendices A and B.

4.1 Model Configuration

We initialize the acoustic encoder with pretrained Whisper-large-v3 [Radford et al., 2023] and the autoregressive decoder with Qwen2.5-7B-Instruct [Yang et al., 2024]. Following the notation in Section 3, the encoder produces acoustic features $\mathbf{h}_t$ with encoder dimension $d_{\mathrm{enc}} = 1{,}280$. The decoder's input embedding table $\mathbf{E}$ has embedding dimension $d_{\mathrm{llm}} = 3{,}584$ and $V = 152{,}064$ rows. The C-Gate scorer maps acoustic features and vocabulary embeddings into queries $\mathbf{q}_t$ and keys $\mathbf{K}$ with a shared scoring dimension $d_p = 512$. We set the support size k to 16 and the mean-pooling stride r to 4.

During adaptation, we train the C-Gate scorer and the query, key, value, and output attention projections in decoder layers 0–23. These are 2.5M scorer parameters and 704.8M attention parameters, which together make up 8.6% of the 8.26B parameters of the full model. All remaining parameters are frozen, including the Whisper-large-v3 encoder, the input embeddings, MLPs, normalization layers, language-model head, and decoder layers 24–27. The embedding values mixed by C-Gate therefore remain fixed throughout training.

4.2 Training Data and Optimization

ASR. We train on the 960 hours of LibriSpeech [Panayotov et al., 2015].

Emotion Recognition. We pool the English portion of CAMEO [Christop and Czajka, 2025], which includes CREMA-D, EMNS, eNTERFACE, JL-Corpus, and RAVDESS, with ESD English [Zhou et al., 2021], IEMOCAP [Busso et al., 2008], SAVEE [Jackson and Haq, 2014], TESS [Dupuis and Pichora-Fuller, 2011], and MELD audio [Poria et al., 2019], and map their labels to eight classes. We split each corpus by recording into a training and a test portion. We train on 53,246 recordings, and the emotion test split has 2,063 recordings from CREMA-D, EMNS, eNTERFACE, JL-Corpus, ESD, IEMOCAP, SAVEE, and TESS.

Reasoning. We use question-answer corpora with synthesized spoken questions, including ARC [Clark et al., 2018], GSM8K [Cobbe et al., 2021], MATH [Hendrycks et al., 2021b], and MMLU-Pro [Wang et al., 2024b]. Appendix A lists all sources and their sizes.

Training Configurations. C-Gate denotes the three-task model trained jointly on ASR, emotion recognition, and reasoning. The other training-task combinations are training-data ablations: a two-task model trained on ASR and emotion, and single-task models trained on ASR, emotion, or reasoning alone. All configurations share the bridge, the adaptation scope, the learning rates, and the batch size. Appendix A lists the training length and learning-rate schedule of each configuration.

Optimization. We optimize with AdamW at learning rates of 10^{-4} for the bridge and 2×10^{-5} for decoder attention, a linear warm-up over 5,000 updates, and an effective batch size of 256. Appendix A provides the remaining settings.

Table 1: **Overall Task Performance.** WER and accuracy are reported in percent. The ESD, IEMOCAP, and CREMA-D columns use the test recordings of our train/test split of each corpus.

Training data			ASR		Emotion			Reasoning		
ASR	Emotion	Reasoning	Clean	Other	ESD	IEMOCAP	CREMA-D	BBH	SpMMLU	MMAU
✓	×	×	3.81	7.39	–	–	–	–	–	–
×	✓	×	–	–	95.4	63.4	75.0	–	–	–
×	×	✓	–	–	–	–	–	28.6	53.2	44.0
✓	✓	×	4.36	8.34	95.7	63.8	75.5	–	–	–
✓	✓	✓	3.98	8.42	94.7	64.6	74.2	43.5	61.4	48.3

4.3 Evaluation Benchmarks and Metrics

Decoding is greedy throughout, and Appendix B gives the prompts, output limits, scoring rules, and the construction of spoken BBH.

ASR. We report WER on the LibriSpeech test-clean and test-other splits, which contain 2,620 and 2,939 utterances.

Emotion Recognition. We report the accuracy of the generated completion on our test recordings of ESD, IEMOCAP, and CREMA-D, the three corpora with the most test recordings, and give the other corpora in Appendix C. The analyses in Section 6 use the whole emotion test split.

Reasoning. We report answer accuracy on spoken BBH [Suzgun et al., 2023, Chen et al., 2026], SpeechMMLU (SpMMLU) [Tan et al., 2025], and MMAU [Sakshi et al., 2025]. In these three benchmarks the text prompt contains the written question, and the audio carries the spoken question or the audio clip that the question refers to.

Text-Only Capabilities. To assess text-only capabilities after speech adaptation, we compare C-Gate with the base Qwen2.5-7B-Instruct model. Examples, prompts, shot counts, decoding settings, parsers, and seeds are matched within each benchmark. We report prompt-level strict IFEval accuracy [Zhou et al., 2023] and 0-shot normalized HellaSwag accuracy [Zellers et al., 2019].

5 Main Results

We report the experimental results in terms of word error rate (LibriSpeech test-clean and test-other) and accuracy (emotion recognition and reasoning benchmarks) in Table 1. Appendix C provides additional results, including the LibriSpeech WERs of published speech LLMs and emotion accuracy for every corpus and by class.

5.1 Overall Task Performance

Semantic and Acoustic Tasks. We evaluate ASR and emotion recognition as the representative semantic and acoustic tasks. C-Gate, trained jointly on ASR, emotion recognition, and reasoning (last row of Table 1), attains 3.98% and 8.42% WER on LibriSpeech test-clean and test-other and emotion accuracy of 94.7% on ESD, 64.6% on IEMOCAP, and 74.2% on CREMA-D. In both cases, one decoder generates the transcripts and the emotion labels from mixtures of 16 frozen token embeddings per frame. This shows that continuous mixtures of fixed text embeddings support a semantic and an acoustic task under partial decoder adaptation.

Effect of Joint Training. We compare C-Gate with models trained on subsets of the three tasks. For ASR, the WER of the joint models stays within 0.55 points of the ASR-only model on test-clean and 1.03 points on test-other. For emotion recognition, the three models trained on emotion data agree within 1.3 points on ESD, IEMOCAP, and CREMA-D, and their accuracy over the whole emotion test split is 84.0%, 84.1%, and 84.3%. The jointly trained models therefore score close to the single-task models on both tasks, and Section 6 examines how the bridge sequence carries both.

Table 2: **Text-Only Performance after Adaptation.** Scores are percentages under matched evaluation settings, and differences are percentage points computed before rounding.

Benchmark	Items	Base	Adapted	Difference
IFEval	541	71.35	59.33	-12.01
HellaSwag	10,042	64.30	66.56	$+2.26$

Table 3: **Routing Statistics across Tasks.** Statistics use renormalized weights over the selected top-16 support, averaged over the valid frames of the emotion test split, the first 500 test-clean utterances, or 1,000 VoiceBench BBH questions. $H(\boldsymbol{\alpha}_t)$ is entropy in nats per frame, $H/\log 16$ normalizes it by the maximum entropy, and KL is divergence from the uniform distribution U_{16}. High normalized entropy and low KL indicate nearly uniform weights within the selected support.

Training data			Input task	Routing statistics		
ASR	Emotion	Reasoning		$H(\boldsymbol{\alpha}_t)$	$H/\log 16$	KL
×	✓	×	Emotion	2.744	0.990	0.0282
✓	×	×	ASR	2.660	0.960	0.1122
✓	✓	×	Emotion	2.709	0.977	0.0633
			ASR	2.730	0.985	0.0428
✓	✓	✓	Emotion	2.688	0.970	0.0844
			ASR	2.673	0.964	0.0997
			Reasoning	2.677	0.965	0.0959

Reasoning Benchmarks. C-Gate scores above the reasoning-only model on all three reasoning benchmarks, with 43.5% against 28.6% on BBH. These benchmarks place the written question in the prompt, so the reasoning columns reflect the decoder's answering behavior after joint training (Appendix C).

5.2 Text-Only Performance after Speech Adaptation

To examine how speech adaptation changes the decoder's text-only capabilities, we compare C-Gate with the base Qwen2.5-7B-Instruct model under matched evaluation settings (Table 2). IFEval [Zhou et al., 2023] prompt-level strict accuracy decreases by 12.01 percentage points after speech adaptation, while HellaSwag [Zellers et al., 2019] normalized accuracy improves by 2.26 points. The input embedding table stays fixed, so both changes come from the adapted attention projections. Appendix C extends this comparison to MMLU and GSM8K.

6 Representation and Readout Analysis

We ask where the bridge sequence carries task information and how the decoder reads it. We examine the mixture weights, the selected tokens, and the decoder's attention to the audio frames, and then intervene on the bridge at fixed model weights. All interventions use the emotion test split and the first 500 test-clean utterances, on which the unmodified C-Gate attains 84.29% emotion accuracy and 4.01% WER.

6.1 Mixture Concentration

We first examine how much task information the mixture weights carry. The renormalized mixture weights are close to uniform in every training configuration and for every input task (Table 3). Normalized entropy ranges from 0.960 to 0.990, and the bridge vectors of C-Gate have a mean cosine similarity of 0.94 to 0.95 with the unweighted mean of their 16 selected embeddings. To measure what this residual variation contributes, we set every weight to exactly $1/16$ on the same selected tokens, which lowers emotion accuracy to 81.63% and raises WER to 5.85% (Table 17 in Appendix E). Appendix D shows that the softmax before truncation is nearly flat as well and that near-uniform weights still pass gradient to the scorer. The weights therefore carry a small part of the

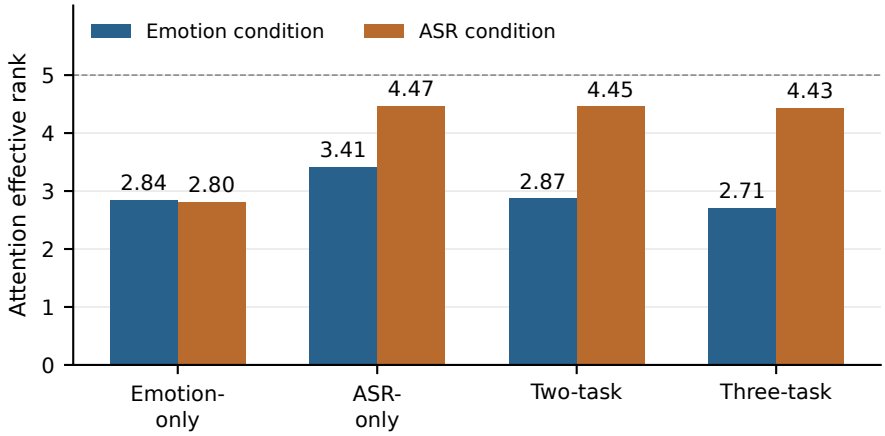

Figure 2: **Decoder Attention to the Audio Frames under Emotion and ASR Inputs.** The effective rank at decoder layer 14 is higher when the first five answer tokens attend to different frames, and it is at most five (dashed line).

task information, and the model keeps most of its accuracy without any variation in them. This leaves two main carriers that Sections 6.2 and 6.4 examine in turn: the identity of the selected tokens and their order over time.

6.2 Emotion Probes across Representations

We next test whether the identities of the selected tokens alone carry emotion information. To this end, we train logistic-regression probes on the emotion test split, with speaker-disjoint folds repeated over ten random partitions (Table 15 in Appendix D). A bag-of-tokens probe sees only how often each of the 1,000 most frequently selected tokens occurs in a recording's supports, without weights or frame order. It reaches 63.3% accuracy, against 57.6% for mean-pooled Whisper states and 66.8% for mean-pooled bridge vectors, and exceeds the Whisper probe in every partition. For reference, the most frequent class accounts for 23.7% of the recordings. This readout is not lexical: a label word appears in only 0.08% of the supports, and the most frequent tokens are unrelated word pieces that occur in nearly every recording. This shows that the selected token identities alone support a linear emotion readout at least as accurate as that of the pooled encoder states.

6.3 Task-Conditional Attention

We then examine how the decoder attends to the audio frames under each task. For each example, we feed the evaluation prompt, the bridge frames, and the reference answer, and take the attention from the first five answer tokens to the audio frames at decoder layer 14, averaged over heads. We summarize this matrix by its entropy-based effective rank [Roy and Vetterli, 2007], $\exp(-\sum_j p_j \log p_j)$ with $p_j = \sigma_j / \sum_i \sigma_i$ for singular values σ_j, which is at most five. A low rank means that the answer tokens share one attention pattern over the audio, and a rank near five means that each token attends to different frames. When transcribing, the three models trained on ASR reach ranks of 4.43 to 4.47, while the emotion-only model stays at 2.80 on the same inputs (Fig. 2). For emotion completion, the two joint models have ranks of 2.87 and 2.71. We report layer 14 because layers 14 and 15 are where the transcription rank peaks, and Appendix D summarizes the other layers and the full-transcript ranks. The joint models thus show position-specific attention when transcribing and more shared attention over the emotion answer.

6.4 Bridge Mapping and Temporal Sensitivity

We test whether the decoder's outputs depend on the acoustic input, on the learned correspondence between keys and values, and on frame order, by intervening on the trained C-Gate checkpoint without retraining.

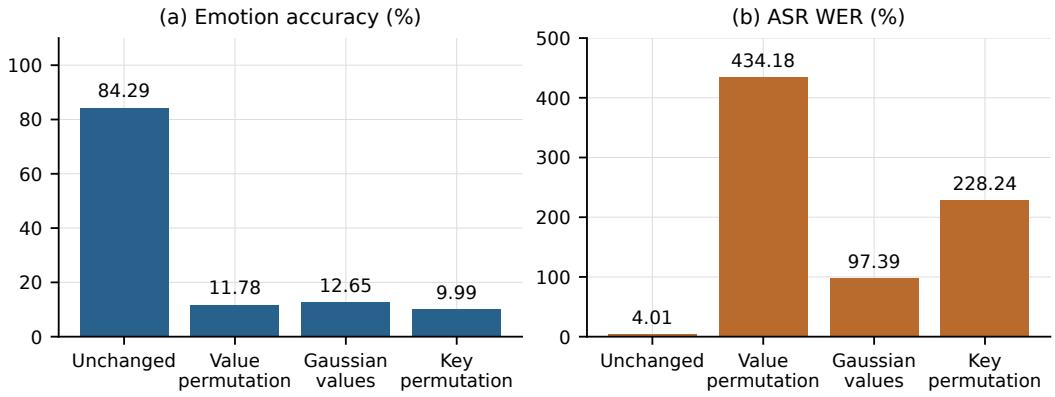

Figure 3: **Performance of C-Gate after Perturbing Its Keys or Values.** Each perturbation is applied to one trained checkpoint without retraining.

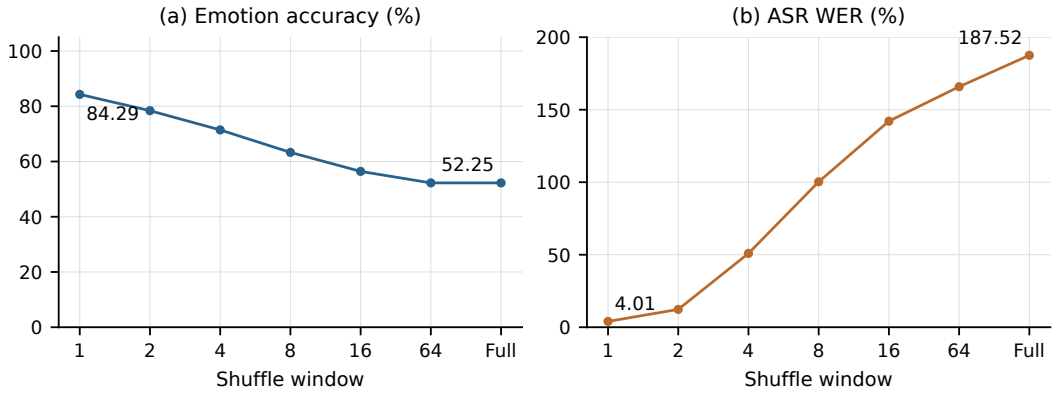

Figure 4: **Performance of C-Gate after Shuffling the Bridge Frames in Time.** Frames are permuted within non-overlapping windows, where a window of 1 preserves the order and Full shuffles the entire sequence. Scores are means over three permutation seeds.

Acoustic Input. Replacing the waveform with zeros or with noise lowers emotion accuracy to 17.2% and 20.0%, below the 23.7% share of the most frequent class, and raises WER to 97.7% and 98.8% (Appendix E). Both tasks therefore depend on the acoustic input.

Key–Value Mapping. We perturb keys or values inside C-Gate while prompt embeddings and the language-model head stay unchanged. Permuting the values, replacing them with Gaussian vectors, or permuting the keys lowers emotion accuracy to 10–13% and raises WER above 97% (Fig. 3). These conditions differ in what they preserve: a row permutation keeps the set of embedding vectors and hence its convex hull, $\mathrm{conv}(\mathbf{PE}) = \mathrm{conv}(\mathbf{E})$ for a permutation matrix $\mathbf{P}$, while Gaussian substitution additionally changes the embedding geometry. The permutation controls therefore remove only the learned correspondence between a score and the vector it retrieves, and the outputs depend on that correspondence beyond membership in the hull.

Temporal Order. We permute the bridge frames within non-overlapping windows of increasing size. Averaged over three permutation seeds, a full shuffle lowers emotion accuracy to 52.25% and raises WER to 187.52% (Fig. 4). Transcription fails once local order is lost, with 51% WER at a window of four frames, while emotion accuracy degrades gradually and stays far above the most frequent class under a full shuffle, in line with the order-free probe of Section 6.2.

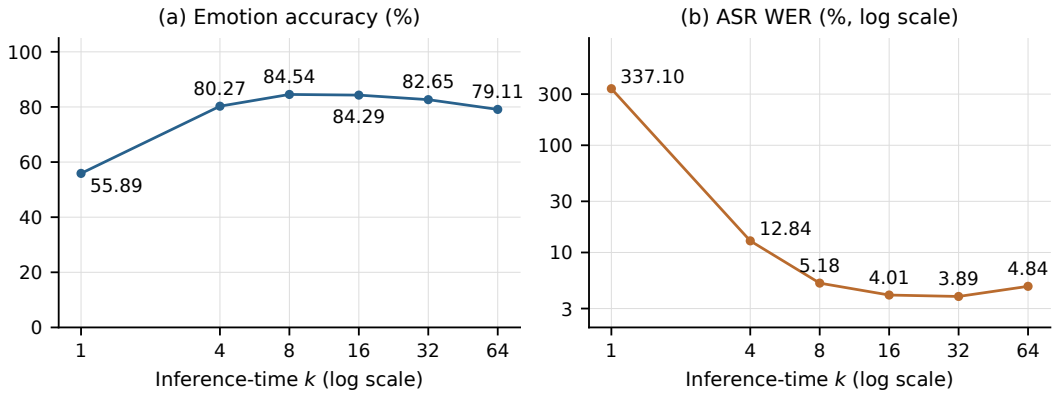

Figure 5: **Changing the Support Size k at Inference.** Emotion accuracy on the emotion test split and WER on the first 500 test-clean utterances for one C-Gate checkpoint trained with $k = 16$. WER exceeds 100% at $k = 1$ because it counts insertions.

6.5 Support Size and Pooling Stride

Finally, we vary the support size k and the pooling stride at inference in the C-Gate checkpoint trained with $k = 16$ and stride 4, without retraining (Fig. 5). Small supports break both tasks: $k = 1$ raises WER to 337.10% and lowers emotion accuracy to 55.89%. Around the trained setting, by contrast, the changes are small. At $k = 8$ emotion accuracy is unchanged at 84.54% and WER rises to 5.18%, and at $k = 32$ WER is unchanged at 3.89% and emotion accuracy falls to 82.65%, where unchanged means that the paired 95% interval includes zero (Appendix E). At $k = 64$ emotion accuracy falls to 79.11% and WER rises to 4.84%. The stride behaves similarly: changing it from 4 to 2 or 8 degrades both scores, to 14.07% and 36.50% WER and to 81.63% and 70.29% emotion accuracy (Appendix E). For this checkpoint, both tasks therefore depend on mixing several embeddings per frame and on the trained stride.

7 Discussion and Conclusion

We propose C-Gate, a speech–LLM bridge that represents each speech frame as a sparse convex mixture of frozen text embeddings. The mixture construction fixes the available embedding values, while task supervision learns which values to select and combine. With a partially adapted decoder, C-Gate supports both transcription and emotion recognition from one frozen embedding table. Our analysis shows that the mixture weights stay near uniform, and replacing them with exactly uniform weights costs 2.7 points of emotion accuracy and 1.8 points of WER, so the speech-dependent variation lies mainly in which tokens are selected and in their order. In particular, token identities alone support a linear emotion readout at least as accurate as that of the pooled encoder states, and the decoder's outputs degrade when frame order or the key–value correspondence is disturbed.

Limitations and Future Work. Our study uses one encoder–LLM pair and one training run per configuration. Matched ablations of the convex-hull constraint and of attention adaptation are left to future work, and Appendix G summarizes the evidence behind each design choice. On the evaluation side, speech adaptation lowers IFEval and MMLU accuracy, and the reasoning scores depend little on the audio in the tested diagnostic. Finally, scoring the full vocabulary takes about 2.9×10^{10} multiply–accumulate operations per 30-second input (Appendix F), and shortlist scoring is left to future work.

Implications and Broader Impact. Fixed embedding mixtures provide an inspectable interface for studying how language models use acoustic information. Joint speech understanding could support accessible voice interfaces, while emotion inference from acted speech can be unreliable in natural settings and can enable intrusive speaker profiling. More broadly, speech quality, text retention, and audio contribution are best evaluated together.

Acknowledgments and Disclosure of Funding

The work of Zhizheng Wu is supported by the National Natural Science Foundation of China under Grant 62376237. This work is supported by the Program for Guangdong Introducing Innovative and Entrepreneurial Teams, Grant No. 2023ZT10X044. The authors have no financial relationships relevant to this work with any entity other than their listed affiliations.

References

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob L. Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karén Simonyan. Flamingo: a Visual Language Model for Few-Shot Learning. In NeurIPS, 2022.

Carlos Busso, Murtaza Bulut, Chi-Chun Lee, Abe Kazemzadeh, Emily Mower, Samuel Kim, Jeannette N. 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.

Houwei Cao, David G. Cooper, Michael K. Keutmann, Ruben C. Gur, Ani Nenkova, and Ragini Verma. CREMA-D: Crowd-Sourced Emotional Multimodal Actors Dataset. IEEE Trans. Affect. Comput., 5(4):377–390, 2014. doi: 10.1109/TAFFC.2014.2336244.

Yiming Chen, Xianghu Yue, Chen Zhang, Xiaoxue Gao, Robby T. Tan, and Haizhou Li. VoiceBench: Benchmarking LLM-Based Voice Assistants. Trans. Assoc. Comput. Linguistics, 14:378–398, 2026.

Iwona Christop and Maciej Czajka. CAMEO: Collection of Multilingual Emotional Speech Corpora, 2025. URL https://arxiv.org/abs/2505.11051.

Yunfei Chu, Jin Xu, Xiaohuan Zhou, Qian Yang, Shiliang Zhang, Zhijie Yan, Chang Zhou, and Jingren Zhou. Qwen-Audio: Advancing Universal Audio Understanding via Unified Large-Scale Audio-Language Models. CoRR, abs/2311.07919, 2023.

Yunfei Chu, Jin Xu, Qian Yang, Haojie Wei, Xipin Wei, Zhifang Guo, Yichong Leng, Yuanjun Lv, Jinzheng He, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen2-Audio Technical Report. CoRR, abs/2407.10759, 2024.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think You Have Solved Question Answering? Try ARC, the AI2 Reasoning Challenge, 2018. URL https://arxiv.org/abs/1803.05457.

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, 2021. URL https://arxiv.org/abs/2110.14168.

Santiago Cuervo, Skyler Seto, Maureen de Seyssel, Richard He Bai, Zijin Gu, Tatiana Likhomanenko, Navdeep Jaitly, and Zakaria Aldeneh. Closing the Gap Between Text and Speech Understanding in LLMs. CoRR, abs/2510.13632, 2025.

Keqi Deng, Guangzhi Sun, and Phil Woodland. Wav2Prompt: End-to-End Speech Prompt Learning and Task-based Fine-tuning for Text-based LLMs. In Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 6940–6956. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.naacl-long.354.

Kate Dupuis and M. Kathleen Pichora-Fuller. Recognition of Emotional Speech for Younger and Older Talkers: Behavioural Findings from the Toronto Emotional Speech Set. Canadian Acoustics, 39(3):182–183, 2011. URL https://jcaa.caa-aca.ca/index.php/jcaa/article/view/2471.

EleutherAI. Language Model Evaluation Harness, version 0.4.5, 2024. URL https://github.com/EleutherAI/lm-evaluation-harness/releases/tag/v0.4.5.

Ruchao Fan, Bo Ren, Yuxuan Hu, Rui Zhao, Shujie Liu, and Jinyu Li. AlignFormer: Modality Matching Can Achieve Better Zero-Shot Instruction-Following Speech-LLM. IEEE J. Sel. Top. Signal Process., 19(7):1329–1337, 2025.

Arushi Goel, Sreyan Ghosh, Jaehyeon Kim, Sonal Kumar, Zhifeng Kong, Sang-gil Lee, Chao-Han Huck Yang, Ramani Duraiswami, Dinesh Manocha, Rafael Valle, and Bryan Catanzaro. Audio Flamingo 3: Advancing Audio Intelligence with Fully Open Large Audio Language Models. CoRR, abs/2507.08128, 2025.

Haolin He, Xingjian Du, Renhe Sun, Zheqi Dai, Yujia Xiao, Mingru Yang, Jiayi Zhou, Xiquan Li, Zhengxi Liu, Zining Liang, Chunyat Wu, Qianhua He, Tan Lee, Xie Chen, Wei-Long Zheng, Weiqiang Wang, Mark Plumbley, Jian Liu, and Qiuqiang Kong. Measuring Audio's Impact on Correctness: Audio-Contribution-Aware Post-Training of Large Audio Language Models, 2026. URL https://arxiv.org/abs/2509.21060.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring Massive Multitask Language Understanding. In International Conference on Learning Representations, 2021a. URL https://arxiv.org/abs/2009.03300.

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 Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks, volume 1, 2021b. URL https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/be83ab3ecd0db773eb2dc1b0a17836a1-Abstract-round2.html.

Ming-Hao Hsu, Xiaohai Tian, Jun Zhang, Lu Lu, Yuxuan Wang, and Zhizheng Wu. Geometric Alignment without Functional Equivalence: A Layer-wise Analysis of the Speech–Text Modality Gap. In NeurIPS, 2026. URL https://openreview.net/forum?id=KNuSWirJEG.

Ming-Hao Hsu, Xiaohai Tian, Jun Zhang, and Zhizheng Wu. Entity Binding Failures in Speech LLM Reasoning: Diagnosis and Chain-of-Thought Intervention. In Interspeech, 2026.

Ming-Hao Hsu, Liang-Hsuan Tseng, Hung-yi Lee, and Zhizheng Wu. TASLA: Text-Aligned Speech Tokens with Multiple Layer-Aggregation. CoRR, abs/2510.14934, 2025.

Shujie Hu, Long Zhou, Shujie Liu, Sanyuan Chen, Lingwei Meng, Hongkun Hao, Jing Pan, Xunying Liu, Jinyu Li, Sunit Sivasankaran, Linquan Liu, and Furu Wei. WavLLM: Towards Robust and Adaptive Speech Large Language Model. In EMNLP (Findings), Findings of ACL, pages 4552–4572. Association for Computational Linguistics, 2024.

Philip Jackson and Sanaul Haq. Surrey Audio-Visual Expressed Emotion (SAVEE) Database. University of Surrey, 2014. URL https://personalpages.surrey.ac.uk/p.jackson/SAVEE/.

Jesin James, Li Tian, and Catherine Inez Watson. An Open Source Emotional Speech Corpus for Human Robot Interaction Applications. In Interspeech, pages 2768–2772. ISCA, 2018. doi: 10.21437/Interspeech.2018-1349.

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

Junseok Lee and Chang-Jae Chun. FastSLM: Hierarchical Temporal Abstraction for Efficient Long-Form Speech Adaptation. CoRR, abs/2601.06199, 2026. URL https://arxiv.org/abs/2601.06199v5.

Brian Lester, Rami Al-Rfou, and Noah Constant. The Power of Scale for Parameter-Efficient Prompt Tuning. In EMNLP (1), pages 3045–3059. Association for Computational Linguistics, 2021.

Jia Li, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Costa Huang, Kashif Rasul, Longhui Yu, Albert Jiang, Ziju Shen, Zihan Qin, Bin Dong, Li Zhou, Yann Fleureau, Guillaume Lample, and Stanislas Polu. NuminaMath. Hugging Face repository, 2024. URL https://huggingface.co/AI-MO/NuminaMath-CoT.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven C. H. Hoi. BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models. In ICML, Proceedings of Machine Learning Research, pages 19730–19742. PMLR, 2023.

Xiang Lisa Li and Percy Liang. Prefix-Tuning: Optimizing Continuous Prompts for Generation. In ACL/IJCNLP (1), pages 4582–4597. Association for Computational Linguistics, 2021.

Shikun Liu, Edward Johns, and Andrew J. Davison. End-To-End Multi-Task Learning With Attention. In CVPR, pages 1871–1880. Computer Vision Foundation / IEEE, 2019.

Steven R. Livingstone and Frank A. Russo. The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in North American English. PLOS ONE, 13(5):e0196391, 2018. doi: 10.1371/journal.pone.0196391.

Fernando López, Santosh Kesiraju, and Jordi Luque. Robustness assessment of large audio language models in multiple-choice evaluation, 2025. URL https://arxiv.org/abs/2510.04584.

Rao Ma, Tongzhou Chen, Kartik Audhkhasi, and Bhuvana Ramabhadran. LegoSLM: Connecting LLM with Speech Encoder using CTC Posteriors. In Findings of the Association for Computational Linguistics: EMNLP 2025, pages 18171–18186. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-emnlp.985.

Ziyang Ma, Guanrou Yang, Yifan Yang, Zhifu Gao, Jiaming Wang, Zhihao Du, Fan Yu, Qian Chen, Siqi Zheng, Shiliang Zhang, and Xie Chen. An Embarrassingly Simple Approach for LLM with Strong ASR Capacity. CoRR, abs/2402.08846, 2024a. URL https://arxiv.org/abs/2402.08846.

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 ACL (Findings), Findings of ACL, pages 15747–15760. Association for Computational Linguistics, 2024b.

Olivier Martin, Irene Kotsia, Benoit Macq, and Ioannis Pitas. The eNTERFACE'05 Audio-Visual Emotion Database. In ICDE Workshops, page 8. IEEE, 2006. doi: 10.1109/ICDEW.2006.145.

Arindam Mitra, Hamed Khanpour, Corby Rosset, and Ahmed Awadallah. Orca-Math: Unlocking the Potential of SLMs in Grade School Math. CoRR, abs/2402.14830, 2024.

Pooneh Mousavi, Jarod Duret, Darius Petermann, Artem Ploujnikov, Luca Della Libera, Anastasia Kuznetsova, Cem Subakan, and Mirco Ravanelli. DASB – Discrete Audio and Speech Benchmark. CoRR, abs/2406.14294, 2024.

Kari Ali Noriy, Xiaosong Yang, and Jian Jun Zhang. EMNS /Imz/ Corpus: An Emotive Single-Speaker Dataset for Narrative Storytelling in Games, Television and Graphic Novels. CoRR, abs/2305.13137, 2023.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: An ASR corpus based on public domain audio books. In ICASSP, pages 5206–5210. IEEE, 2015.

Jing Peng, Yi Yang, Xu Li, Yu Xi, Quanwei Tang, Yangui Fang, Junjie Li, and Kai Yu. TASU: Text-Only Alignment for Speech Understanding, 2026. URL https://arxiv.org/abs/2511.03310.

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, pages 527–536. Association for Computational Linguistics, 2019. doi: 10.18653/v1/P19-1050.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust Speech Recognition via Large-Scale Weak Supervision. In ICML, Proceedings of Machine Learning Research, pages 28492–28518. PMLR, 2023.

Andrew Rouditchenko, Saurabhchand Bhati, Edson Araujo, Samuel Thomas, Hilde Kuehne, Rogerio Feris, and James Glass. Omni-R1: Do You Really Need Audio to Fine-Tune Your Audio LLM? CoRR, abs/2505.09439, 2025. URL https://arxiv.org/abs/2505.09439.

Olivier Roy and Martin Vetterli. The effective rank: A measure of effective dimensionality. In EUSIPCO, pages 606–610. IEEE, 2007.

Paul K. Rubenstein, Chulayuth Asawaroengchai, Duc Dung Nguyen, Ankur Bapna, Zalán Borsos, Félix de Chaumont Quitry, Peter Chen, Dalia El Badawy, Wei Han, Eugene Kharitonov, Hannah Muckenhirn, Dirk Padfield, James Qin, Danny Rozenberg, Tara N. Sainath, Johan Schalkwyk, Matthew Sharifi, Michelle Tadmor Ramanovich, Marco Tagliasacchi, Alexandru Tudor, Mihajlo Velimirovic, Damien Vincent, Jiahui Yu, Yongqiang Wang, Vicky Zayats, Neil Zeghidour, Yu Zhang, Zhishuai Zhang, Lukas Zilka, and Christian Havnø Frank. AudioPaLM: A Large Language Model That Can Speak and Listen. CoRR, abs/2306.12925, 2023.

S. Sakshi, Utkarsh Tyagi, Sonal Kumar, Ashish Seth, Ramaneswaran Selvakumar, Oriol Nieto, Ramani Duraiswami, Sreyan Ghosh, and Dinesh Manocha. MMAU: A Massive Multi-Task Audio Understanding and Reasoning Benchmark. In ICLR. OpenReview.net, 2025.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V. Le, Ed H. Chi, Denny Zhou, and Jason Wei. Challenging BIG-Bench Tasks and Whether Chain-of-Thought Can Solve Them. In ACL (Findings), Findings of ACL, pages 13003–13051. Association for Computational Linguistics, 2023.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. CommonsenseQA: A Question Answering Challenge Targeting Commonsense Knowledge. In NAACL-HLT (1), pages 4149–4158. Association for Computational Linguistics, 2019. doi: 10.18653/v1/N19-1421.

Weiting Tan, Hirofumi Inaguma, Ning Dong, Paden Tomasello, and Xutai Ma. SSR: Alignment-Aware Modality Connector for Speech Language Models. In Proceedings of the 22nd International Conference on Spoken Language Translation (IWSLT 2025), pages 56–75, 2025. doi: 10.18653/v1/2025.iwslt-1.5.

Changli Tang, Wenyi Yu, Guangzhi Sun, Xianzhao Chen, Tian Tan, Wei Li, Lu Lu, Zejun Ma, and Chao Zhang. SALMONN: Towards Generic Hearing Abilities for Large Language Models. In ICLR. OpenReview.net, 2024.

Teknium. OpenHermes 2.5: An Open Dataset of Synthetic Data for Generalist LLM Assistants. Hugging Face repository, 2023. URL https://huggingface.co/datasets/teknium/OpenHermes-2.5.

Shubham Toshniwal, Wei Du, Ivan Moshkov, Branislav Kisacanin, Alexan Ayrapetyan, and Igor Gitman. OpenMathInstruct-2: Accelerating AI for Math with Massive Open-Source Instruction Data. CoRR, abs/2410.01560, 2024.

Chaoren Wang, Heng Lu, Xueyao Zhang, Shujie Liu, Yan Lu, Jinyu Li, and Zhizheng Wu. Closing the Modality Reasoning Gap for Speech Large Language Models. In Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 18821–18835. Association for Computational Linguistics, 2026a. doi: 10.18653/v1/2026.acl-long.857.

Chen Wang, Minpeng Liao, Zhongqiang Huang, Jinliang Lu, Junhong Wu, Yuchen Liu, Chengqing Zong, and Jiajun Zhang. BLSP: Bootstrapping Language-Speech Pre-training via Behavior Alignment of Continuation Writing. CoRR, abs/2309.00916, 2023.

Chen Wang, Minpeng Liao, Zhongqiang Huang, Junhong Wu, Chengqing Zong, and Jiajun Zhang. BLSP-Emo: Towards Empathetic Large Speech-Language Models. In Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 19186–19199. Association for Computational Linguistics, 2024a. doi: 10.18653/v1/2024.emnlp-main.1070.

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, 2026b. URL https://arxiv.org/abs/2506.04779.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. MMLU-Pro: A More Robust and Challenging Multi-Task Language Understanding Benchmark. In Advances in Neural Information Processing Systems, volume 37, pages 95266–95290. Curran Associates, Inc., 2024b. doi: 10.52202/079017-3018.

Yuxiang Wang, Shengbo Cai, Yingda Shen, Ming-Hao Hsu, Qinke Ni, Liqiang Zhang, Teddy Sun, Steve Yevs, and Zhizheng Wu. EvoAudio: Recursive Self-Improvement for Audio Understanding. CoRR, abs/2609.27389, 2026.

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. CoRR, abs/2503.20215, 2025.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 Technical Report. CoRR, abs/2412.15115, 2024.

Mu Yang, Szu-Jui Chen, Jiamin Xie, and John H. L. Hansen. Bridging the Modality Gap: Softly Discretizing Audio Representation for LLM-based Automatic Speech Recognition. In ASRU, pages 1–7. IEEE, 2025.

Xiang Yue, Xingwei Qu, Ge Zhang, Yao Fu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. MAmmoTH: Building Math Generalist Models through Hybrid Instruction Tuning. CoRR, abs/2309.05653, 2023.

Yongyi Zang, Sean O'Brien, Taylor Berg-Kirkpatrick, Julian McAuley, and Zachary Novack. Are you really listening? Boosting Perceptual Awareness in Music-QA Benchmarks. CoRR, abs/2504.00369, 2025. URL `https://arxiv.org/abs/2504.00369`.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a Machine Really Finish Your Sentence? In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 4791–4800. Association for Computational Linguistics, 2019. doi: 10.18653/v1/P19-1472.

Dong Zhang, Shimin Li, Xin Zhang, Jun Zhan, Pengyu Wang, Yaqian Zhou, and Xipeng Qiu. SpeechGPT: Empowering Large Language Models with Intrinsic Cross-Modal Conversational Abilities. In EMNLP (Findings), Findings of ACL, pages 15757–15773. Association for Computational Linguistics, 2023.

Xin Zhang, Dong Zhang, Shimin Li, Yaqian Zhou, and Xipeng Qiu. SpeechTokenizer: Unified Speech Tokenizer for Speech Large Language Models. In ICLR, 2024.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-Following Evaluation for Large Language Models, 2023. URL `https://arxiv.org/abs/2311.07911`.

Kun Zhou, Berrak Sisman, Rui Liu, and Haizhou Li. Seen and Unseen Emotional Style Transfer for Voice Conversion with a New Emotional Speech Dataset. In ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing, pages 920–924. IEEE, 2021. URL `https://arxiv.org/abs/2010.14794`.

A Training Details

This appendix describes what was trained for each row of Table 1, including the adaptation scope, the training data, the training configurations, and the optimization settings.

Table 4: **Trainable and Frozen Parameters.** Counts cover the Whisper-large-v3 encoder, the C-Gate scorer, and Qwen2.5-7B-Instruct.

Component	Parameters	Status
C-Gate scorer	2,492,417	Trained
Decoder attention projections, layers 0–23	704,753,664	Trained
Decoder MLPs and normalization, layers 0–23	4,888,633,344	Frozen
Decoder layers 24–27 and final normalization	932,234,752	Frozen
Input embedding table	544,997,376	Frozen
Language-model head	544,997,376	Frozen
Whisper-large-v3 encoder	636,968,960	Frozen
Total	8,255,077,889	707,246,081 trained

A.1 Adaptation Scope

Table 4 lists the trainable and frozen parameters.

A.2 Training Data

ASR. ASR uses the 281,241 utterances of LibriSpeech train-clean-100, train-clean-360, and train-other-500.

Emotion Recognition. The emotion pool draws on the English part of CAMEO [Christop and Czajka, 2025], which comprises CREMA-D [Cao et al., 2014], EMNS [Noriy et al., 2023], eNTERFACE [Martin et al., 2006], JL-Corpus [James et al., 2018], and RAVDESS [Livingstone and Russo, 2018], together with ESD English [Zhou et al., 2021], IEMOCAP [Busso et al., 2008], SAVEE [Jackson and Haq, 2014], TESS [Dupuis and Pichora-Fuller, 2011], and MELD audio [Poria et al., 2019]. We map the source labels to eight classes and split the 55,398 recordings of the pool by recording into a training portion of 53,251 recordings and 47.55 hours and a test portion of 2,147 recordings and 1.95 hours. After removing recordings longer than 30 seconds, 53,246 recordings are used for training. MELD and RAVDESS are used for training only, so the emotion test split consists of the 2,063 test recordings of CREMA-D, EMNS, eNTERFACE, JL-Corpus, ESD, IEMOCAP, SAVEE, and TESS, spoken by 132 speakers.

Reasoning. The reasoning sources are ARC [Clark et al., 2018], CommonsenseQA [Talmor et al., 2019], GSM8K [Cobbe et al., 2021], MATH [Hendrycks et al., 2021b], MathInstruct [Yue et al., 2023], MMLU-Pro [Wang et al., 2024b], NuminaMath [Li et al., 2024], OpenMath2 [Toshniwal et al., 2024], and OrcaMath [Mitra et al., 2024], with every question synthesized as speech. They contain 1,312,333 examples, of which 1,286,024 remain after removing empty answers and audio longer than 30 seconds. The reasoning audio lasts at least 712 hours.

Table 5: **Training Configurations behind Table 1.** Steps count training-loop iterations, and four iterations make one optimizer update. Each configuration is trained once.

Configuration	Training tasks	Evaluated step
ASR-only	ASR	100,000
Emotion-only	Emotion	85,000
Reasoning-only	Reasoning	100,000
Two-task	ASR, emotion	90,000
Three-task	ASR, emotion, reasoning	60,000

A.3 Training Configurations

Table 5 gives the evaluated checkpoint of each configuration. All configurations share the bridge, the adaptation scope, the learning rates, and the batch size. They differ in the training tasks, in training length, and in the maximum target length, which is 384 tokens for the three-task model and 256 tokens otherwise. The reasoning-only model was trained on an earlier reasoning pool that additionally contains OpenHermes 2.5 instruction data [Teknium, 2023] rendered as speech.

A.4 Optimization

We use AdamW with coefficients $(0.9, 0.999)$, weight decay 0.01, and gradient clipping at 1.0. The learning rate is 10^{-4} for the bridge and 2×10^{-5} for decoder attention, with a linear warm-up over 5,000 optimizer updates followed by a cosine schedule whose horizon is 300,000 optimizer updates for the joint configurations and 100,000 for the single-task configurations. Each training-loop iteration processes four micro-batches of two examples on each of eight GPUs, and the optimizer is updated every four iterations, which gives the effective batch size of 256. Training uses bfloat16 mixed precision. The data sampler is seeded.

A.5 Task-Loss Reweighting

Multi-task configurations combine the per-task cross-entropy losses as

$$\mathcal{L}^{(s)} = \sum_{i \in \mathcal{T}} w_i^{(s)} \mathcal{L}_i^{(s)}, \qquad w_i^{(s)} \propto \left(\frac{\mathcal{L}_i^{\mathrm{EMA},s}}{\mathcal{L}_i^{\mathrm{init}}} \right)^{\alpha}, \tag{7}$$

where s indexes training-loop iterations and $\mathcal{T}$ is the task set. A task whose exponential moving average (EMA) loss has fallen less, relative to its initial loss, receives a larger weight. The rule is inspired by dynamic task weighting [Liu et al., 2019] and replaces that method's adjacent-epoch ratio and softmax normalization with an EMA-to-initial-loss ratio. We use $\alpha = 1$, EMA decay 0.9, a 1,000-step warm-up, and weights clipped to $[0.2, 5.0]$. Single-task configurations optimize their cross-entropy loss directly.

A.6 Compute

Training C-Gate to the reported checkpoint took 18.75 hours on eight H100 80 GB GPUs, about 150 GPU-hours. The reasoning-only model was trained on eight A100 40 GB GPUs. Evaluating one model on both LibriSpeech splits takes 34 to 44 minutes on one A100 80 GB GPU.

B Evaluation Protocols

This appendix describes how each reported number is produced. Table 6 lists the evaluation sets.

Table 6: **Evaluation Sets.** Every row uses the checkpoints in Table 5 unless a step is given.

Evaluation	Items
ASR	LibriSpeech test-clean, 2,620 utterances, and test-other, 2,939 utterances.
Emotion test split	2,063 test recordings of eight corpora, with ESD, IEMOCAP, and CREMA-D in Table 1.
Spoken BBH	4,392 questions in 22 categories.
SpeechMMLU and MMAU	8,549 and 1,000 questions.
Text-only benchmarks	The same items for the base and adapted models.
Routing statistics	The emotion test split, the first 500 test-clean utterances, and 1,000 VoiceBench BBH questions.
Emotion probes	The emotion test split.
Fixed-checkpoint interventions	The emotion test split and the first 500 test-clean utterances.
Reasoning audio replacement	1,000 VoiceBench BBH questions, C-Gate at 70,000 steps.

B.1 ASR

The instruction is "Transcribe the following speech." Decoding is greedy with at most 256 new tokens. The prompt embeddings precede the audio embeddings. Audio is sampled at 16 kHz and truncated to 30 seconds, which affects nine test-clean and seven test-other utterances whose full reference transcripts stay in the denominator. We uppercase references and predictions, split on whitespace, and divide the total number of word errors by the total number of reference words. For C-Gate, applying Whisper's English text normalizer instead gives 3.83% and 8.23% WER, and excluding the truncated utterances gives 3.88% and 8.21%. The teacher-forced WER in Table 13 appends the reference tokens to the same prompt and audio, predicts each token from its reference prefix, and scores the decoded argmax sequence.

B.2 Emotion Recognition

The instruction is "The speaker sounds", which matches the training target "The speaker sounds {emotion}." Decoding is greedy with at most 32 new tokens. We lowercase the output and take the first whole-word occurrence of a label or alias, where the labels are angry, happy, sad, neutral, fearful, disgusted, surprised, and calm, and the aliases are anger, happiness, sadness, fear, disgust, and surprise. Outputs without a label count as incorrect.

B.3 Reasoning

For spoken BBH, SpeechMMLU, and MMAU, the text prompt contains the written question, with answer options where the task has them, followed by an answer-format instruction. The audio carries the spoken question for BBH and SpeechMMLU and the audio clip under question for MMAU. Generation is limited to 8 tokens for yes/no, true/false, and valid/invalid BBH categories, 32 tokens for free-form BBH categories, and 64 tokens for multiple-choice questions. Answer extraction is category-specific for BBH and letter-based for multiple-choice questions. Our SpeechMMLU protocol scores generated answers, whereas the benchmark's source scores choice likelihoods [Tan et al., 2025].

B.4 Spoken BBH

Spoken BBH combines the 1,000 questions of the four VoiceBench BBH categories, which are hyperbaton, navigate, sports understanding, and web of lies, with 3,392 questions from 18 further BBH categories [Suzgun et al., 2023, Chen et al., 2026]. For the further categories, we take the question text and reference answers from the BBH release, filter for questions that can be read aloud, and synthesize the audio with Edge TTS using the `en-US-AriaNeural` voice. Accuracy and parse rate weight the 4,392 questions equally.

B.5 Text-Only Benchmarks

The base and adapted models are evaluated through the text path of the decoder with lm-evaluation-harness 0.4.5 [EleutherAI, 2024]. Examples, chat template, shot counts, decoding, parsers, and seeds are identical for the two models.

C Additional Results

This appendix expands the results of Section 5 with a comparison to published speech LLMs, per-corpus and per-class emotion results, the reasoning controls, further text-only benchmarks, and a comparison of adaptation scopes.

C.1 ASR with Limited Adaptation Data

Table 7 places C-Gate beside published speech LLMs to give scale context for its WER. C-Gate starts from pretrained Whisper-large-v3 and Qwen2.5-7B-Instruct and adapts them with 960 hours of transcribed speech, together with the emotion and spoken-reasoning data, by updating the bridge and the decoder attention projections. Two of the reference systems add a large-scale audio pretraining

Table 7: **LibriSpeech WER (%) of C-Gate and Published Speech LLMs.** Reference scores are copied from the original reports, which differ in training data, backbone, and text normalization.

System	test-clean	test-other
Qwen-Audio	2.0	4.2
Qwen2-Audio	1.6	3.6
Qwen2.5-Omni-7B	1.8	3.4
Kimi-Audio	1.28	2.42
Audio Flamingo 3	1.57	3.13
WavLLM	2.0	4.8
SALMONN	2.1	4.9
AlignFormer	3.52	6.47
C-Gate	3.98	8.42

stage: Kimi-Audio pretrains on approximately 13 million hours of raw audio [KimiTeam et al., 2025], and the second pretraining stage of Qwen2.5-Omni-7B includes 300 billion audio-related tokens [Xu et al., 2025]. A fixed-codebook interface therefore supports transcription without an audio pretraining stage beyond the pretrained encoder, within 0.46 points of the nearest reference system on test-clean and 1.95 points on test-other.

The reference rows are copied from the original reports. The Qwen-Audio, Qwen2-Audio, Qwen2.5-Omni-7B, Kimi-Audio, and Audio Flamingo 3 entries come from their technical reports [Chu et al., 2023, 2024, Xu et al., 2025, KimiTeam et al., 2025, Goel et al., 2025]. The WavLLM, SALMONN, and AlignFormer entries come from their papers [Hu et al., 2024, Tang et al., 2024, Fan et al., 2025]. Qwen2-Audio uses its original-framework results, SALMONN its 13B model, and AlignFormer its audio-first configuration before supervised fine-tuning. These systems differ from C-Gate in training data, backbone, adaptation scope, and text normalization.

C.2 Emotion Accuracy by Corpus and Class

Table 8: **Emotion Accuracy by Corpus on the Emotion Test Split.** Accuracy (%) of the three C-Gate models trained with emotion data on the test recordings of our train/test split of each corpus.

Corpus	Recordings	Emotion-only	Two-task	Three-task
ESD	892	95.4	95.7	94.7
IEMOCAP	478	63.4	63.8	64.6
CREMA-D	388	75.0	75.5	74.2
TESS	132	100.0	98.5	100.0
eNTERFACE	56	94.6	98.2	96.4
EMNS	53	86.8	84.9	100.0
JL-Corpus	49	95.9	85.7	93.9
SAVEE	15	60.0	73.3	80.0
All	2,063	84.0	84.1	84.3

Table 8 reports accuracy on the emotion test split for every corpus, including the three shown in Table 1. On every corpus with more than 100 recordings the three models agree within two points, and accuracy on these corpora ranges from 63–65% on IEMOCAP to at least 94.7% on ESD and TESS. Over the whole test split, accuracy is 84.0%, 84.1%, and 84.3% for the emotion-only, two-task, and three-task models. Table 9 reports recall by class. Recall is lowest for the disgusted and fearful classes in all three models, and the three-task model has the highest recall for sad and surprised.

C.3 Reasoning and Answer Extraction

Table 10 separates answer extraction from correctness on spoken BBH. The reasoning-only model often starts an explanation that the generation limit cuts off, so 20.9% of its outputs have no extractable answer, against 0.8% for the three-task model. Among parsed outputs the two models score 36.1% and 43.9%, and always giving the most frequent answer of each category scores 33.4%.

Table 9: **Emotion Recall by Class on the Emotion Test Split.** Recall (%) within each reference class for the three C-Gate models trained with emotion data. The calm class does not occur in the test split.

Class	Recordings	Emotion-only	Two-task	Three-task
Angry	488	86.7	85.5	84.0
Disgusted	90	78.9	66.7	68.9
Fearful	96	63.5	72.9	68.8
Happy	432	81.7	85.4	83.8
Neutral	387	82.7	83.7	82.4
Sad	354	88.1	83.9	89.8
Surprised	216	88.9	91.7	93.5
Accuracy	2,063	84.0	84.1	84.3

Table 10: **Spoken BBH Accuracy and Answer Extraction.** Scores are percentages over 4,392 questions. Parsed accuracy counts only outputs from which an answer could be extracted. Majority is the accuracy of always giving the most frequent answer of each category.

Model	Accuracy	Parse rate	Parsed accuracy
Reasoning-only	28.6	79.1	36.1
Three-task	43.5	99.2	43.9
Majority	33.4	–	–

Table 11: **Replacing the Audio on VoiceBench BBH.** Accuracy (%) of C-Gate at 70,000 steps on 1,000 questions, 250 per category, with the written question kept in the text prompt. All four categories have two answer options. Majority is the accuracy of always giving the most frequent answer of each category.

Audio input	Overall	Parse rate	Hyperbaton	Navigate	Sports	Web of lies
Spoken question	57.9	99.3	60.0	57.2	63.6	50.8
Noise matched to the waveform level	54.8	97.7	50.0	56.8	63.6	48.8
Noise at a fixed level	55.0	97.9	50.0	56.4	63.2	50.4
Zeros	55.4	99.5	49.2	57.6	64.0	50.8
Majority	53.7	–	51.6	58.0	54.0	51.2

Table 11 tests how much the answers depend on the audio when the written question remains in the prompt. We replace the waveform with Gaussian noise scaled to the level of the original waveform, Gaussian noise at a fixed level, or zeros. This diagnostic uses a different instruction format from the benchmark runs of Table 1, so its absolute accuracy is not comparable with that table. Spoken questions score 57.9%, replaced audio scores 54.8–55.4%, and the majority answer scores 53.7%. The difference is confined to hyperbaton, an adjective-ordering task, where accuracy falls from 60.0% to about 50%, and the three yes/no categories change by at most 2.0 points. These observations locate the reasoning scores in answer formatting and in the written question.

C.4 Text-Only Benchmarks

Table 12 extends Table 2. MMLU [Hendrycks et al., 2021a] accuracy falls by 4.22 points after adaptation, and GSM8K [Cobbe et al., 2021] accuracy with flexible answer extraction is unchanged. Both benchmarks are related to the reasoning training data, which contain GSM8K and MMLU-Pro, so they do not measure retention on unseen material. The strict GSM8K metric requires the "#### answer" format of the GSM8K references, which the base model rarely produces and the adapted model has been trained on. The instruction-level IFEval metric falls by 10.91 points, in line with the prompt-level metric in Table 2.

Table 12: **Additional Text-Only Results for Base and Speech-Adapted Qwen2.5-7B-Instruct.** Scores are percentages, and differences are percentage points computed before rounding. Items are prompts for IFEval. The reasoning training data contain GSM8K and MMLU-Pro.

Benchmark and metric	Items	Base	Adapted	Difference
MMLU, 5-shot accuracy	14,042	73.35	69.14	−4.22
GSM8K, 5-shot, flexible answer extraction	1,319	69.45	69.52	+0.08
GSM8K, 5-shot, strict answer format	1,319	8.11	68.69	+60.58
IFEval, instruction-level strict accuracy	541	79.62	68.71	−10.91

Table 13: **Adaptation Scope under an Earlier Training Recipe.** Both models use the C-Gate bridge and the earlier recipe described in this appendix, and differ only in which decoder parameters of layers 0–23 are trained. Scores are percentages, with AR and TF denoting autoregressive and teacher-forced WER on LibriSpeech test-clean and test-other. Emotion is accuracy on the emotion test split.

Trained decoder parameters	AR WER		TF WER		Emotion	BBH	SpMMLU	MMAU
	Clean	Other	Clean	Other				
All parameters	4.51	9.35	4.08	7.99	66.8	37.4	68.2	47.3
Attention projections	5.70	10.62	5.40	9.48	53.8	40.1	72.3	50.3

C.5 Adaptation Scope under an Earlier Recipe

Table 13 compares two models that use the C-Gate bridge and differ only in the trained decoder parameters of layers 0–23. One model trains all parameters of these layers, 5.59 billion in total, and the other trains the attention projections, as C-Gate does. Both come from an earlier recipe that differs from C-Gate's in data and schedule. It uses 460 hours of LibriSpeech, the emotion training recordings, spoken Alpaca instructions, and text-only replay with fixed sampling ratios, includes no reasoning speech and no loss reweighting, and is evaluated at 300,000 steps. Under this recipe, training all decoder parameters gives lower WER and higher emotion accuracy, and training attention projections gives higher scores on the three reasoning benchmarks.

D Routing and Representation Diagnostics

This appendix gives the protocols and full results behind Sections 6.1–6.3.

D.1 Routing Statistics

Table 14: **Support Statistics beyond the Renormalized Weights.** Means over valid frames. Full entropy is the entropy of the softmax over all 152,064 embedding rows divided by its maximum. Mass is the probability that this softmax assigns to the 16 selected entries, in units of 10^{-4}, where a uniform distribution assigns 1.05. Cosine compares each bridge vector with the unweighted mean of its selected embeddings. Overlap is the Jaccard index between the supports of adjacent frames.

Model	Input task	Frames	Full entropy	Mass	Cosine	Overlap
Emotion-only	Emotion	84,663	0.9983	2.94	0.984	0.13
ASR-only	ASR	49,574	0.9916	9.96	0.945	0.16
Two-task	Emotion	84,663	0.9965	4.13	0.964	0.14
	ASR	49,574	0.9960	4.26	0.974	0.13
Three-task	Emotion	84,663	0.9995	2.39	0.950	0.25
	ASR	49,574	0.9995	2.57	0.941	0.26
	Reasoning	279,819	0.9995	2.55	0.942	0.29

Table 3 reports the renormalized weights over the selected support, and Table 14 adds statistics of the softmax before truncation and of the supports. Valid frames are the bridge frames that cover the recording, as opposed to the padding of the 30-second encoder window. All statistics are means over the valid frames of the emotion test split for emotion input, the first 500 test-clean utterances for ASR input, and 1,000 VoiceBench BBH questions for reasoning input. The softmax over all 152,064 embedding rows is nearly flat before truncation. Its entropy is at least 99.1% of the maximum in every row of the table, and the 16 selected entries hold between 2.4×10^{-4} and 1.0×10^{-3} of the probability mass, 2 to 10 times what a uniform distribution assigns to 16 entries. Truncation and renormalization, more than the sharpness of the softmax, therefore determine the mixture. Adjacent frames share few selected tokens, with a mean Jaccard overlap between 0.13 and 0.29.

D.2 Gradient of the Renormalized Weights

Near-uniform weights could indicate a saturated scorer that no longer receives gradient. For a fixed support S_t, the renormalized weights are a softmax over the selected scores:

$$\alpha_{t,v} = \frac{\exp z_{t,v}}{\sum_{u \in S_t} \exp z_{t,u}}, \qquad \frac{\partial \alpha_{t,v}}{\partial z_{t,u}} = \alpha_{t,v}(\mathbf{1}_{v=u} - \alpha_{t,u}), \quad u, v \in S_t. \tag{8}$$

At uniform weights over k entries, this Jacobian has $k - 1$ eigenvalues equal to $1/k$, whereas it vanishes when a single weight approaches one. Near-uniform weights therefore leave the scorer trainable wherever the support is unchanged.

D.3 Selected Tokens

On the 84,663 valid frames of the emotion test split, C-Gate selects 19,997 distinct tokens, and 43 of them account for half of all selections. The most frequent tokens are word pieces unrelated to the input, such as “CSS”, “foundation”, “repeat”, and “.day”, together with Chinese word pieces, and each of them occurs in nearly every recording. A token that spells an emotion label or one of its aliases appears in the support of 70 frames, 0.08% of the total. The supports thus act as a learned code for the decoder and do not name the emotion.

D.4 Emotion Probes

Table 15: **Linear Emotion Probes on C-Gate Representations.** Accuracy (%) on the emotion test split is the mean $\pm$ standard deviation over ten random speaker-disjoint partitions.

Representation	Dimension	Accuracy
Mean-pooled Whisper states	1,280	57.6 ± 5.7
Bag of selected token identities	1,000	63.3 ± 5.2
Mean-pooled bridge output	3,584	66.8 ± 3.9
PCA of pooled bridge output	32	67.8 ± 4.0
Most frequent class	–	23.7

Table 15 reports multinomial logistic-regression probes with regularization parameter $C = 1$ on C-Gate. We assign the 132 speakers of the emotion test split at random to five folds, score every recording once, and repeat this over ten random partitions. Features are computed over the valid frames of each recording and standardized with training-fold statistics. The bag-of-tokens features are the relative frequencies of the 1,000 tokens selected most often in the training folds, and the PCA is fitted on the training folds. The bag exceeds the pooled Whisper states in all ten partitions, by 3.5 to 8.2 points.

D.5 Attention Rank

Figure 2 uses 100 recordings of the emotion test split and 100 test-clean utterances, taken at regular intervals from the sets of Table 6. The input follows the evaluation layout: the chat-formatted instruction, the valid bridge frames, and the teacher-forced reference answer. The answer is “The speaker sounds {emotion}.” for emotion, which has five tokens, and the reference transcript for ASR,

where three utterances with fewer than five answer tokens are excluded. Rows of the attention matrix are the first five answer tokens, columns are the audio frames, and attention is averaged over heads without renormalizing the rows. For the three models trained on ASR, the ASR rank is highest at layers 14 and 15, between 4.43 and 4.56, and averages 3.05 to 3.12 over the 28 layers. Over all answer tokens of an utterance, 35.7 on average, the ASR rank at layer 14 is 30.1 to 30.5 for these models and 11.1 for the emotion-only model. Under the emotion condition, the rank at layer 14 is 2.71 to 2.87 for the models trained on emotion and 3.41 for the ASR-only model.

E Fixed-Checkpoint Interventions

This appendix gives the protocols and values behind Sections 6.4 and 6.5. Every intervention changes one component of the trained C-Gate checkpoint at inference and keeps the weights, prompts, decoding, and scoring fixed. All interventions use the 2,063 recordings of the emotion test split and the first 500 test-clean utterances, and unparsed emotion outputs count as incorrect.

Table 16: **Replacing the Waveform.** Scores (%) of one C-Gate checkpoint on the emotion test split and the first 500 test-clean utterances.

Audio input	Emotion accuracy	WER
Original waveform	84.29	4.01
Zeros	17.16	97.71
Gaussian noise at the waveform level	20.02	98.84

Table 17: **Replacing the Mixture Weights and Perturbing the Key–Value Mapping.** Scores (%) of one C-Gate checkpoint on the emotion test split and the first 500 test-clean utterances, with prompt embeddings and all model weights unchanged. Differences are relative to the unchanged bridge in percentage points, with paired 95% block-bootstrap intervals. Fig. 3 plots the unchanged bridge and the three mapping conditions.

C-Gate bridge	Emotion accuracy	Difference [95% interval]	WER	Difference [95% interval]
Unchanged	84.29	–	4.01	–
Uniform weights	81.63	−2.67 [−4.88, −1.36]	5.85	+1.84 [+1.16, +2.46]
Value permutation	11.78	−72.52 [−78.76, −63.63]	434.18	+430.17 [+364.39, +498.07]
Gaussian values	12.65	−71.64 [−78.88, −61.29]	97.39	+93.38 [+92.74, +93.93]
Key permutation	9.99	−74.31 [−82.49, −62.42]	228.24	+224.23 [+177.51, +271.30]

Table 18: **Graded Temporal Shuffling.** Scores (%) are means over three permutation seeds on the emotion test split and the first 500 test-clean utterances. Window 1 preserves order, and Full shuffles the entire sequence. This table gives the values plotted in Fig. 4.

Window	1	2	4	8	16	64	Full
Emotion accuracy	84.29	78.40	71.43	63.31	56.42	52.25	52.25
ASR WER	4.01	12.30	50.91	100.32	142.11	165.87	187.52

E.1 Audio Replacement

Table 16 replaces the waveform with zeros or with Gaussian noise scaled to the root-mean-square level of the original waveform. Emotion accuracy falls below the 23.7% share of the most frequent class, and WER approaches 100%.

E.2 Mixture Weights and Key–Value Mapping

Table 17 changes one component inside C-Gate. Uniform weights keep the 16 tokens selected for each frame and set every weight to $1/16$, so each bridge vector becomes the unweighted mean of its selected embeddings. Value permutation applies one random permutation to the embedding rows

that the selected indices retrieve. Key permutation applies it to the rows from which the keys are projected, so that scores select different entries while the values stay in place. Gaussian values replace the embedding rows with independent Gaussian vectors rescaled to the mean and standard deviation of each embedding dimension. The ordinary text path of the decoder uses the original embedding table in all conditions.

E.3 Temporal Shuffling

Table 18 permutes the bridge frames within non-overlapping windows, with a separate permutation for each utterance and seed. The three seeds vary only the permutations.

Table 19: **Support-Size Sensitivity without Retraining.** Scores (%) use the emotion test split and the first 500 test-clean utterances, with the model trained at $k = 16$. Differences are relative to $k = 16$ in percentage points, with paired 95% block-bootstrap intervals. This table gives the values plotted in Fig. 5.

k	Emotion accuracy	Difference [95% interval]	WER	Difference [95% interval]
1	55.89	−28.41 [−31.61, −24.34]	337.10	+333.09 [+315.24, +354.71]
4	80.27	−4.02 [−6.24, −2.36]	12.84	+8.83 [+8.00, +9.73]
8	84.54	+0.24 [−0.97, +1.23]	5.18	+1.17 [+0.68, +1.73]
16	84.29	–	4.01	–
32	82.65	−1.65 [−3.58, −0.33]	3.89	−0.12 [−0.45, +0.18]
64	79.11	−5.19 [−8.92, −3.15]	4.84	+0.83 [+0.25, +1.57]

Table 20: **Pooling-Stride Sensitivity without Retraining.** Scores (%) use the emotion test split and the first 500 test-clean utterances, with the model trained at stride 4. Frames gives the mean number of bridge frames per ASR utterance. Differences are relative to stride 4 in percentage points, with paired 95% block-bootstrap intervals.

Stride	Frames	Emotion accuracy	Difference [95% interval]	WER	Difference [95% interval]
2	197.8	81.63	−2.67 [−4.93, −1.24]	14.07	+10.06 [+8.53, +11.69]
4	99.1	84.29	–	4.01	–
8	49.8	70.29	−14.01 [−16.91, −10.28]	36.50	+32.49 [+29.41, +35.41]

E.4 Support Size and Stride

Tables 19 and 20 change the support size or the pooling stride at inference for the checkpoint trained with $k = 16$ and stride 4. The intervals are paired percentile bootstrap intervals from 10,000 resamples of the 132 speakers of the emotion test split or the 9 LibriSpeech speakers, without adjustment for multiple comparisons. A stride of 2 doubles the number of bridge frames and a stride of 8 halves it, so the stride conditions change the sequence length as well as the temporal resolution.

F Cost of Full-Vocabulary Scoring

This appendix counts the operations of the bridge. C-Gate scores every vocabulary entry for every frame before selecting k entries, so selection does not reduce the scoring cost. With $V = 152{,}064$ embedding rows, $d_{\mathrm{llm}} = 3{,}584$, $d_p = 512$, and $T' = 375$ frames for a 30-second input, projecting the keys costs $Vd_{\mathrm{llm}}d_p \approx 2.8 \times 10^{11}$ multiply–accumulate operations and scoring costs $T'Vd_p \approx 2.9 \times 10^{10}$ per input. The $T' \times V$ score matrix has 5.7×10^7 entries, which occupy 109 MiB in bfloat16 and 218 MiB in 32-bit floating point. Key projection runs once per batch during training and can be cached at inference, because the keys depend only on fixed weights. Mixing the selected values adds $T'kd_{\mathrm{llm}}$ operations. These counts cover the bridge alone and exclude end-to-end latency.

Table 21: **Evidence for Each Design Choice.**

Design choice	Evidence
Convex constraint	Weights are nonnegative and sum to one over frozen embedding rows (Eq. 4), so hull membership holds by construction.
Mixture weights	Setting every weight to 1/16 on the same selected tokens lowers emotion accuracy by 2.7 points and raises WER by 1.8 points (Section 6.1).
Learned key–value mapping	Permuting keys or values, or replacing values with Gaussian vectors, degrades both tasks (Section 6.4).
Temporal order	Windowed shuffles degrade both tasks, more for larger windows, in three seeds (Section 6.4).
Support size and stride	Changing k or the stride at inference changes both scores (Section 6.5).
Partial decoder adaptation	Parameter account (Table 4) and text-only evaluation (Tables 2 and 12).
Loss reweighting	Used in both multi-task configurations (Appendix A), with no separate ablation.

G Evidence for Each Design Choice

Table 21 summarizes the evidence in this paper for each design choice. The interventions on the mixture weights, the key–value mapping, the temporal order, the support size, and the stride are applied at inference to one trained checkpoint without retraining.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract states the convex-hull construction, the partial decoder adaptation, the joint task performance, and the analysis findings. Sections 5 and 6 support each of them, and Section 6 presents the interventions as sensitivity of one trained checkpoint.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 7 discusses single training runs, the matched ablations of the convex-hull constraint and of attention adaptation left to future work, text regressions, the weak audio dependence of the reasoning scores, and the cost of full-vocabulary scoring. Appendix G summarizes the evidence for each design choice.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper contains no theorems. Convex-hull membership follows directly from the nonnegative, normalized weights in Eq. 4.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Sections 3 and 4 and Appendices A and B specify the bridge, the trained parameters, the data, the optimization settings, and the evaluation protocols.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility.

In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: The paper does not link code or checkpoints. Its training and evaluation sources are public datasets and benchmarks, some rendered as speech, and Appendices A and B describe the training and evaluation procedures.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 4 and Appendices A and B report the data, the splits, the optimizer settings, the evaluated checkpoints, and the prompts, decoding, and scoring rules.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Each configuration is trained once, so Table 1 has no error bars. Appendix E reports paired bootstrap intervals for the support-size and stride interventions, and Table 15 reports fold variability for the probes.

   Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: Appendix A reports the hardware and training time of the main model and the evaluation time per model, but not the training time of the other configurations. Exact training times of the ablations were not recorded.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The work uses existing corpora and models without new human-subject experiments and discloses text regressions and emotion-inference limitations. The authors reviewed the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Section 7 discusses accessible voice interfaces, unreliable natural-emotion inference, intrusive speaker profiling, and reduced instruction following after adaptation.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper releases no data or models.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [No]

    Justification: The paper cites the creators of every model, dataset, and benchmark it uses, and Appendix A lists the training sources. It does not state the license of each asset.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The study uses previously released speech and reasoning corpora without new crowdsourcing or participant studies.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The study evaluates existing datasets without new recruitment or interventions with human participants.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: Section 4.1 identifies the partially adapted Qwen2.5-7B-Instruct decoder and frozen Whisper-large-v3 encoder. Language-model assistance supported prose revision, consistency checks, and manuscript preparation, without generating experimental observations.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.