Rethinking Entropy Allocation in LLM-based ASR: Understanding the Dynamics between Speech Encoders and Large Language Models

Yuan Xie*, **Jiaqi Song***, **Xianliang Wang**, **Ming Lei**, **Jie Gao**
LLM Team, Advanced Intelligence Systems Group, NIO
{ryan.xie2, jiaqi.song2}@nio.com

Abstract

Integrating large language models (LLMs) into automatic speech recognition (ASR) has become a dominant paradigm. Although recent LLM-based ASR models have shown promising performance on public benchmarks, it remains challenging to balance recognition quality with latency and overhead, while hallucinations further limit real-world deployment. In this study, we revisit LLM-based ASR from an entropy allocation perspective and introduce three encoder-side diagnostics to characterize how training paradigms shape uncertainty reduction across the speech encoder–LLM interface. To remedy entropy-allocation inefficiencies in prevailing approaches, we propose a capability-boundary-aware multi-stage training strategy that targets parameter efficiency and robustness to hallucinations. Specifically, we redesign the pretraining strategy to alleviate the speech-text modality gap, and further introduce an iterative asynchronous SFT stage between alignment and joint SFT to preserve functional decoupling and constrain encoder representation drift. Experiments on various benchmarks show that our method achieves competitive performance against state-of-the-art models using only 2.3B parameters, while also effectively mitigating hallucinations through our decoupling-oriented design.

1 Introduction

With the rapid advancement of large language models (LLMs), the mainstream automatic speech recognition (ASR) paradigm is shifting from traditional architectures [7, 10, 20, 21] toward LLM-based frameworks. Recent LLM-based ASR (hereafter LLM-ASR) models, including Seed-ASR [4], FireRedASR-LLM [52], Voxtral Mini Transcribe [33], Fun-ASR [2], and Qwen3-ASR [43], are specialized for transcription. By leveraging LLMs' world knowledge and contextual reasoning to resolve semantic ambiguity, LLM-ASR models have achieved promising results on public benchmarks. In contrast to Large Audio-Language Models (LALMs) [11, 19, 49], which target broad audio understanding tasks beyond ASR, LLM-ASR is more suitable for industrial speech applications, where accuracy, latency, computational overhead, and controllability are all critical concerns, and thus serves as the focus of this study.

Despite strong benchmark performance, LLM-ASR still faces two practical challenges in real-world deployment. First, efficiency and recognition quality remain difficult to balance. In lightweight settings, LLM-ASR models suffer not only from expected scaling-down degradation, but also from the inherent cost of bridging the speech-text modality gap, which consumes non-trivial model capacity [1, 62] and imposes a disproportionate tax on smaller models [17]. The second challenge is hallucination. During joint training, the encoder is susceptible to being dominated by the LLM's

*Equal contribution; Listing order is random.

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

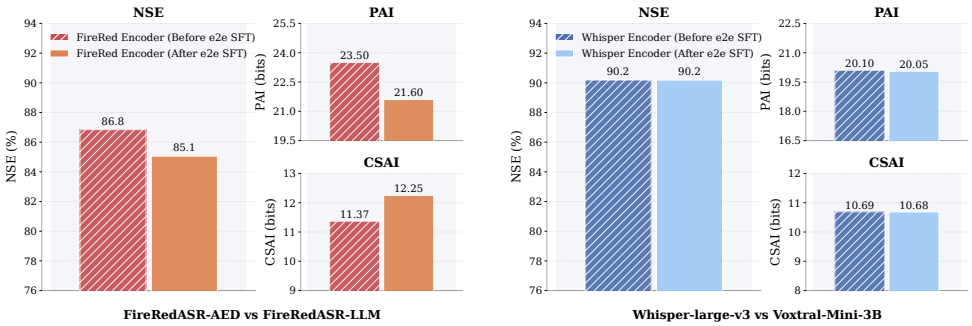


Figure 1: The shift in encoder-side metrics before and after joint training. We hold the encoder architecture fixed within each group to examine the impact of joint training on encoder representations.

gradients, which can induce representation drift: encoder representations may increasingly encode linguistic shortcuts at the expense of acoustic fidelity, thereby amplifying hallucination risk [5, 63].

To step beyond empirical observation and provide a principled diagnosis of these limitations, we revisit LLM-ASR from an entropy allocation perspective, viewing ASR as compressing high-entropy speech signals into low-entropy linguistic symbols. From this perspective, we propose a set of encoder-side metrics (defined in Section 3.2) to diagnose how different training paradigms shape the encoder–LLM interface. Specifically, normalized spectral entropy (NSE) quantifies the spectral entropy of encoder representations, while phonetic accessible information (PAI) and conditional semantic accessible information (CSAI) serve as probe-inspired proxies for linearly accessible phonetic information and residual semantic information beyond the phonetic proxy, respectively. Figure 1 compares the metric shifts before and after end-to-end joint training across two representative ASR model families, revealing distinct behavioral regimes. From FireRedASR-AED to FireRedASR-LLM with the same inherited Conformer encoder, joint training markedly reduces NSE, indicating increased encoder-side compression. Yet, the concurrent decrease in PAI and increase in CSAI suggest that this compression is associated with a shift away from phonetic specialization toward semantic accessibility, consistent with the representation drift discussed above. In contrast, Voxtral's Whisper encoder shows minimal changes during joint training, with all metrics remaining largely stable. Compared to FireRedASR, it exhibits higher NSE but lower PAI and CSAI, indicating a lighter encoder-side entropy-reduction load and a greater share of residual uncertainty left to the LLM. While this mitigates representation drift, it increases downstream capacity demand and forces the LLM to resolve phonetic uncertainties outside its primary modeling strengths, leading to reduced parameter efficiency.

These observations reveal two suboptimal regimes in prevailing LLM-ASR systems: one exhibits representation drift and over-reliance on semantic priors, while the other offloads residual uncertainty to the LLM at the cost of parameter efficiency. Motivated by this diagnosis, we propose a capability-boundary-aware design principle that refines multi-stage training. We redesign the pretraining strategy to ground encoder representations in phonemes at a structural granularity approaching text tokens, which reduces modality alignment cost and separates acoustic modeling from premature semantic anchoring. We further introduce an iterative asynchronous SFT (IA-SFT) stage between alignment and joint SFT, which deepens the speech-text modality alignment while explicitly preserving functional decoupling to avoid representation drift. This refined multi-stage design encourages each module to devote more of its capacity to its designated role, improving parameter efficiency and mitigating hallucinations. Our main contributions are summarized as follows.

- We present an entropy-based perspective for LLM-ASR, with metrics on encoder representations that characterize how training paradigms allocate entropy reduction between encoder and LLM.
- We propose a capability-boundary-aware multi-stage training paradigm that redesigns encoder pretraining and introduces an intermediate IA-SFT stage, jointly promoting clear module specialization and more stable joint training.
- Our model achieves competitive performance across Mandarin, dialect, English, and code-switching benchmarks with only 2.3B parameters, while effectively mitigating hallucinations, facilitating efficient and robust real-world deployment.

2 Related Work

Recent advances in LLMs have redefined ASR, shifting the paradigm from acoustic transcription toward semantically informed generation. By coupling speech encoders with LLMs, LLM-ASR leverages rich world knowledge and long-context semantic modeling to enhance recognition. In recent research, the encoder–adaptor–LLM architecture has emerged as a standard backbone, with variations primarily arising from training strategies and functional priorities.

LLM-based ASR models. Recent LLM-ASR models have emerged along diverse design principles. One line emphasizes lightweight adaptation: for instance, FireRedASR-LLM [52] achieves strong performance on Mandarin and English benchmarks via LoRA fine-tuning. Another line of works such as Seed-ASR [4] and Fun-ASR [2] scale up training with industrial-scale speech corpora, and further leverage context-SFT and reinforcement learning (RL) to extend model capabilities. Moreover, a related trend leverages pretrained multi-modal LLM backbones to build specialized ASR models—for example, Qwen3-ASR [43] built upon Qwen3-Omni [51] and GPT-4o Transcribe [36] derived from GPT-4o [26]. These models inherit extensive speech understanding knowledge and apply targeted optimizations for ASR capabilities. Additionally, general-purpose LALMs such as GLM-4-Voice [59], Step-Audio 2 [49], and Kimi-Audio [15] can also serve as relevant baselines in ASR benchmarking, though seldom deployed for practical ASR applications. Although LLM-ASR models can achieve strong benchmark results, latency and deployment cost remain major constraints for real-time speech interaction. As a result, many works [2, 43] have released lightweight variants; however, these typically rely on straightforward capacity reduction without principled changes to the training paradigm, leaving a substantial performance gap relative to their larger counterparts. Moreover, the inherent modality gap between acoustic and textual representations inevitably amplifies hallucinations [38], posing challenges to the reliability required for real-world deployment.

Divergent paths in uncertainty resolution. Most LLM-ASR models adopt the encoder–adaptor–LLM architecture, but differ in how uncertainty is resolved across the encoder and the LLM. During encoder pretraining, supervised ASR objectives such as CTC, AED, or their hybrid variants [40, 50, 52] encourage the encoder to learn transcription-discriminative representations and resolve much of the acoustic uncertainty before LLM integration. In contrast, SSL objectives such as Best-RQ [9] are used in several works [2, 4] to learn more general-purpose acoustic representations. These choices lead to distinct internal operating regimes, analyzed in Appendix C. Following encoder pretraining, LLM-ASR models typically proceed through an alignment stage and then joint training with the LLM under natural-language supervision [25], which further shapes how uncertainty is resolved. As the system maps high-dimensional speech representations into low-dimensional symbolic sequences, the granularity retained in the encoder representations directly determines how much entropy reduction must be handled by the downstream LLM, thereby dictating its effective capacity. In principle, the optimal scale of the LLM-ASR model should be jointly informed by its training paradigm, yet existing approaches remain largely data-driven and pay limited attention to this relationship.

3 Methodology

3.1 Preliminary

LLM-based ASR as conditional generation. Let x denote the input speech features (e.g., log-Mel spectrograms), $t = (t_1, \ldots, t_M)$ denote the text instruction prompt, and $y = (y_1, \ldots, y_N)$ denote the output transcriptions. We formulate LLM-based ASR as conditional language modeling with parameters $\theta = \{\phi, \psi, \omega\}$, corresponding to the speech encoder, adaptor, and LLM decoder, respectively. The speech encoder $\mathcal{E}_\phi$ maps acoustic features to continuous representations, and the adaptor $\mathcal{A}_\psi$ projects them into the LLM embedding space:

$$E = \mathcal{E}_\phi(x) \in \mathbb{R}^{L \times d_e}, \qquad Z = \mathcal{A}_\psi(\mathrm{Down}_r(E)) \in \mathbb{R}^{L' \times d_m}, \tag{1}$$

where L is the original frame length, L' is the length after downsampling, d_e is the encoder hidden size, d_m is the LLM embedding dimension, and $\mathrm{Down}_r(\cdot)$ denotes downsampling with factor r. Let $\mathcal{W}(\cdot)$ denote the LLM text token embedding operator. We then construct a unified continuous input sequence by concatenating the text prompt embeddings and the projected speech embeddings:

$$\mathbf{Z}_{\mathrm{prompt}}(x, t) = \big[\mathcal{W}(t_1), \ldots, \mathcal{W}(t_M)\,; Z_1, \ldots, Z_{L'}\big] \in \mathbb{R}^{(M+L') \times d_m}. \tag{2}$$

where $\{Z_\ell\}_{\ell=1}^{L'}$ are speech embeddings produced by the encoder–adaptor stack. The LLM decoder $\mathcal{D}_\omega$ then generates the transcription autoregressively:

$$p_\theta(y \mid x, t) = \prod_{n=1}^{N} p_\omega(y_n \mid y_{<n}, \mathbf{Z}_{\text{prompt}}(x, t)) \,. \tag{3}$$

Entropy allocation at the encoder–LLM interface. The encoder–LLM interface is constituted by the speech embedding Z that the LLM directly consumes. Since $\mathcal{A}_\psi$ is a lightweight MLP, the structural properties of Z are largely inherited from E, which we therefore analyze as the encoder-side proxy of the interface. Given a fixed transcription target per utterance, the encoder and LLM face a complementary trade-off in resolving uncertainty: what the encoder absorbs upstream reduces what the LLM must resolve downstream, making E a principled lens on cross-module entropy allocation.

3.2 Metrics on Encoder Representations

For each utterance, let $E' \in \mathbb{R}^{L_{valid} \times d_e}$ denote the valid-frame representation obtained from E after removing padded positions according to the sequence mask. We characterize E' from two complementary aspects: the overall spectral entropy retained in representations, and proxy estimates of how uncertainty reduction remains accessible in phonetic and semantic target spaces.

Normalized spectral entropy (NSE). We perform singular value decomposition (SVD) on the encoder representation matrix $E' = U\Sigma V^\top$, where $\Sigma = \text{diag}(\sigma)$, with $\sigma = (\sigma_1, \ldots, \sigma_d)^\top$ being singular values in descending order and $d = \min(L_{valid}, d_e)$. Treating the normalized singular values as a distribution [42], we define normalized spectral entropy [55] as

$$\text{NSE}(E') = -\frac{1}{\log d} \sum_{i=1}^{d} \bar{\sigma}_i \log \bar{\sigma}_i, \quad \bar{\sigma}_i = \frac{\sigma_i}{\sum_{j=1}^{d} \sigma_j}, \tag{4}$$

which captures the global spectral geometry of E'. Unlike multi-task regimes where isotropy is typically desirable, LLM-ASR has a single objective; here, high NSE reflects an isotropic representation space with excess entropy, whereas low NSE reflects a more anisotropic, compressed one.

Phonetic and conditional semantic accessible information. While NSE quantifies the global spectral entropy retained in E', it remains agnostic to whether the entropy aligns with task-relevant subspaces. We therefore introduce two complementary probe-inspired accessible-information proxies: **phonetic accessible information (PAI)**, measuring how much phonetic information is linearly accessible from E'; and **conditional semantic accessible information (CSAI)**, measuring how much residual semantic information is accessible beyond what the phonetic target space linearly explains. The dynamics of PAI and CSAI can reflect how the representation space evolves during training.

To derive them, we first summarize the valid-frame representation E' via temporal mean and standard deviation pooling, followed by Principal Component Analysis (PCA) and standardization, yielding a vector u. For the phonetic target P, we convert the reference transcript into phoneme tokens, form an ℓ_1-normalized bag-of-phones count vector, and apply the same PCA-standardization pipeline. For the semantic target S, we encode the transcript using a frozen Qwen3-Embedding-8B [54] model with last-token pooling and ℓ_2 normalization, followed by PCA and standardization. In practice, P and S are projected into the same dimension.

Let $q = [u^\top, P^\top, S^\top]^\top$ and let $\hat{\Sigma} = \text{Cov}(q)$ denote the empirical covariance estimated over the evaluation set. We use a ridge-regularized covariance $\tilde{\Sigma} = \hat{\Sigma} + \lambda I$, where $\lambda > 0$ ensures numerical stability. All covariance blocks below are taken as principal submatrices of $\tilde{\Sigma}$. Under a joint Gaussian approximation on (u, P, S), we define

$$\text{PAI}(E') = \left[\frac{1}{2 \log 2} \log \frac{\det \tilde{\Sigma}_{uu} \, \det \tilde{\Sigma}_{PP}}{\det \tilde{\Sigma}_{[u,P]}} \right]_+ \tag{5}$$

where $[\cdot]_+ = \max(\cdot, 0)$ clips residual negative values caused by numerical error. This quantity serves as a regularized Gaussian accessible-information proxy for the mutual information between u and P. Similarly, letting $\tilde{\Sigma}_{\cdot|P}$ denote the corresponding regularized conditional covariance matrices given P,

computed from the same joint covariance $\tilde{\Sigma}$, we define

$$\mathrm{CSAI}(E') = \left[\frac{1}{2 \log 2} \log \frac{\det \tilde{\Sigma}_{uu|P} \, \det \tilde{\Sigma}_{SS|P}}{\det \tilde{\Sigma}_{[u,S]|P}} \right]_+ \tag{6}$$

which serves as a regularized Gaussian proxy for the conditional accessible-information between u and S given P. Intuitively, CSAI captures the semantic information in u unexplained by the bag-of-phones target, by removing the portion of u–S dependence mediated by the phonetic proxy. It is worth noting that PAI and CSAI, derived under a Gaussian approximation with transcript-based targets, serve as relative within-protocol proxies rather than exact mutual-information estimates.

Together, NSE, PAI, and CSAI form a practical diagnostic: NSE quantifies the spectral entropy, while PAI and CSAI quantify phonetic accessibility and residual semantic accessibility. Jointly analyzing these metrics helps diagnose entropy-allocation dynamics across model components. As discussed in Section 1 regarding Figure 1, the trends of these metrics can reveal suboptimal behavior during training. In a desirable training trajectory, we expect NSE not to remain persistently high, thereby reducing the capacity demand on the LLM; meanwhile, we expect CSAI not to exhibit a sustained rise at the expense of declining PAI during joint training, as this pattern indicates that the improvement stems not from representation refinement but from linguistic shortcuts acquired at the cost of acoustic fidelity—a trend that can amplify hallucination risk.

3.3 Design Principle

With these metrics, we can provide an intuitive diagnosis of entropy allocation in LLM-ASR models. Figure 1 exposes two contrasting suboptimal modes, which have been analyzed in detail in Section 1. We therefore adopt a **capability-boundary-aware** design principle with two requirements:

- The encoder should be guided toward low-entropy, acoustically grounded representations before being exposed to LLM-dominated joint optimization. This narrows the modality gap early and reduces the risk that subsequent training resorts to semantically biased shortcuts.
- During joint optimization, the functional boundary between modules should be explicitly maintained, ensuring that further encoder compression proceeds along an acoustically grounded direction rather than at the expense of acoustic representation quality.

3.4 Multi-Stage Training Paradigm

As shown in Figure 2, we illustrate our proposed training pipeline alongside a comparison with the traditional one. Our design features two core components: phoneme-level CTC pretraining, and an additional IA-SFT stage that runs asynchronously in parallel with the pretraining stage.

Phoneme-level CTC Pretraining. The pretraining stage starts from a Conformer encoder initialized with FireRedASR-AED weights, with the original autoregressive decoder replaced by a lightweight linear CTC head trained under the CR-CTC [56] objective. Motivated by prior findings that phoneme-based representations can offer a more universal and acoustically grounded interface than grapheme or subword units [57], we adopt phoneme-level rather than character- or subword-level supervision.

It is worth noting that we intentionally adopt CTC-based pretraining rather than the more prevalent AED-style or self-supervised alternatives. This choice is driven by two structural considerations. First, the CTC objective, through its peaky behavior [60] and monotonic alignment constraint, encourages the encoder to compress continuous speech into representations more aligned with the underlying token sequence [65], bridging the structural gap with discrete text. Second, the lightweight CTC head acts as a capacity bottleneck, pushing the encoder toward low-entropy representations. Together, these properties narrow the speech–text modality gap at the representation level, while the lower-entropy encoder representation relaxes the capacity demand on the downstream LLM through the interface.

Alignment and IA-SFT. In traditional training paradigms, alignment and joint SFT are conducted sequentially after pretraining completes. To explicitly constrain representation drift, we introduce an additional IA-SFT stage between alignment and joint SFT. As shown in Figure 2, neither alignment nor IA-SFT waits for pretraining to complete; instead, they proceed asynchronously in parallel with pretraining. To govern encoder initialization and updates for the alignment and IA-SFT stages, we monitor representation drift using Centered Kernel Alignment (CKA) [30], which is applied to

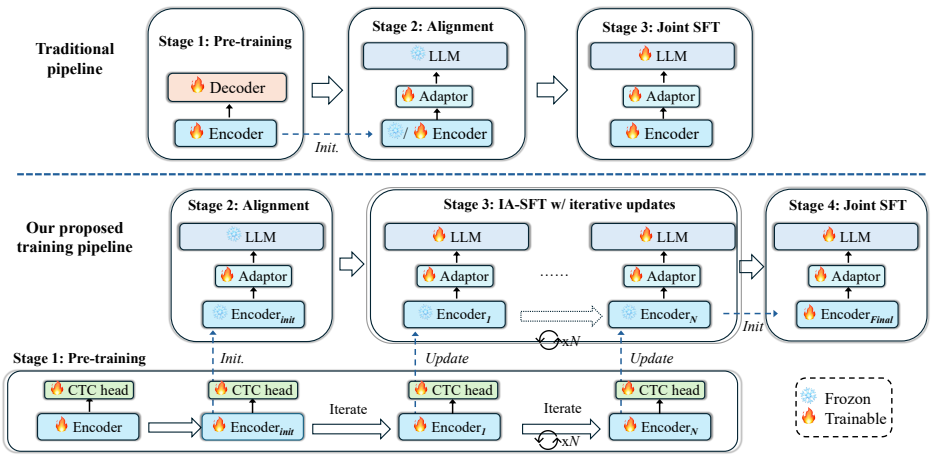


Figure 2: Comparison of our multi-stage training design with the traditional training pipeline.

compare the evolving encoder with a reference checkpoint that is initialized and updated throughout pretraining. The detailed procedure is as follows:

(1) Alignment: Once pretraining has run for a sufficient number of steps, we snapshot the encoder to initialize the reference checkpoint. As pretraining continues, we monitor the CKA score between the evolving encoder and reference checkpoint. When the CKA score first drops below a preset threshold, we take a new snapshot that both initializes the speech encoder for alignment and updates the reference checkpoint. During alignment, both the encoder and LLM are frozen, and only the adaptor is trained;

(2) IA-SFT: After alignment completes, the model enters the IA-SFT stage, where the encoder is frozen while the adaptor and LLM are trained. In essence, IA-SFT also serves an alignment purpose—it can be viewed as a preparatory curriculum that strengthens the LLM's capacity to comprehend speech representations before joint SFT. To expose the LLM to diverse encoder representations, we perform CKA-driven iterative encoder hot-swapping: whenever the CKA score drops below the threshold, the latest encoder checkpoint from the ongoing pretraining is hot-swapped into the frozen encoder used in IA-SFT, and the reference checkpoint is updated accordingly. The CKA constraint ensures that each swap delivers a substantive shift in encoder representations, enabling the LLM to learn from progressively refined encoders in a curriculum-like fashion. Throughout the IA-SFT stage, this iterative hot-swapping cycle repeats until pretraining concludes. Further details of IA-SFT are supplemented in Appendix B.4, including a more detailed procedural description, the formal definition of CKA, specific training configurations, and an analysis of why direct encoder hot-swapping without realignment is viable.

Joint SFT. After IA-SFT, we proceed to joint SFT, where all modules are trained end-to-end. By this point, the modality gap has been substantially narrowed, and the speech representations can preserve their acoustic-phonetic structure. This enables the LLM to reliably interpret speech-derived representations, thereby mitigating the risk of representation drift during joint training. Notably, although an additional stage is introduced, both alignment and IA-SFT run asynchronously in parallel with pretraining from its midpoint, ensuring that our overall pipeline remains time-efficient in large-scale industrial settings.

3.5 Empirical Analysis of Metric Dynamics

To empirically examine whether our multi-stage training paradigm adheres to the design principles, we trace the proposed metrics across training stages and compare them against the direct transfer path from FireRedASR-AED to FireRedASR-LLM, as illustrated in Figure 3. The alignment stage is omitted from the figure, as no metric change occurs in this encoder-frozen stage. Together, the metric trajectories reveal not only how much entropy reduction the encoder assumes across stages, but also whether such compression remains acoustically grounded or drifts toward semantic bias.

During phoneme-level CTC pretraining, NSE drops noticeably, as the CTC bottleneck compels the encoder to form compact, low-entropy representations. Meanwhile, PAI decreases while CSAI rises

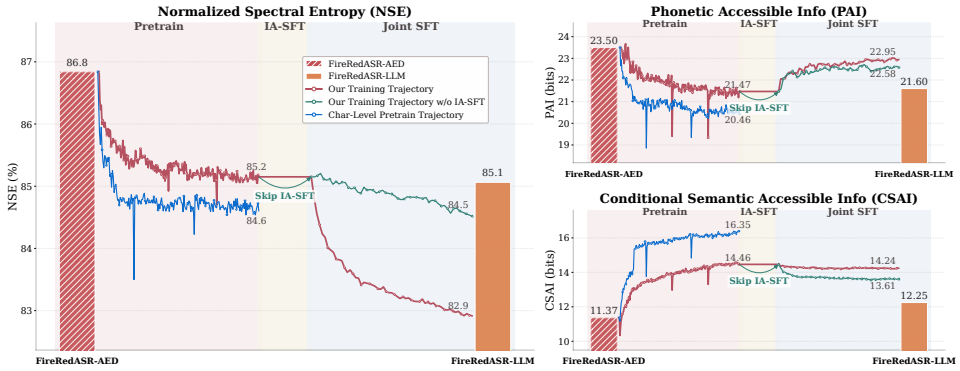


Figure 3: Comparison of the three metrics along our training trajectory and the FireRedASR-AED → FireRedASR-LLM transition, sharing the same encoder architecture. "Our Training Trajectory" corresponds to the full standard "phoneme-level pretraining → IA-SFT → joint SFT" pipeline.

steadily. Given that only phoneme-level supervision is provided, the PAI decline does not indicate acoustic information loss; rather, it is a structural consequence of the peaky, sparse output distributions induced by CTC. Such distributions bias representations toward discrete, symbol-like forms, which dilute the globally accessible linear phonetic structure after pooling and PCA. The moderate CSAI increase follows naturally, as monotonic alignment progressively drives representations toward finer token-level granularity. Moreover, we present a comparison between phoneme-level and character-level pretraining variants. Phoneme-level pretraining consistently achieves higher PAI and lower CSAI than its character-level counterpart throughout training, suggesting an advantage in instilling a robust acoustic foundation while suppressing premature semantic anchoring in the encoder.

During joint SFT, NSE continues to decrease along our training trajectory, indicating that end-to-end optimization further drives the encoder toward lower-entropy representations and progressively relieves downstream capacity pressure on the LLM. Furthermore, a key pattern emerges in our trajectory: PAI recovers notably while CSAI remains flat or declines slightly—consistent with our design expectations. Since joint optimization begins from an already-narrowed modality gap and a well-aligned interface, it mitigates representation drift and instead promotes a desirable cross-modal division of labor. Specifically, LLM gradients incentivize the encoder to refine phonetic-acoustic representations rather than exploit linguistic shortcuts, while the encoder refrains from absorbing semantic functions better delegated to the LLM. This emergent modular separation reflects efficient parameter utilization across the architecture.

To quantify the contribution of IA-SFT, we present an ablation in which this stage is removed (green curves in Figure 3). Without IA-SFT, NSE decays more slowly relative to our full training trajectory, while both PAI and CSAI converge to lower final values—confirming degraded linear accessibility along both acoustic and semantic dimensions, or equivalently, a reduction in the effective signal-to-noise ratio of encoder representations. These results underscore the critical role of IA-SFT as a representational buffer: by progressively deepening the alignment between the LLM's embedding manifold and encoder representations, it both preserves the acoustic grounding established during pretraining and shields against representation drift in the subsequent joint optimization stage.

4 Experiments

4.1 Evaluation setting

We evaluate on public ASR benchmarks covering Mandarin [6, 16], Chinese dialects [13, 31, 45], English [37, 48], and Chinese–English code-switching [34, 64]. We report Character Error Rate (CER) for Chinese and Word Error Rate (WER) for English. Beyond LLM-ASR baselines, we include two large-scale LALMs [49, 51] as references. Despite their substantially higher latency and overhead, they help assess competitiveness beyond lightweight settings. All transcriptions are normalized with the *WeTextProcessing* toolkit. All models are evaluated in offline mode with their official decoding configurations, and our model uses beam search with a beam size of 3.

Table 1: Comparison with advanced LLM-ASR and LALM baselines on public benchmarks.

	Fun-ASR-nano	**GLM-ASR-nano**	**Qwen3-ASR-1.7B**	**FireRedASR-LLM**	**Step-Audio2-mini**	**Qwen3-Omni-Inst**	**Ours**
Model Size	0.8B	1.5B	2.0B	8B+	8B+	30B-A3B	2.3B
Mandarin							
AISHELL-1 *dev* | *test*	1.59 | 1.81	2.40 | 2.41	1.40 | 1.51	0.71 | 0.73	0.76 | 0.81	0.86 | 0.92	**0.45** | **0.59**
AISHELL-2-ios *dev* | *test*	2.62 | 2.73	3.21 | 3.45	2.41 | 2.60	**2.08** | **2.12**	2.24 | 2.29	2.11 | 2.31	2.32 | 2.45
AISHELL-2021-*Eval A* | *C* | *D*	4.75 | 4.29 | 2.33	7.25 | 9.48 | 3.40	4.22 | 3.51 | 1.82	12.61 | 4.06 | 7.38	4.54 | 3.69 | 2.34	5.19 | 3.34 | **1.66**	**3.45** | **1.71** | 2.53
Chinese Dialect							
WeNetSpeech-Chuan *easy* | *hard*	12.69 | 23.76	20.95 | 33.61	11.40 | **20.35**	12.14 | 24.76	13.99 | 25.35	14.13 | 25.16	**10.94** | 21.93
WeNetSpeech-Yue *short* | *long*	7.31 | 10.02	16.78 | 13.97	5.79 | **8.00**	- | -	7.78 | 8.44	6.97 | 8.60	**5.22** | 9.45
KeSpeech	7.18	9.59	4.98	**3.53**	3.98	6.00	4.56
English							
LibriSpeech-dev *clean* | *other*	1.63 | 4.06	1.82 | 3.93	1.54 | 3.14	1.25 | 2.92	**1.06** | 2.48	1.08 | **2.10**	1.11 | 2.57
LibriSpeech-test *clean* | *other*	1.63 | 4.35	1.96 | 4.29	1.56 | 3.49	1.37 | 3.36	1.22 | 2.61	**1.15** | **2.38**	1.23 | 2.63
VoxPopuli *dev* | *test*	7.86 | 7.70	8.78 | 8.52	7.58 | 7.42	10.65 | 10.26	8.86 | 8.37	6.86 | 6.75	**6.25** | **6.22**
Chinese–English Code-switch							
CS-Dialogue	5.37	6.15	5.44	5.10	9.46	8.51	**4.99**
ASCEND	11.91	12.29	**10.87**	11.25	13.50	18.68	11.79
Avg. CER/WER	**6.28**	**8.71**	**5.45**	**6.46**	**6.19**	**6.24**	**5.12**

4.2 Main Recognition Results

Table 1 compares our model with advanced ASR baselines [2, 43, 49, 51, 52, 58]. With only 2.3B parameters, our model achieves competitive results across multilingual benchmarks, surpassing several industrial-scale models with over 8B parameters on multiple scenarios. Beyond classic benchmarks such as AISHELL and LibriSpeech, our model attains SOTA performance on entity-dense sets like AISHELL-2021-Eval (in-car, telephony), suggesting that aligning the LLM to low-entropy speech representations does not cause catastrophic forgetting of world knowledge. Furthermore, our model achieves leading results on several dialect benchmarks, indicating strong robustness to phonetic shifts induced by pronunciation variation. This behavior is supported by the elevated PAI values in Figure 3, validating our acoustically grounded design principle. Our model also performs well on code-switching benchmarks, owing in part to the phoneme-level pretraining strategy, which leverages language-agnostic phonetic representations shared across languages, thereby enabling more robust modeling under code-switching conditions.

4.3 Hallucination Mitigation

Table 2: Hallucination rates on scenario benchmarks.

Model	Mandarin	Dialect	English	Code-switch
Fun-ASR-nano	0.018%	0.217%	0.014%	0.397%
GLM-ASR-nano	0.030%	0.201%	0.014%	0.315%
Qwen3-ASR-1.7B	0.018%	**0.120%**	0.014%	0.345%
FireRedASR-LLM	0.053%	0.228%	0.014%	0.324%
Step-Audio2-mini	0.020%	0.194%	0.014%	1.255%
Qwen3-Omni-Inst	0.013%	0.370%	**0.007%**	1.778%
Ours (w/o IA-SFT)	0.005%	0.198%	0.014%	0.356%
Ours	**0.003%**	0.122%	**0.007%**	**0.261%**

A core contribution of our method is its principled approach to hallucination mitigation. To validate this, we use the same models and benchmarks from Table 1 and report the average hallucination rate per scenario, defined as the ratio of hallucinated samples to total samples across all benchmarks within each scenario. Specifically, a sample is classified as hallucinated if its transcription is more than 50% longer than the ground truth and has no unigram overlap with it, i.e., a BLEU-1 score of zero. As shown in Table 2, our model achieves substantially lower hallucination rates than all baselines. This suggests that our design mitigates hallucination at a fundamental level, owing to the explicit constraint during training that enforces the encoder to remain acoustically grounded, thereby suppressing representation drift. We further ablate the IA-SFT module and identify it as a major contributor to our low hallucination rate. Without IA-SFT, even low-entropy representations are susceptible to being influenced by the LLM's gradients, causing the model to fall back on linguistic shortcuts and produce hallucinations—a failure mode that IA-SFT can help mitigate.

4.4 Layer-wise Alignment: From Encoder to Adaptor

While the preceding analysis focuses on encoder representations, we further examine post-adaptor embeddings. Figure 4 visualizes the progression of semantic alignment by computing linear CKA scores between ground-truth text embeddings and representations at each encoder layer as well as the post-adaptor embedding. Linear CKA is the special case of the general CKA defined in Eq. (8), measuring alignment across features of different dimensionalities via centered Gram matrices that capture inter-sample similarity structures. To handle sequence length mismatches, we apply temporal

mean pooling to obtain utterance-level representations $X_l \in \mathbb{R}^{B\times d_e}$ or $Z \in \mathbb{R}^{B\times d_m}$ from the l-th layer (B denotes the number of utterances), with the same pooling applied to ground-truth text embeddings $Y \in \mathbb{R}^{B\times d_m}$:

$$\mathrm{CKA}_{\text{linear}}(X_l, Y) = \frac{\|X_l^\top Y\|_F^2}{\|X_l^\top X_l\|_F \, \|Y^\top Y\|_F}, \tag{7}$$

where $\|\cdot\|_F$ denotes the Frobenius norm. We further include FireRedASR-LLM as a reference, whose encoder and adaptor share the same architectures as ours, enabling a controlled comparison. As shown in Figure 4, our model and FireRedASR-LLM exhibit markedly different behaviors at the adaptor interface. For FireRedASR-LLM, the CKA score decreases after adaptor projection, indicating that its high-entropy encoder representations are difficult to align with the text embedding space. The adaptor has to simultaneously perform dimensional projection, stabilize the interface, and reconcile semantic geometry—a compound burden that exceeds its limited capacity. In contrast, our model consistently shows a pronounced CKA increase at this interface, regardless of whether joint SFT has been applied. Since our encoder already produces low-entropy representations roughly synchronized with text tokens, and IA-SFT further familiarizes the LLM with these representations, the adaptor is largely relieved of compensatory duties. It needs only to perform a straightforward dimensional mapping while correcting residual geometric misalignment, yielding embeddings structurally better matched to the text manifold.

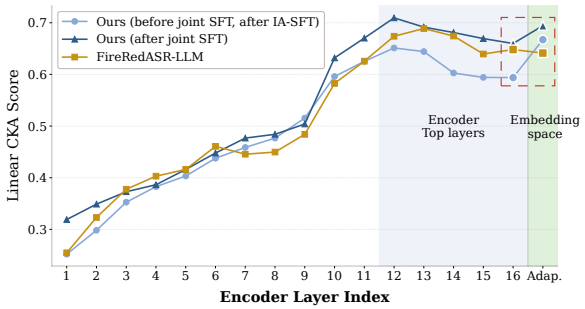


Figure 4: Linear CKA scores between layer-wise representations and ground-truth text embeddings, averaged over 1,000 AISHELL/LibriSpeech utterances; "Adap." denotes the post-adaptor embedding.

4.5 Ablation Study

Table 3: Ablation study on post-training strategies. "– Encoder iter. in IA-SFT" denotes removing encoder hot-swapping and asynchronous parallel during IA-SFT, reducing it to a standard encoder-frozen stage with only the adaptor and LLM trained.

Configuration	Mandarin	Dialect	English	Code-switch
Our full pipeline	**1.93**	**10.42**	**3.34**	**8.39**
– Joint SFT	2.18	12.84	4.22	10.15
– IA-SFT	2.08	11.47	3.79	9.11
– Encoder iter. in IA-SFT	1.95	10.87	3.40	8.57

We conduct ablation studies on post-training strategies under controlled conditions, using identical training data and configurations across all experiments, each trained until validation loss plateaus for three consecutive checkpoints (10k-step intervals). As shown in Table 3, ablating joint SFT yields the largest degradation, confirming that end-to-end optimization is essential for refining the representation space. Moreover, ablating IA-SFT also causes substantial performance drops, consistent with the analysis in Section 3.5 and Table 2: without the drift constraint imposed by IA-SFT, speech representations tend to shift toward the semantic subspace, undermining phoneme discrimination and amplifying hallucination. We further ablate the encoder hot-swapping mechanism within IA-SFT, reducing it to a standard encoder-frozen stage where only the adaptor and LLM are trained using the final encoder checkpoint. This variant still underperforms our full IA-SFT, since static representations from a frozen encoder offer insufficient diversity for robust adaptation, which may make the LLM more prone to suboptimal patterns within a narrow phonetic subspace. In contrast, encoder hot-swapping mitigates this by acting as implicit regularization: exposing the LLM to progressively evolving encoder states encourages learning of robust patterns shared across encoders, rather than those specific to any single encoder.

5 Conclusion and Limitations

In this work, we revisit LLM-based ASR from an entropy-allocation perspective and propose a capability-boundary-aware framework that encourages a clearer division of labor between the encoder and LLM for acoustic uncertainty reduction and semantic ambiguity resolution. Through phoneme-level CTC pretraining and IA-SFT, our approach improves module-wise specialization and achieves strong performance with low hallucination rates across diverse benchmarks using only 2.3B parameters. This study also has several limitations that naturally point to future work, including scaling the framework to larger LLM backbones, extending the entropy-allocation analysis to RL, and developing stronger diagnostics beyond the current linear-Gaussian proxies. A detailed discussion of these limitations and corresponding future directions is provided in Appendix A.

Acknowledgments and Disclosure of Funding

We thank our colleagues in the LLM Team of the Advanced Intelligence Systems Group at NIO for their valuable discussions and support.

Funding and Competing Interests. This research was conducted at NIO. The authors declare no additional external funding or competing interests beyond their disclosed affiliation with NIO.

References

[1] Armen Aghajanyan, Lili Yu, Alexis Conneau, Wei-Ning Hsu, Karen Hambardzumyan, Susan Zhang, Stephen Roller, Naman Goyal, Omer Levy, and Luke Zettlemoyer. Scaling laws for generative mixed-modal language models. In *International Conference on Machine Learning*, pages 265–279. PMLR, 2023.

[2] Keyu An, Yanni Chen, Chong Deng, Changfeng Gao, Zhifu Gao, Bo Gong, Xiangang Li, Yabin Li, Xiang Lv, Yunjie Ji, et al. Funaudio-asr technical report. *arXiv preprint arXiv:2509.12508*, 2025.

[3] Rosana Ardila, Megan Branson, Kelly Davis, Michael Kohler, Josh Meyer, Michael Henretty, Reuben Morais, Lindsay Saunders, Francis Tyers, and Gregor Weber. Common voice: A massively-multilingual speech corpus. In *Proceedings of the twelfth language resources and evaluation conference*, pages 4218–4222, 2020.

[4] Ye Bai, Jingping Chen, Jitong Chen, Wei Chen, Zhuo Chen, Chuang Ding, Linhao Dong, Qianqian Dong, Yujiao Du, Kepan Gao, et al. Seed-asr: Understanding diverse speech and contexts with llm-based speech recognition. *arXiv preprint arXiv:2407.04675*, 2024.

[5] Zechen Bai, Pichao Wang, Tianjun Xiao, Tong He, Zongbo Han, Zheng Zhang, and Mike Zheng Shou. Hallucination of multimodal large language models: A survey. *arXiv preprint arXiv:2404.18930*, 2024.

[6] Hui Bu, Jiayu Du, Xingyu Na, Bengu Wu, and Hao Zheng. Aishell-1: An open-source mandarin speech corpus and a speech recognition baseline. In *2017 20th conference of the oriental chapter of the international coordinating committee on speech databases and speech I/O systems and assessment (O-COCOSDA)*, pages 1–5. IEEE, 2017.

[7] William Chan, Navdeep Jaitly, Quoc Le, and Oriol Vinyals. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 4960–4964. IEEE, 2016.

[8] Guoguo Chen, Shuzhou Chai, Guanbo Wang, Jiayu Du, Wei-Qiang Zhang, Chao Weng, Dan Su, Daniel Povey, Jan Trmal, Junbo Zhang, et al. Gigaspeech: An evolving, multi-domain asr corpus with 10,000 hours of transcribed audio. *arXiv preprint arXiv:2106.06909*, 2021.

[9] Chung-Cheng Chiu, James Qin, Yu Zhang, Jiahui Yu, and Yonghui Wu. Self-supervised learning with random-projection quantizer for speech recognition. In *International Conference on Machine Learning*, pages 3915–3924. PMLR, 2022.

[10] Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio. Attention-based models for speech recognition. *Advances in neural information processing systems*, 28, 2015.

[11] Yunfei Chu, Jin Xu, Qian Yang, Haojie Wei, Xipin Wei, Zhifang Guo, Yichong Leng, Yuanjun Lv, Jinzheng He, Junyang Lin, et al. Qwen2-audio technical report. *arXiv preprint arXiv:2407.10759*, 2024.

[12] Alexis Conneau, Min Ma, Simran Khanuja, Yu Zhang, Vera Axelrod, Siddharth Dalmia, Jason Riesa, Clara Rivera, and Ankur Bapna. Fleurs: Few-shot learning evaluation of universal representations of speech. In *2022 IEEE Spoken Language Technology Workshop (SLT)*, pages 798–805. IEEE, 2023.

[13] Yuhang Dai, Ziyu Zhang, Shuai Wang, Longhao Li, Zhao Guo, Tianlun Zuo, Shuiyuan Wang, Hongfei Xue, Chengyou Wang, Qing Wang, et al. Wenetspeech-chuan: A large-scale sichuanese corpus with rich annotation for dialectal speech processing. *arXiv preprint arXiv:2509.18004*, 2025.

[14] Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*, 2023.

[15] Ding Ding, Zeqian Ju, Yichong Leng, Songxiang Liu, Tong Liu, Zeyu Shang, Kai Shen, Wei Song, Xu Tan, Heyi Tang, et al. Kimi-audio technical report. *arXiv preprint arXiv:2504.18425*, 2025.

[16] Jiayu Du, Xingyu Na, Xuechen Liu, and Hui Bu. Aishell-2: Transforming mandarin asr research into industrial scale. *arXiv preprint arXiv:1808.10583*, 2018.

[17] Mark Endo and Serena Yeung-Levy. Downscaling intelligence: Exploring perception and reasoning bottlenecks in small multimodal models. *arXiv preprint arXiv:2511.17487*, 2025.

[18] Daniel Galvez, Greg Diamos, Juan Ciro, Juan Felipe Cerón, Keith Achorn, Anjali Gopi, David Kanter, Maximilian Lam, Mark Mazumder, and Vijay Janapa Reddi. The people's speech: A large-scale diverse english speech recognition dataset for commercial usage. *arXiv preprint arXiv:2111.09344*, 2021.

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

[20] Alex Graves. Sequence transduction with recurrent neural networks. *arXiv preprint arXiv:1211.3711*, 2012.

[21] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376, 2006.

[22] Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, et al. Conformer: Convolution-augmented transformer for speech recognition. *arXiv preprint arXiv:2005.08100*, 2020.

[23] Haorui He, Zengqiang Shang, Chaoren Wang, Xuyuan Li, Yicheng Gu, Hua Hua, Liwei Liu, Chen Yang, Jiaqi Li, Peiyang Shi, et al. Emilia: An extensive, multilingual, and diverse speech dataset for large-scale speech generation. In *2024 IEEE Spoken Language Technology Workshop (SLT)*, pages 885–890. IEEE, 2024.

[24] François Hernandez, Vincent Nguyen, Sahar Ghannay, Natalia Tomashenko, and Yannick Esteve. Ted-lium 3: Twice as much data and corpus repartition for experiments on speaker adaptation. In *International conference on speech and computer*, pages 198–208. Springer, 2018.

[25] Yukiya Hono, Koh Mitsuda, Tianyu Zhao, Kentaro Mitsui, Toshiaki Wakatsuki, and Kei Sawada. Integrating pre-trained speech and language models for end-to-end speech recognition. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 13289–13305, 2024.

[26] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

[27] Wei Kang, Xiaoyu Yang, Zengwei Yao, Fangjun Kuang, Yifan Yang, Liyong Guo, Long Lin, and Daniel Povey. Libriheavy: A 50,000 hours asr corpus with punctuation casing and context. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 10991–10995. IEEE, 2024.

[28] Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[29] Nithin Rao Koluguri, Monica Sekoyan, George Zelenfroynd, Sasha Meister, Shuoyang Ding, Sofia Kostandian, He Huang, Nikolay Karpov, Jagadeesh Balam, Vitaly Lavrukhin, et al. Granary: Speech recognition and translation dataset in 25 european languages. *arXiv preprint arXiv:2505.13404*, 2025.

[30] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International conference on machine learning*, pages 3519–3529. PMlR, 2019.

[31] Longhao Li, Zhao Guo, Hongjie Chen, Yuhang Dai, Ziyu Zhang, Hongfei Xue, Tianlun Zuo, Chengyou Wang, Shuiyuan Wang, Xin Xu, et al. Wenetspeech-yue: A large-scale cantonese speech corpus with multi-dimensional annotation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 31627–31635, 2026.

[32] Song Li, Yongbin You, Xuezhi Wang, Zhengkun Tian, Ke Ding, and Guanglu Wan. Msr-86k: An evolving, multilingual corpus with 86,300 hours of transcribed audio for speech recognition research. *arXiv preprint arXiv:2406.18301*, 2024.

[33] Alexander H Liu, Andy Ehrenberg, Andy Lo, Clément Denoix, Corentin Barreau, Guillaume Lample, Jean-Malo Delignon, Khyathi Raghavi Chandu, Patrick von Platen, Pavankumar Reddy Muddireddy, et al. Voxtral. *arXiv preprint arXiv:2507.13264*, 2025.

[34] Holy Lovenia, Samuel Cahyawijaya, Genta Indra Winata, Peng Xu, Yan Xu, Zihan Liu, Rita Frieske, Tiezheng Yu, Wenliang Dai, Elham J Barezi, et al. Ascend: A spontaneous chinese-english dataset for code-switching in multi-turn conversation. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 7259–7268, 2022.

[35] Patrick K O'Neill, Vitaly Lavrukhin, Somshubra Majumdar, Vahid Noroozi, Yuekai Zhang, Oleksii Kuchaiev, Jagadeesh Balam, Yuliya Dovzhenko, Keenan Freyberg, Michael D Shulman, et al. Spgispeech: 5,000 hours of transcribed financial audio for fully formatted end-to-end speech recognition. *arXiv preprint arXiv:2104.02014*, 2021.

[36] OpenAI. Introducing next-generation audio models in the API. `https://openai.com/index/introducing-our-next-generation-audio-models/`, 2025.

[37] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5206–5210. IEEE, 2015.

[38] Jing Peng, Yucheng Wang, Bohan Li, Yiwei Guo, Hankun Wang, Yangui Fang, Yu Xi, Haoyu Li, Xu Li, Ke Zhang, et al. A survey on speech large language models for understanding. *arXiv preprint arXiv:2410.18908*, 2024.

[39] Vineel Pratap, Qiantong Xu, Anuroop Sriram, Gabriel Synnaeve, and Ronan Collobert. Mls: A large-scale multilingual dataset for speech research. *arXiv preprint arXiv:2012.03411*, 2020.

[40] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 28492–28518. PMLR, 2023.

[41] Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. Zero: Memory optimizations toward training trillion parameter models. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–16. IEEE, 2020.

[42] Olivier Roy and Martin Vetterli. The effective rank: A measure of effective dimensionality. In *2007 15th European signal processing conference*, pages 606–610. IEEE, 2007.

[43] Xian Shi, Xiong Wang, Zhifang Guo, Yongqi Wang, Pei Zhang, Xinyu Zhang, Zishan Guo, Hongkun Hao, Yu Xi, Baosong Yang, et al. Qwen3-asr technical report. *arXiv preprint arXiv:2601.21337*, 2026.

[44] Ravid Shwartz-Ziv and Naftali Tishby. Opening the black box of deep neural networks via information. *arXiv preprint arXiv:1703.00810*, 2017.

[45] Zhiyuan Tang, Dong Wang, Yanguang Xu, Jianwei Sun, Xiaoning Lei, Shuaijiang Zhao, Cheng Wen, Xingjun Tan, Chuandong Xie, Shuran Zhou, et al. Kespeech: An open source speech dataset of mandarin and its eight subdialects. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

[46] Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. In *2015 IEEE Information Theory Workshop (ITW)*, pages 1–5. IEEE, 2015. doi: 10.1109/ITW.2015.7133169.

[47] Naftali Tishby, Fernando C. Pereira, and William Bialek. The information bottleneck method. In *Proceedings of the 37th Annual Allerton Conference on Communication, Control, and Computing*, 1999.

[48] Changhan Wang, Morgane Riviere, Ann Lee, Anne Wu, Chaitanya Talnikar, Daniel Haziza, Mary Williamson, Juan Pino, and Emmanuel Dupoux. Voxpopuli: A large-scale multilingual speech corpus for representation learning, semi-supervised learning and interpretation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 993–1003, 2021.

[49] Boyong Wu, Chao Yan, Chen Hu, Cheng Yi, Chengli Feng, Fei Tian, Feiyu Shen, Gang Yu, Haoyang Zhang, Jingbei Li, et al. Step-audio 2 technical report. *arXiv preprint arXiv:2507.16632*, 2025.

[50] Yinfeng Xia, Jian Tang, Junfeng Hou, Gaopeng Xu, and Haitao Yao. Uni-asr: Unified llm-based architecture for non-streaming and streaming automatic speech recognition. *arXiv preprint arXiv:2603.11123*, 2026.

[51] Jin Xu, Zhifang Guo, Hangrui Hu, Yunfei Chu, Xiong Wang, Jinzheng He, Yuxuan Wang, Xian Shi, Ting He, Xinfa Zhu, et al. Qwen3-omni technical report. *arXiv preprint arXiv:2509.17765*, 2025.

[52] Kai-Tuo Xu, Feng-Long Xie, Xu Tang, and Yao Hu. Fireredasr: Open-source industrial-grade mandarin speech recognition models from encoder-decoder to llm integration. *arXiv preprint arXiv:2501.14350*, 2025.

[53] Junichi Yamagishi, Christophe Veaux, and Kirsten MacDonald. Cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit (version 0.92). *The Rainbow Passage which the speakers read out can be found in the International Dialects of English Archive:(http://web. ku. edu/˜ idea/readings/rainbow. htm).*, 2019.

[54] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[55] Wenye Yang, Jerry D Gibson, and Tao He. Coefficient rate and lossy source coding. *IEEE Transactions on Information Theory*, 51(1):381–386, 2005.

[56] Zengwei Yao, Wei Kang, Xiaoyu Yang, Fangjun Kuang, Liyong Guo, Han Zhu, Zengrui Jin, Zhaoqing Li, Long Lin, and Daniel Povey. Cr-ctc: Consistency regularization on ctc for improved speech recognition. *arXiv preprint arXiv:2410.05101*, 2024.

[57] Saierdaer Yusuyin, Te Ma, Hao Huang, Wenbo Zhao, and Zhijian Ou. Whistle: Data-efficient multilingual and crosslingual speech recognition via weakly phonetic supervision. *IEEE Transactions on Audio, Speech and Language Processing*, 2025.

[58] Z.AI. GLM-ASR-Nano-2512: A robust and compact speech recognition model. `https://huggingface.co/zai-org/GLM-ASR-Nano-2512`, December 2025.

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

[60] Albert Zeyer, Ralf Schlüter, and Hermann Ney. Why does ctc result in peaky behavior? *arXiv preprint arXiv:2105.14849*, 2021.

[61] Binbin Zhang, Hang Lv, Pengcheng Guo, Qijie Shao, Chao Yang, Lei Xie, Xin Xu, Hui Bu, Xiaoyu Chen, Chenchen Zeng, et al. Wenetspeech: A 10000+ hours multi-domain mandarin corpus for speech recognition. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6182–6186. IEEE, 2022.

[62] Yu Zhang, Mufan Xu, Xuefeng Bai, Pengfei Zhang, Yang Xiang, Min Zhang, et al. Instruction anchors: Dissecting the causal dynamics of modality arbitration. *arXiv preprint arXiv:2602.03677*, 2026.

[63] Guanyu Zhou, Yibo Yan, Xin Zou, Kun Wang, Aiwei Liu, and Xuming Hu. Mitigating modality prior-induced hallucinations in multimodal large language models via deciphering attention causality. *arXiv preprint arXiv:2410.04780*, 2024.

[64] Jiaming Zhou, Yujie Guo, Shiwan Zhao, Haoqin Sun, Hui Wang, Jiabei He, Aobo Kong, Shiyao Wang, Xi Yang, Yequan Wang, et al. Cs-dialogue: A 104-hour dataset of spontaneous mandarin-english code-switching dialogues for speech recognition. *arXiv preprint arXiv:2502.18913*, 2025.

[65] Wei Zhou, Junteng Jia, Leda Sari, Jay Mahadeokar, and Ozlem Kalinli. Cjst: Ctc compressor based joint speech and text training for decoder-only asr. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2025.

A Detailed Limitations and Future Work

While our entropy-allocation framework and the proposed multi-stage training paradigm provide a principled account of the encoder–LLM division of labor in LLM-based ASR, several limitations remain and naturally point to directions for future work.

Scalability to Larger Backbones. This work targets efficient and stable LLM-ASR models and demonstrates strong effectiveness in lightweight deployment scenarios, but the proposed framework is not inherently confined to small-scale models. Due to time and resource constraints, we have not yet explored larger backbones. Future work will instantiate the framework with larger yet inference-efficient backbones, such as MoE-based Qwen3-30B-A3B, to assess whether the capability-boundary-aware design and IA-SFT paradigm generalize to medium-scale models and further improve the performance upper bound of LLM-ASR systems.

Limited training paradigm coverage. This study focuses on encoder pretraining, alignment, and SFT, and does not yet cover reinforcement learning. Given its growing adoption in recent LLM-ASR systems such as Seed-ASR and Fun-ASR, an important future direction is to examine how RL reshapes entropy allocation and whether capability-boundary-aware constraints can be incorporated into reward design or regularization to improve robustness and reduce hallucinations.

Diagnostic Metrics as Relative Proxies. The PAI and CSAI metrics introduced in Section 3.2 are based on a joint Gaussian approximation with ridge-regularized covariance estimation, and are designed as linear-Gaussian accessible-information proxies rather than exact mutual-information estimates. They offer a practical and consistent diagnostic tool for relative comparisons across training paradigms in our setting, while their interpretation naturally depends on the chosen probe representations, such as bag-of-phones for P and frozen LLM-based semantic embeddings for S. Future work will explore tighter non-linear estimators and validate these diagnostics across more encoder families.

B Training Details

B.1 Training Data Statistics

Across the pretraining, alignment, and SFT stages, we use the same speech corpora, with only the training steps and objectives varying at each stage. Table 4 summarizes detailed statistics of the training data, including language coverage, domain diversity, and overall scale. Our training corpora consist of annotated speech-text pairs totaling approximately 560K hours.

B.2 Our LLM-based ASR Architecture

Feature extraction. We extract 80-dimensional log-Mel spectrograms using a 25ms window and a 10ms frame shift, followed by global mean and variance normalization.

Speech encoder. The backbone of our encoder is inherited from FireRedASR-AED [52], consisting of a $4\times$ downsampling convolutional module followed by a stack of Conformer blocks [22], with a total of approximately 600M parameters. The encoder converts speech into continuous representations at a frame rate of 25 Hz (40 ms temporal resolution).

CTC head. For encoder pretraining, we attach a three-layer MLP as a CTC head after the speech encoder, which maps its hidden representations to the target vocabulary and is optimized with the connectionist temporal classification loss [21, 56]. The CTC head is used exclusively during pretraining.

Speech adaptor. A lightweight speech adaptor, composed of an MLP with two linear layers, is responsible for mapping the encoder's speech representations into the text embedding space of the LLM. Prior to the projection, a $4\times$ downsampling is applied by concatenating 4 consecutive frames along the feature dimension to reduce the sequence length. Following this downsampling process, the frame rate is reduced to 6.25 Hz, corresponding to a temporal resolution of 160 ms per token.

Table 4: An overview of the Mandarin and English speech corpora used across all training stages.

Dataset	Language(s)	Domain	Hours
YODAS-Granary [29]	English	Variety	120K
Emilia [23]	English / Mandarin	Variety	189K
MLS [39]	English	Audiobook	45K
MSR-86K [32]	English	YouTube	10K
Common-Voice-v15 [3]	English / Mandarin	Read	3K
GigaSpeech [8]	English	Variety	10K
LibriHeavy [27]	English	Audiobook	50K
LibriSpeech [37]	English	Audiobook	960
SPGISpeech [35]	English	Finance	5K
PeopleSpeech [18]	English	Variety	30K
VCTK [53]	English	Read	25
TEDLIUM3 [24]	English	Talk	500
WeNetSpeech-Yue [31]	Chinese dialects	Variety	22K
WeNetSpeech-Chuan [13]	Chinese dialects	Variety	10K
WenetSpeech [61]	Mandarin	Variety	11K
FLEURS [12]	English / Mandarin	News	100
AISHELL-1 [6]	Mandarin	Read	150
AISHELL-2 [16]	Mandarin	Read	1K
KeSpeech [45]	Mandarin and 8 Subdialects	Conversation	1.6K
CS-Dialogue [64]	Mandarin English Code Switch	Variety	104
ASCEND [34]	Mandarin English Code Switch	Conversation	10
In-house data	English / Mandarin	Conversation	$\sim$50K
Total	English / Mandarin	All	$\sim$560K

LLM decoder. The decoder is initialized from Qwen3-1.7B [54] and generates the final transcription conditioned on both text prompts ("Transcribe the speech into text.") and speech tokens.

B.3 Training Setups

In this study, we adopt stage-specific training configurations described as follows:

Pretraining: A dynamic batch strategy based on speech frames is employed, with a maximum of 20k frames per batch to efficiently handle variable-length utterances. The learning rate follows a schedule with a linear warm-up to 5.0e-4 over the first 8k steps, followed by exponential decay for the remainder of training.

Alignment: The batch size is set to 10k frames. The learning rate follows the same schedule as in pretraining, with a linear warm-up to a maximum of 1.0e-3.

IA-SFT and joint SFT: The batch size is further reduced to 7k frames. The learning rate also follows the same warm-up and exponential decay schedule, with a maximum learning rate of 1.0e-5. In pilot experiments, we also explored lower learning rates during joint SFT to reduce representation drift. Empirically, after IA-SFT has sufficiently narrowed the modality gap, a learning rate of 1.0e-5 maintained the interface stability established in prior stages and performed better than the more conservative alternatives.

The Adam optimizer [28] is used across all training stages. Experiments are conducted on NVIDIA A100 80GB GPUs using DeepSpeed ZeRO Stage-2 [41], with gradient accumulation over 4 steps, bfloat16 precision, and FlashAttention-2 [14].

B.4 Training Details of IA-SFT

Here, we provide additional details on the implementation of IA-SFT.

CKA-guided encoder update schedule. During IA-SFT, we perform several rounds of encoder hot-swapping to ensure diversity in the representations exposed to the LLM. Meanwhile, we aim to limit representation drift, as measured by CKA, so that the adaptor–LLM interface does not need to spend excessive effort adapting to each newly swapped encoder. To this end, we monitor changes in the representation distribution on a fixed validation set using CKA scores, as described in Section 3.4.

Given the current encoder checkpoint $\mathcal{E}_{\text{cur}}$ and a reference checkpoint $\mathcal{E}_{\text{ref}}$, we compute the CKA score between their representations and trigger an update once the score falls below a predefined threshold τ. Given two sets of representation $E^{(a)}, E^{(b)} \in \mathbb{R}^{L \times d_e}$ extracted from the same input by two encoders $\mathcal{E}_{\text{cur}}$ and $\mathcal{E}_{\text{ref}}$, CKA is defined as

$$\text{CKA}(E^{(a)}, E^{(b)}) = \frac{\langle \tilde{K}^{(a)}, \tilde{K}^{(b)} \rangle_F}{\sqrt{\langle \tilde{K}^{(a)}, \tilde{K}^{(a)} \rangle_F \cdot \langle \tilde{K}^{(b)}, \tilde{K}^{(b)} \rangle_F}}, \tag{8}$$

where $\tilde{K}^{(a)}$ and $\tilde{K}^{(b)}$ are centered Gram matrices computed as $\tilde{K}^{(x)} = \tilde{H} E^{(x)} E^{(x)\top} \tilde{H}$. The centering matrix is defined as $\tilde{H} = I_L - \frac{1}{L} J_L$, where I_L is the identity matrix and J_L is the all-ones matrix. CKA measures the geometric similarity of representation spaces, invariant to orthogonal transformations and isotropic scaling.

Iterative schedule and hot-swapping. Whenever an update is triggered, the current pretraining encoder $\mathcal{E}_{\text{cur}}$ is used to simultaneously update both (i) the frozen encoder $\mathcal{E}^{\text{SFT}}$ in the IA-SFT pipeline, and (ii) the reference checkpoint $\mathcal{E}_{\text{ref}}$ used for subsequent CKA monitoring during pretraining:

$$\mathcal{E}^{\text{SFT}} \leftarrow \mathcal{E}_{\text{cur}}, \qquad \mathcal{E}_{\text{ref}} \leftarrow \mathcal{E}_{\text{cur}}. \tag{9}$$

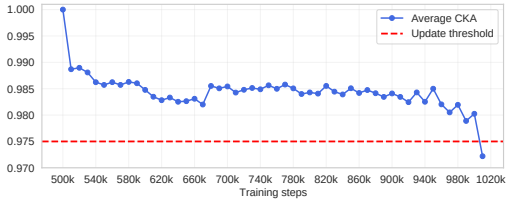
We begin monitoring CKA scores every 10k steps once pretraining reaches 500k steps, at which point the model starts to exhibit convergence trends. The encoder at 500k steps serves as the initial reference checkpoint $\mathcal{E}_{\text{ref}}$. Since pretraining focuses solely on the ASR task, the optimization direction remains largely consistent, causing CKA scores to generally decrease, with occasional rebounds throughout encoder evolution—this partly explains why direct encoder hot-swapping during IA-SFT works effectively without requiring realignment. Based on our experience, we set the CKA threshold $\tau = 0.975$, which we find to be a moderate choice: a higher threshold triggers more frequent updates, incurring unnecessary overhead; a lower threshold leads to less frequent updates, permitting greater representation drift between swaps, which may reduce the LLM's ability to capture robust patterns shared across encoder states.

When pretraining reaches 1.01M steps, the CKA score first drops below this threshold (see Figure 5a). At this point, the corresponding encoder is used to initialize the encoder in our encoder–adaptor–LLM model, and the alignment stage begins. After 1.3M alignment steps, we proceed to the IA-SFT stage, where the adaptor and LLM are jointly optimized. From then on, IA-SFT and pretraining are executed asynchronously in parallel. When pretraining reaches 1.32M steps, the CKA score again drops below 0.975 (see Figure 5b). We then directly update both $\mathcal{E}^{\text{SFT}}$ and $\mathcal{E}_{\text{ref}}$, and continue asynchronous training. Finally, when pretraining reaches the maximum step of 2M (see Figure 5c), we perform the last encoder hot-swapping for IA-SFT. As summarized in Table 5, the SFT-stage encoder is initialized once and updated twice throughout the process.

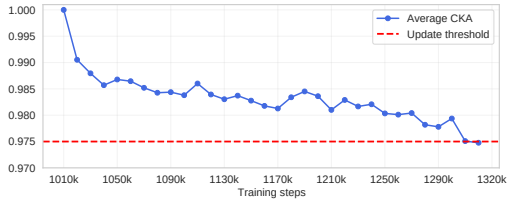
Why Encoder Updates without Realignment Work? First, in our design, pretraining and SFT target the same supervised ASR task, differing only in the supervision signals and loss functions, which helps maintain consistency in the overall optimization direction. Moreover, as shown in Figure 5, CKA scores between encoders remain consistently high throughout pretraining, suggesting substantial similarity in the dominant encoder subspace across checkpoints. In this regime, subsequent pretraining appears to mainly reduce acoustic redundancy and refine acoustic preferences, rather than reshape the global manifold. Whenever the updated encoder is applied to the SFT model, downstream components receive representations with lower entropy while remaining broadly distributionally consistent. The adaptor and LLM can therefore continue adapting to these inputs with limited additional alignment cost, which may help explain why explicit realignment was not necessary in our experiments.

In general, compared to the traditional pipeline, IA-SFT provides the following main advantages:

- **Maintaining functional decoupling.** In contrast to sequential SFT with trainable encoders, IA-SFT helps limit the tendency of the encoder to drift toward the LLM's semantic manifold. The



(a) CKA scores between encoders (0.50 M-1.01 M) and reference checkpoint (0.50 M)



(b) CKA scores between encoders (1.01 M-1.32 M) and reference checkpoint (1.01 M)



(c) CKA scores between encoders (1.32 M-2.00 M) and reference checkpoint (1.32 M)

Figure 5: Trajectory of CKA scores during pretraining. It reports the average CKA between the encoders and the corresponding reference checkpoint.

Table 5: Detailed training procedures, including the encoder update schedule, CKA thresholds, and stage-wise training steps. Here, the step counts for pretraining, alignment, and IA-SFT are measured independently in their respective training processes.

Pretraining Step	**Trigger**	**Action Details**
0.50 M	–	① **Pretraining begins.** The reference checkpoint is initialized at the 0.50 M pretraining step.
1.01 M	CKA < 0.975	② **Pretraining continues.** The encoder snapshot updates the reference checkpoint and initializes the encoder for post-training. The alignment stage then begins and runs for 1.3 M steps in total. Upon its completion, IA-SFT begins and runs for 1.0 M steps, while pretraining continues asynchronously.
1.32 M	CKA < 0.975	③ **Pretraining continues.** The encoder snapshot updates both the reference checkpoint and the encoder for IA-SFT, which then runs for another 1.0 M steps.
2.00 M (end)	–	④ **Pretraining ends.** The encoder snapshot updates the encoder for IA-SFT, which then runs for another 2.0 M steps.
–	–	⑤ **IA-SFT ends**, followed by joint-SFT, which runs for 2.00 M steps in total.

encoder can therefore continue optimizing primarily for acoustic uncertainty reduction, which is more consistent with preserving fine-grained acoustic discrimination and limiting semantic dependence.

- **Regularization from multiple encoder perspectives.** Beyond the reduction in entropy, the iteratively updated encoders exhibit distinct preference patterns for acoustic cues. IA-SFT exposes the LLM to representations with different signal-to-noise levels in a curriculum-like manner. This process can also act as a form of regularization, encouraging the LLM to focus on acoustic features that remain consistent across encoder states rather than overfitting to idiosyncratic biases of any single checkpoint.

- **Parallelization and efficiency.** Converting sequential pretraining and SFT into asynchronous parallel processes substantially reduces the total training time.
- **Flexibility and transferability.** Benefiting from functional decoupling, our encoder remains compatible with both the CTC head and the LLM decoder in our setup. This may improve deployment flexibility by enabling scalable configurations tailored to different computational constraints based on a shared encoder. It also creates a cleaner interface for iterating on individual components, which may reduce the cost of integrating newer LLM backbones in future system updates.

C Formalization of Encoder Representation Dynamics

This section provides a geometric perspective on encoder representation dynamics in LLM-based ASR systems with an encoder–adaptor–LLM architecture. Our goal is to characterize how different pretraining and post-training paradigms shape the spectral structure and functional composition of encoder representations, thereby inducing distinct operating regimes at the encoder–LLM interface.

C.1 Problem Setup and Spectral Decomposition

Let X be a random variable representing input speech features, and let $\mathcal{E}_\phi$ denote a speech encoder. For a realization x, the encoder produces

$$E = \mathcal{E}_\phi(x) = [e_1, \ldots, e_L]^\top \in \mathbb{R}^{L \times d_e}, \tag{10}$$

where L is the sequence length and d_e is the hidden dimension. Let $E' \in \mathbb{R}^{L_{valid} \times d_e}$ denote the valid-frame representation obtained from E after removing padded positions according to the sequence mask. Consider the singular value decomposition:

$$E' = U \Sigma V^\top, \tag{11}$$

where $\Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_d)$ with $d = \min(L_{valid}, d_e)$.

The singular values $\{\sigma_i\}$ characterize how variance is distributed across principal directions. By normalizing the singular values and treating them as a discrete spectrum, we can further quantify their concentration using an entropy measure, leading to the normalized spectral entropy (NSE). Moreover, a concentrated spectrum corresponds to a lower-entropy and more compact representation, whereas a flatter spectrum indicates higher residual uncertainty distributed across multiple directions. Since E is the encoder-side representation feeding into the adaptor–LLM stack, its spectral structure reflects how much uncertainty has already been reduced by the encoder before reaching the interface.

C.2 Functional Subspaces of Speech Representations

We view the encoder feature space as composed of multiple overlapping functional subspaces:

$$\mathbb{R}^{d_e} \supset \mathcal{S}_{\text{linguistic}} \oplus \mathcal{S}_{\text{paralinguistic}} \oplus \mathcal{S}_{\text{non-linguistic}}, \tag{12}$$

where

- $\mathcal{S}_{\text{linguistic}}$ captures transcription-relevant structure (e.g., phonetic and lexical information);
- $\mathcal{S}_{\text{paralinguistic}}$ includes speaker prosody and emotion;
- $\mathcal{S}_{\text{non-linguistic}}$ corresponds to environmental noise and nuisance variability.

These subspaces are not strictly orthogonal, but this decomposition provides a useful abstraction for analyzing how training redistributes variance. An effective ASR encoder should concentrate variance within $\mathcal{S}_{\text{linguistic}}$ while suppressing irrelevant variability, thereby forming a compact and acoustically grounded interface.

C.3 Accessible-Information Proxies under a Gaussian Approximation

Beyond spectral concentration, we also wish to characterize how much transcription-relevant information remains accessible from the encoder representation. Let u denote an utterance-level summary of the encoder representation, P a phonetic target variable, and S a semantic target variable. To quantify

their statistical dependence in a tractable form, we adopt a joint Gaussian approximation on (u, P, S) and use mutual-information-inspired log-determinant quantities as accessible-information proxies.

For a Gaussian random vector $\mathbf{r} \in \mathbb{R}^k$ with covariance $\Sigma_{\mathbf{r}}$, its differential entropy is given by

$$h(\mathbf{r}) = \frac{1}{2} \log\big((2\pi e)^k \det \Sigma_{\mathbf{r}}\big). \tag{13}$$

This shows that, under a Gaussian assumption, entropy is fully determined by the covariance structure through its log-determinant.

Proposition C.1 (Gaussian mutual information in log-det form)**.** *Let $A \in \mathbb{R}^{d_A}$ and $B \in \mathbb{R}^{d_B}$ be jointly Gaussian random variables with joint covariance*

$$\Sigma_{[A,B]} = \begin{bmatrix} \Sigma_{AA} & \Sigma_{AB} \\ \Sigma_{BA} & \Sigma_{BB} \end{bmatrix}. \tag{14}$$

Then their mutual information admits the closed-form expression

$$I(A; B) = \frac{1}{2} \log \frac{\det \Sigma_{AA} \, \det \Sigma_{BB}}{\det \Sigma_{[A,B]}}. \tag{15}$$

Proof. By definition,

$$I(A; B) = h(A) + h(B) - h(A, B). \tag{16}$$

Since (A, B) is jointly Gaussian, its marginals are also Gaussian. Therefore,

$$h(A) = \frac{1}{2} \log\big((2\pi e)^{d_A} \det \Sigma_{AA}\big), \tag{17}$$

$$h(B) = \frac{1}{2} \log\big((2\pi e)^{d_B} \det \Sigma_{BB}\big), \tag{18}$$

$$h(A, B) = \frac{1}{2} \log\big((2\pi e)^{d_A + d_B} \det \Sigma_{[A,B]}\big). \tag{19}$$

Substituting these expressions into the mutual information identity gives

$$\begin{aligned} I(A; B) = \frac{1}{2} \log\big((2\pi e)^{d_A} \det \Sigma_{AA}\big) + \frac{1}{2} \log\big((2\pi e)^{d_B} \det \Sigma_{BB}\big) \\ - \frac{1}{2} \log\big((2\pi e)^{d_A + d_B} \det \Sigma_{[A,B]}\big). \end{aligned} \tag{20}$$

Using logarithm rules, the constant terms cancel, yielding

$$I(A; B) = \frac{1}{2} \log \frac{\det \Sigma_{AA} \, \det \Sigma_{BB}}{\det \Sigma_{[A,B]}}. \tag{21}$$

□

Applying this result to (u, P) yields a Gaussian proxy for phonetic accessible information:

$$\mathrm{PAI}(E') \;\propto\; I(u; P) = \frac{1}{2} \log \frac{\det \Sigma_{uu} \, \det \Sigma_{PP}}{\det \Sigma_{[u,P]}}. \tag{22}$$

Proposition C.2 (Gaussian conditional mutual information in log-det form)**.** *Let $A \in \mathbb{R}^{d_A}$, $B \in \mathbb{R}^{d_B}$, and $C \in \mathbb{R}^{d_C}$ be jointly Gaussian. Then*

$$I(A; B \mid C) = \frac{1}{2} \log \frac{\det \Sigma_{AA|C} \, \det \Sigma_{BB|C}}{\det \Sigma_{[A,B]|C}}, \tag{23}$$

where the conditional covariance matrices are defined via the Schur complement, e.g.

$$\Sigma_{AA|C} = \Sigma_{AA} - \Sigma_{AC} \Sigma_{CC}^{-1} \Sigma_{CA}, \tag{24}$$

and analogously for $\Sigma_{BB|C}$ and $\Sigma_{[A,B]|C}$. Here, $\Sigma_{AA|C}$ captures the residual variability of A after removing the components that can be linearly explained by C.

Proof. By definition,

$$I(A;B \mid C) = h(A \mid C) + h(B \mid C) - h(A,B \mid C). \tag{25}$$

For jointly Gaussian variables, conditional distributions remain Gaussian, and their conditional entropies are determined by conditional covariance matrices:

$$h(A \mid C) = \frac{1}{2}\log\big((2\pi e)^{d_A} \det \Sigma_{AA|C}\big), \tag{26}$$

$$h(B \mid C) = \frac{1}{2}\log\big((2\pi e)^{d_B} \det \Sigma_{BB|C}\big), \tag{27}$$

$$h(A,B \mid C) = \frac{1}{2}\log\big((2\pi e)^{d_A+d_B} \det \Sigma_{[A,B]|C}\big). \tag{28}$$

Substituting into the definition of conditional mutual information yields

$$\begin{aligned} I(A;B \mid C) = \frac{1}{2}\log\big((2\pi e)^{d_A} \det \Sigma_{AA|C}\big) + \frac{1}{2}\log\big((2\pi e)^{d_B} \det \Sigma_{BB|C}\big) \\ - \frac{1}{2}\log\big((2\pi e)^{d_A+d_B} \det \Sigma_{[A,B]|C}\big). \end{aligned} \tag{29}$$

Again, the constant terms cancel, giving

$$I(A;B \mid C) = \frac{1}{2}\log \frac{\det \Sigma_{AA|C} \, \det \Sigma_{BB|C}}{\det \Sigma_{[A,B]|C}}. \tag{30}$$

□

By setting $(A,B,C) = (u,S,P)$, we obtain the conditional semantic accessible information:

$$\mathrm{CSAI}(E') \;\propto\; I(u;S \mid P) = \frac{1}{2}\log \frac{\det \Sigma_{uu|P} \, \det \Sigma_{SS|P}}{\det \Sigma_{[u,S]|P}}. \tag{31}$$

In practice, we estimate covariance matrices empirically with ridge regularization and convert the logarithm to base 2, leading to the regularized forms in Eqs. (5) and (6). These quantities should be interpreted as linear-Gaussian accessible-information proxies for relative comparison, rather than exact mutual information estimates. Unlike PAI and CSAI, NSE measures the entropy of the normalized singular-value spectrum, reflecting the spectral concentration and geometric compactness of encoder representations. Together, these metrics provide complementary perspectives on the evolution of encoder representations during training.

C.4 Encoder Pretraining and Spectral Bias

Different pretraining objectives induce distinct spectral biases, shaping how acoustic uncertainty is distributed across the encoder–LLM interface.

Character-level CTC. Character-level CTC enforces monotonic alignment between speech frames and transcription labels:

$$\mathcal{L}_{\text{CTC-char}} = -\log \sum_{\pi \in \mathcal{B}^{-1}(y^{\text{char}})} \prod_{i=1}^{L} p_\theta(\pi_i \mid e_i). \tag{32}$$

where y^{char} denotes the target character sequence, π is a frame-level alignment path, and $\mathcal{B}$ is the collapse operator. This objective encourages compact, transcription-aligned representations, but introduces early coupling to language-specific semantic units.

Phoneme-level CTC. Phoneme-level supervision mitigates premature semantic coupling:

$$\mathcal{L}_{\text{CTC-phoneme}} = -\log \sum_{\pi \in \mathcal{B}^{-1}(y^{\text{phoneme}})} \prod_{i=1}^{L} p_\theta(\pi_i \mid e_i). \tag{33}$$

where y^{phoneme} denotes the phoneme sequence. As phonemes are more directly tied to acoustic structure, this objective yields more language-agnostic and acoustically grounded representations, forming a more stable encoder–LLM interface.

AED-based pretraining. The AED objective formulates speech modeling as sequence-to-sequence prediction:

$$\mathcal{L}_{\text{AED}} = -\sum_{n=1}^{N} \log p_\theta(y_n \mid y_{<n}, E) \tag{34}$$

where the decoder attends to the encoder representation sequence. Compared with CTC, AED does not impose an explicit frame-level alignment constraint. Instead, it allows the encoder to retain broader contextual cues that can assist the autoregressive decoder. Moreover, the sequence-to-sequence formulation can be naturally extended beyond pure transcription to other speech-conditioned tasks. Consequently, although AED-pretrained encoders can support ASR, their representations tend to preserve a broader range of variability and exhibit less concentrated spectra than those optimized under strongly alignment-driven objectives. This broader representational capacity makes AED-pretrained encoders well suited for initializing LALMs, where the encoder is expected to support diverse audio–language tasks beyond transcription.

Hybrid supervised objectives. Some approaches combine alignment-driven and sequence-level objectives (e.g., CTC + AED):

$$\mathcal{L}_{\text{hybrid}} = \lambda \mathcal{L}_{\text{CTC}} + (1-\lambda)\mathcal{L}_{\text{AED}}. \tag{35}$$

Such hybrid training balances strong alignment constraints with sequence-level modeling, yielding intermediate spectral characteristics and uncertainty allocation.

Self-supervised pretraining (SSL). Self-supervised objectives such as Best-RQ [9] learn acoustic representations by predicting discrete pseudo-targets derived from the input signal:

$$\mathcal{L}_{\text{SSL}} = -\sum_{t \in \mathcal{M}} \log p_\theta(z_t \mid \tilde{E}), \tag{36}$$

where z_t denotes discrete targets obtained via a quantization process, $\tilde{E}$ represents masked or corrupted acoustic features, $\mathcal{M}$ denotes the set of masked positions, and the model predicts a categorical distribution over codebook entries at masked positions. In this framework, continuous speech is first mapped to discrete codebook indices, which serve as prediction targets, and the model is trained to infer these targets at masked positions from surrounding context. Unlike supervised objectives, SSL does not enforce alignment to linguistic units, resulting in higher-entropy representations and deferring more uncertainty to downstream modules such as the LLM.

C.5 Instruction-Based Post-training and Divergent Encoder Regimes

After encoder pretraining, both LLM-ASR and LALM systems are typically further optimized under instruction-conditioned language modeling objectives. Let $E = \{e_i\}_{i=1}^{L}$ denote the encoder representations, t the text instruction prompt, and $y = (y_1, \ldots, y_N)$ the target response sequence. A generic post-training objective can be written as

$$\mathcal{L}_{\text{inst}} = -\sum_{n=1}^{N} \log p_\theta(y_n \mid y_{<n}, E, t). \tag{37}$$

Once the encoder is connected to the adaptor–LLM stack under such supervision, the downstream language modeling objective further reshapes the representation geometry at the encoder–LLM interface. In LLM-ASR, the encoder is usually initialized from an ASR-oriented model trained with transcription objectives such as CTC or AED, so its representations are already biased toward transcription-relevant acoustic structure and may provide a relatively compact interface before LLM integration. Moreover, the subsequent instruction-based supervision is still centered on transcription, which further encourages the encoder–LLM interface to remain specialized for ASR. Importantly, when a well-formed interface has been established prior to full end-to-end coupling, subsequent joint optimization primarily induces local refinements on the existing representation manifold, rather than causing large distribution shifts toward text-dominated regimes. This allows the model to improve cross-module alignment while preserving the capability boundary between acoustic grounding and semantic modeling.

By contrast, LALMs are typically trained to support a broader range of audio–language tasks beyond transcription. Their encoders therefore tend to preserve a wider range of information and exhibit a flatter, higher-entropy representation space. Moreover, the diverse instructions further encourage representations to remain compatible with more general audio understanding. Consequently, although both LLM-ASR and LALMs may adopt the same encoder–adaptor–LLM architecture, differences in both encoder initialization and post-training supervision lead them to evolve under distinct representational regimes.

D Entropy Allocation and Functional Decoupling in LLM-ASR

In this section, we provide an information-theoretic perspective on entropy allocation in LLM-based ASR. Our goal is to clarify how a well-formed encoder–LLM interface enables a capability-aligned division of uncertainty reduction between acoustic grounding and semantic disambiguation.

D.1 Entropy Decomposition at the Encoder–LLM Interface

Let X and Y denote the input speech and target transcription, respectively. The speech encoder and adaptor define a deterministic transformation chain

$$X \xrightarrow{\mathcal{E}_\phi} E \xrightarrow{\mathcal{A}_\psi} Z, \tag{38}$$

where E is the encoder representation and Z is the projected speech embedding consumed by the LLM. At the interface level, the uncertainty of the target transcription can be decomposed as

$$H(Y) = I(Y;Z) + H(Y \mid Z), \tag{39}$$

where $I(Y;Z)$ measures the amount of task-relevant information exposed through the interface, and $H(Y \mid Z)$ represents the residual uncertainty to be resolved by the LLM.

Following the information-theoretic perspective of the Information Bottleneck (IB) principle and its extensions to deep neural networks [44, 46, 47], the data processing inequality implies that

$$I(Y;Z) \le I(Y;E) \le I(Y;X). \tag{40}$$

Accordingly, the task-relevant information loss across the encoder–adaptor boundary can be expressed as

$$\Delta^{Y}_{\text{adapt}} = I(Y;E) - I(Y;Z) = I(Y;E \mid Z) \ge 0. \tag{41}$$

A boundary is task-lossless when $\Delta^{Y}_{\text{adapt}} = 0$, whereas $\Delta^{Y}_{\text{adapt}} > 0$ indicates the loss of task-relevant information.

This decomposition naturally reflects a division of labor across modules. A more informative and structured interface increases $I(Y;Z)$ and reduces the burden on the LLM, whereas a weaker interface shifts more uncertainty to downstream language modeling. However, preserving task-relevant information alone does not necessarily ensure an appropriate allocation of modeling responsibilities across heterogeneous modules. Importantly, the effectiveness of the interface depends not only on the quantity of retained information, but also on whether that information is aligned with the functional roles of each module.

D.2 Capability Boundary and Functional Decoupling

In LLM-based ASR, the encoder and the LLM exhibit complementary inductive biases. The encoder is well suited for resolving local acoustic ambiguity, such as phonetic distinctions and temporal structure, while the LLM is more effective at resolving higher-level ambiguity through linguistic priors and contextual reasoning. Accordingly, a well-formed encoder–LLM interface should satisfy two properties. First, it should provide a sufficiently compact representation to avoid unnecessary burden on the LLM. Second, the retained structure should remain acoustically grounded, preserving evidence derived from the speech signal rather than replacing it with text-correlated shortcuts. Under this perspective, the encoder and LLM operate under functional decoupling: the encoder primarily reduces acoustic uncertainty, while the LLM resolves the remaining semantic ambiguity conditioned on the interface representation.

D.3 Hallucination as Misallocated Uncertainty Reduction

From this viewpoint, hallucination in LLM-based ASR can be interpreted as a consequence of misallocated uncertainty reduction across the encoder–LLM interface. Two representative failure modes are particularly relevant.

Semantic-contaminated encoder representations. One failure mode arises when joint optimization progressively aligns encoder representations with text-correlated regularities. In this regime, part of the uncertainty reduction is achieved through patterns that are not strictly grounded in acoustic evidence. As a result, the interface may become more predictive of the transcription while losing robustness to acoustic variation, increasing the likelihood of fluent but weakly grounded outputs under ambiguous or degraded conditions.

LLM-dominant uncertainty reduction. Another failure mode occurs when the encoder provides a weak or insufficiently structured interface. In this case, a larger portion of uncertainty is deferred to the LLM, making the decoding process more dependent on language priors. This behavior can be further understood through the Bayesian factorization

$$p(Y \mid Z) \propto p(Z \mid Y)\, p(Y). \tag{42}$$

When Z provides limited discrimination among candidate transcriptions, the likelihood term $p(Z \mid Y)$ becomes less informative, and the posterior is increasingly shaped by the prior $p(Y)$. Consequently, the model may generate outputs that are linguistically plausible but not fully supported by the speech signal.

D.4 Rationale of the Decoupled Training Paradigm

The above analysis motivates a training strategy that aligns uncertainty allocation with module capabilities. Phoneme-level pretraining encourages the encoder to form a compact and acoustically grounded interface by emphasizing pronunciation-level structure. Subsequent adaptation stages allow the adaptor and LLM to operate on this interface before full end-to-end coupling. After the interface has been sufficiently established, joint optimization can be introduced to refine the overall system. At this stage, interactions between modules primarily induce local adjustments in the representation manifold, improving alignment at the encoder–LLM interface without substantially altering the underlying representation structure.

From this perspective, hallucination can be understood as a consequence of how uncertainty reduction is distributed across modules. Maintaining an interface that is both compact and acoustically grounded preserves the intended division of labor between acoustic grounding and downstream language modeling. As a result, the encoder can absorb a larger portion of acoustically grounded uncertainty reduction, while leaving higher-level semantic disambiguation to the LLM. This capability-aligned allocation not only mitigates hallucination by reducing reliance on text-side priors, but also improves parameter efficiency by alleviating the burden on the LLM, enabling strong performance with a smaller model scale.

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 three contributions listed in the abstract and Section 1—entropy-allocation metrics (Section 3.2), the capability-boundary-aware training paradigm (Section 3.4), and competitive 2.3B-parameter results with low hallucination (Section 4)—are all substantiated in the paper. The scope is explicitly limited to ASR-oriented LLMs.

   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 5 and Appendix A discuss three limitations—scalability to larger backbones, absence of RL post-training analysis, and the linear-Gaussian approximation underlying PAI/CSAI—each paired with a concrete future-work direction.

   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: [Yes]

Justification: The two propositions in Appendix C are stated under an explicit joint Gaussian assumption and accompanied by complete proofs. They underpin the PAI/CSAI proxies in Eqs. (5)–(6), which are clearly characterized as relative proxies rather than exact mutual-information estimates.

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: We provide the model architecture (Appendix B.2), training data with per-corpus statistics ($\sim$95% public; Appendix B.1), stage-wise hyperparameters (Appendix B.3), the full IA-SFT procedure with CKA thresholds (Appendix B.4), and evaluation/decoding settings (Section 4). These details suffice for reproducing the main results.

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: Training code and checkpoints are not released due to industrial constraints. However, $\sim$95% of the training data are public (Appendix B.1), and all architecture, hyperparameter, and IA-SFT scheduling details are documented in Appendices B.2–B.4 as a complete reimplementation recipe. All evaluation benchmarks and the WeTextProcessing toolkit are publicly available.

   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 specifies evaluation benchmarks, metrics, text normalization, and decoding settings. Appendix B.3 reports stage-wise hyperparameters (batch size, learning-rate schedule, optimizer, precision, hardware). Appendix B.4 details the IA-SFT procedure including the CKA threshold and its motivation.

   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: Following standard practice in large-scale ASR, we report deterministic CER/WER without error bars. Beam-search decoding is deterministic, so repeated runs on the same checkpoint produce identical outputs. The large evaluation sets (typically thousands of utterances each) and consistent gains across diverse benchmarks provide an implicit robustness check.

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: [Yes]

   Justification: Appendix B.3 reports the GPU type (NVIDIA A100 80GB), parallelism strategy (DeepSpeed ZeRO-2), precision (bfloat16), and FlashAttention-2. Stage-wise step counts in Appendix B.4 (Table 5) allow estimation of the overall training cost.

   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: We have reviewed and confirm full compliance with the NeurIPS Code of Ethics. All datasets are either publicly licensed or in-house data collected with proper consent. No human subjects research is involved beyond standard speech corpus usage.

   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: Positive impacts include lower deployment cost for speech interfaces and improved accessibility; our hallucination-mitigation design further enhances reliability in safety-critical scenarios. Potential negative impacts, shared with ASR technology in general, include uneven recognition quality across dialects and demographics, and possible misuse for unauthorized transcription.

    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: No model checkpoints or scraped datasets are released. The work proposes a training methodology for ASR, which poses low risk of misuse; no specific safeguards are required.

    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: [Yes]

    Justification: All public datasets and pretrained models are properly cited (Appendices B.1–B.2), including the FireRedASR-AED encoder, Qwen3-1.7B, Qwen3-Embedding-8B, and all speech corpora. Each asset is used in compliance with its original license; in-house data is collected under applicable consent terms.

    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: No new datasets, models, or other assets are released.

    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: No crowdsourcing or human subjects research is involved. All data come from existing speech corpora or in-house data collected under standard consent terms.

    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: No human subjects research is involved; no IRB approval is required.

    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: [N/A]

    Justification: LLMs are not used as an original or non-standard component of our core methodology. Qwen3-1.7B serves as the ASR decoder and Qwen3-Embedding-8B is used for CSAI computation, both standard publicly available models documented in Appendix B.2 and Section 3.2. A general-purpose LLM was additionally used for language editing during manuscript preparation; all technical content was independently verified by the authors.

    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.