Do We Need Distinct Representations for Every Speech Token? Unveiling and Exploiting Redundancy in Large Speech Language Models

Bajian Xiang* **Tingwei Guo** **Xuan Chen*** **Yang Han**
Beike Inc., Beijing, China
{xiangbajian001,guotingwei002,chenxuan046,hanyang030}@ke.com
https://xchen-zero.github.io/speech-token-redundancy/

Abstract

Large Speech Language Models (LSLMs) typically operate at high token rates (tokens/s) to ensure acoustic fidelity, yet this results in sequence lengths that far exceed the underlying semantic content, incurring prohibitive inference costs. In this paper, we empirically revisit the necessity of such granular token-level processing. Through layer-wise oracle interventions, we unveil a structured redundancy hierarchy: while shallow layers encode essential acoustic details, deep layers exhibit extreme redundancy, allowing for aggressive compression. Motivated by these findings, we introduce *Affinity Pooling*, a training-free, similarity-based token merging mechanism. By strategically applying this method at both input and deep layers, we effectively compress speech representations without compromising semantic information. Extensive evaluations across three tasks demonstrate that our approach reduces prefilling FLOPs by 27.48% while maintaining competitive accuracy. Practical deployment further confirms significant efficiency gains, yielding up to $\sim$1.7$\times$ memory savings and $\sim$1.1$\times$ faster time-to-first-token on long utterances. Our results challenge the necessity of fully distinct token representations, providing new perspectives on LSLM efficiency.

1 Introduction

Large Speech Language Models (LSLMs) typically process audio at high token rates to ensure acoustic fidelity (Cui et al., 2025; Bu et al., 2024). However, speech naturally exhibits highly non-uniform redundancy over time, resulting in a token sequence that grows far faster than the underlying semantic content (Wang et al., 2025a; Zheng et al., 2025). This forces the language backbone to process many redundant tokens, incurring substantial and often unnecessary computation.

Similar redundancy has also been observed in Vision-Language Models (VLMs), motivating a line of work on token compression to reduce sequence length while preserving task-relevant semantics (Wen et al., 2025; Shao et al., 2025b). In contrast, compression for LSLMs remains relatively underexplored: existing speech-centric methods largely borrow VLM techniques by operating on spectrograms (Behera et al., 2024; Lee and Lee, 2025) or apply compression in specialized architectures (Li et al., 2023). More importantly, it remains unclear how redundancy is distributed across layers in LSLMs, hindering principled choices of where and how aggressively to apply compression.

To bridge this gap, we investigate internal redundancy of LSLMs through layer-wise oracle interventions. By dropping or merging tokens based on supervised linguistic boundaries, we unveil a clear hierarchy: shallow layers encode fine-grained details while **deep layers exhibit extreme redundancy, allowing significant token reduction with negligible performance degradation**. We further analyze the layer-wise dynamics of speech token cosine similarity to explain this behavior.

Building on these findings, we introduce *Affinity Pooling*, an unsupervised similarity-based compression algorithm. We first validate it as an intervention probe, demonstrating that **intrinsic similarity captures essential information more effectively than supervised alignment**. We then formalize *Affinity Pooling* and its variant, *Dual Affinity Pooling* (DAP), as training-free mechanisms applied during inference. Extensive evaluation across three semantic speech tasks confirms that DAP reduces FLOPs by 27.48% while preserving or improving accuracy. Practical measurements further show consistent deployment gains of up to $\sim$1.7$\times$ memory saving and $\sim$1.1$\times$ faster time-to-first-token (TTFT) on long utterances.

Our contributions are three-fold: (1) We are the first to anatomize layer-wise redundancy in

*Corresponding authors.

Findings of the Association for Computational Linguistics: ACL 2026, pages 15069–15087

LSLMs via controlled interventions, offering an interpretable view of their inner workings; (2) We propose *Affinity Pooling*, a training-free, similarity-driven token compression algorithm whose design is explicitly grounded in the above analysis; (3) We validate the approach across multiple models and semantic speech tasks, and conduct targeted sensitivity analyses of key hyperparameters.

2 Related Work

2.1 Large Speech Language Models

LSLMs have emerged as a prominent paradigm for processing spoken inputs, typically consisting of a speech encoder, an alignment module, and a Large Language Model (LLM) backbone. The speech encoder converts raw audio into token sequences, which vary in design philosophy: continuous encoders such as Qwen2-Audio (Chu et al., 2024) directly extract acoustic features, discrete encoders such as GLM-4-Voice (Zeng et al., 2024) and Baichuan-Audio (Li et al., 2025) quantize speech into symbolic tokens, and hybrid approaches like Kimi-Audio (KimiTeam et al., 2025) combine both representations to leverage their complementary strengths.

Regardless of the architectural choice, these models often operate at high tokenization rates, typically ranging from 12.5 to 25 tokens per second of audio (Ji et al., 2024), producing sequences substantially longer than their textual counterparts, inherently suggesting a high degree of redundancy.

2.2 Token Compression in Multimodal LLMs

To mitigate the computational overhead of high-resolution inputs in VLMs, the vision-language community has developed diverse compression strategies, ranging from attention-based pruning (Yang et al., 2024; Shao et al., 2025a) to similarity-driven merging (Bolya et al., 2023; Tao et al., 2025). Recent advancements have transcended heuristic methods, prioritizing interpretability to rigorously identify which representations are suitable to pruning or merging (Fu et al., 2025).

Token compression for LSLMs remains underexplored. Recent attempts like SpeechPrune (Lin et al., 2025) and TimeAudio (Wang et al., 2025b) utilize attention-guided pruning or learnable aggregation modules before the LLM input. These initial explorations, while promising, operate primarily through empirical design without grounding in speech-specific representational analysis.

3 Anatomy of Redundancy Through Oracle Interventions

This section investigates the redundancy of audio token representations within LSLMs. We use word-level timestamps as an Oracle to align the audio token stream with its corresponding linguistic units, and then compress the token stream within specific semantic windows through dropping or merging. By assessing the recoverability of the original content from these reduced sequences via an ASR task, we empirically quantify the structural redundancy inherent in the audio representation.

3.1 Methodology

Intervention Framework. We formally define the intervention process within the latent space of the LSLM, as illustrated in Figure 1. Let $\mathbf{H}^{(l)} = [\mathbf{H}_a^{(l)}; \mathbf{H}_t^{(l)}] \in \mathbb{R}^{(T_a+T_t)\times d}$ denote the concatenated hidden states at layer l, where T_a and T_t represent the sequence lengths of audio and text, respectively, and d is the hidden dimension. Our objective is to apply a compression operator Φ solely to the audio component, yielding a reduced representation $\tilde{\mathbf{H}}_a^{(l)} = \Phi(\mathbf{H}_a^{(l)})$, while leaving the text component $\mathbf{H}_t^{(l)}$ intact. The subsequent layer $l{+}1$ then processes the modified sequence $[\tilde{\mathbf{H}}_a^{(l)}; \mathbf{H}_t^{(l)}]$.

Compression Strategies. To quantify redundancy, we partition the audio sequence into semantic units aligned with word-level timestamps. We then compress each unit to a fixed budget of R tokens via three operators: (1) *Random Drop*, which stochastically samples R tokens; (2) *Uniform Drop*, which samples deterministically at regular strides to preserve temporal structure; and (3) *Uniform Merge*, which divides the unit into R equal-sized bins and performs mean-pooling within each bin.

Experimental Setup. We utilize Qwen2-Audio (32 layers, 25 tokens/s) and Kimi-Audio (28 layers, 12.5 tokens/s) on the Librispeech-test-clean test set. We evaluate the semantic recoverability of the compressed representations via Word Error Rate (WER) on an ASR task. For generation, we employ greedy decoding with a maximum token limit of 256. Word alignments are obtained offline via the Montreal Forced Aligner (MFA). To trace the layer-wise evolution of redundancy, we apply interventions **to single layers individually** at intervals of five. The retention budget R is scaled according to frame rates: $R \in \{2, 4, 8, 16\}$ for Qwen2-Audio and $R \in \{1, 2, 4, 8\}$ for Kimi-Audio.

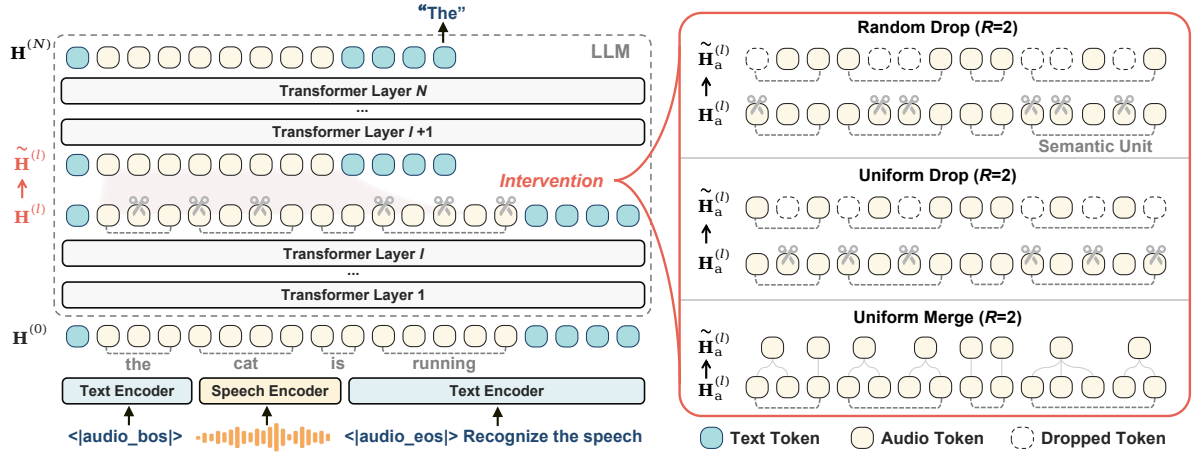


Figure 1: **Framework of oracle intervention experiments.** We align audio tokens to semantic units and apply compression operators **to a single layer at a time** to investigate redundancy.

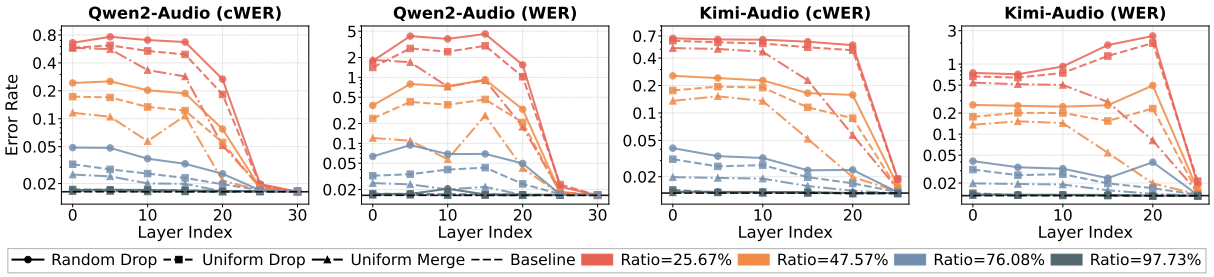


Figure 2: **Layer-wise oracle interventions on Qwen2-Audio and Kimi-Audio**. For each model, we report clamped WER (cWER) and standard WER plotted on **log-scale**. Colors represent different audio token retention rates.

3.2 Layer-wise Redundancy Evolution

Figure 2 illustrates the intervention results, where curves of different colors correspond to varying retention budgets R (converted here into audio token retention ratios). To isolate semantic loss from degenerate decoding behaviors, we report standard WER alongside clamped WER (cWER). Formally, for a dataset with samples indexed by i, we define cWER $= \sum_i \min(E_i, N_i) \,/\, \sum_i N_i$, where E_i and N_i are the edit distance and reference length for the i-th sample. Full numerical results are provided in Appendix A.1. We distill our observations into three primary findings:

(1) **Progressive Growth of Redundancy.** As evidenced by the cWER profiles, the model's sensitivity to token removal decreases monotonically with depth. While shallow layers require high retention budgets to preserve acoustic fidelity, deep layers ($l \geq 25$) exhibit extreme redundancy; performance converges to the baseline even when retaining as few as 25.67% of the original tokens. This observation suggests that deeper layers may harbor a significantly higher degree of redundancy.

(2) **Acoustic-to-Semantic Transition.** A large gap between WER and cWER appears in the middle layers ($l \in [5, 15]$ for Qwen2-Audio; $l \in [15, 20]$ for Kimi-Audio). This gap is primarily caused by repetition loops, as shown in the decoding examples in Table 9 and Table 10 (Appendix D.1). Besides loops, we also observe unstable behaviors like cross-lingual hallucinations and semantic drift. For instance, the model paraphrases "ghost" to "spirit" or "vesture" to "veil." This suggests that the middle layers are in a critical transitional state. **The model has started to abstract high-level semantics from acoustic features** but has not yet fully aligned them with exact lexical tokens.

(3) **Structural Nature of Redundancy.** The hierarchy of intervention strategies highlights that speech redundancy is not random. Across all layers, *Uniform Drop* consistently outperforms *Random Drop*, suggesting that redundancy possesses a temporal structure that benefits from regular sampling. Furthermore, *Uniform Merge* yields the lowest error rates consistently across both models and varying retention budgets. This indicates that tokens deemed "redundant" likely still contain distributed information; consequently, aggregating these features via averaging appears to preserve semantic

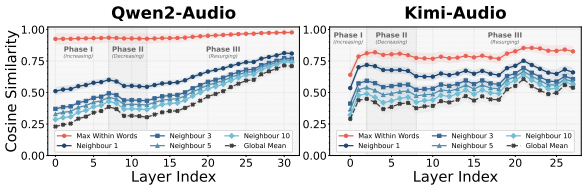


Figure 3: Layer-wise cosine similarity dynamics for Qwen2-Audio and Kimi-Audio.

cues more effectively than simple excision.

3.3 Deep Dive: Feature Dynamics

To understand the intrinsic mechanisms driving the observed redundancy, we analyze the layer-wise evolution of audio representations on the Librispeech-test-clean set. As shown in Figure 3, we track three cosine similarity metrics: (1) Neighbor Similarity, the average similarity between a token and its k-nearest neighbors ($k \in \{1, 3, 5, 10\}$); (2) *Global Mean*, the average similarity across the entire sequence; (3) *Max Within Words*, the maximum adjacent similarity within word boundaries.

The unsupervised metrics display a clear rise–fall–rise trajectory. Phase II indicates that local relationships between nearby tokens become less consistent, suggesting a period of representation reorganization. This aligns with the high intervention sensitivity noted in Section 3.2. We interpret this instability as a symptom of the critical transition from dense acoustic features to the highly redundant semantic abstractions emerging in deeper layers. Notably, Kimi-Audio exhibits a metric drop in Phase III, likely due to its unique design reusing layer 21 for acoustic decoding.

Max Within Words shows an overall upward trend and typically peaks in deep layers; Qwen2-Audio also displays a notably high *Global Mean* near the top. These metrics indicate that deep layers map tokens within the same linguistic unit to highly similar vectors, creating significant representational redundancy, which explains why aggressive merging becomes effective at depth.

4 Similarity–Driven Interventions

While the previous section identified **when** to compress by pinpointing redundant layers, this section investigates **where** to compress by exploring the specific token structures governing this redundancy. We introduce *Affinity Pooling*, a method that aggregates tokens based on feature cosine similarity. We first detail the algorithm, evaluate its efficiency against the oracle baseline, and then interpret the semantic granularity of the merged tokens.

Algorithm 1 Pseudocode for *Affinity Pooling*

Input: Audio sequence $\mathbf{H}_a = [h_1, \ldots, h_{T_a}] \in \mathbb{R}^{T_a \times d}$, lookback window size ω, similarity threshold τ
Output: Merged sequence $\tilde{\mathbf{H}}_a$
$\tilde{\mathbf{H}}_a \leftarrow \emptyset$
$\mathcal{G}_{curr} \leftarrow [h_1]$ ▷ initialize current group
for $t = 2$ to T_a **do**
  $k \leftarrow \min(|\mathcal{G}_{curr}|, \omega)$
  $\mathbf{K} \leftarrow$ last k tokens in $\mathcal{G}_{curr}$
  $s_{\max} \leftarrow \max_{k_i \in \mathbf{K}} \cos(h_t, k_i)$
  if $s_{\max} \geq \tau$ **then**
    Append h_t to $\mathcal{G}_{curr}$
  else
    $\bar{h} \leftarrow \frac{1}{|\mathcal{G}_{curr}|} \sum_{h \in \mathcal{G}_{curr}} h$ ▷ mean-pool group
    Append $\bar{h}$ to $\tilde{\mathbf{H}}_a$
    $\mathcal{G}_{curr} \leftarrow [h_t]$
  end if
end for
$\bar{h} \leftarrow \frac{1}{|\mathcal{G}_{curr}|} \sum_{h \in \mathcal{G}_{curr}} h$
Append $\bar{h}$ to $\tilde{\mathbf{H}}_a$
return $\tilde{\mathbf{H}}_a$

4.1 *Affinity Pooling*

Building on the findings in Section 3, we propose *Affinity Pooling* (Algorithm 1), where the term *affinity* captures the semantic closeness between token representations as measured by cosine similarity.

Our design diverges from existing paradigms in the following aspects: Unlike vision-centric global matching (Bolya et al., 2023), our design strictly adheres to the intrinsic **temporal locality** of speech. Furthermore, distinct from heuristic adaptations in prior speech works (Li et al., 2023), our use of latent similarity is explicitly grounded in the structural redundancy revealed in Section 3.3. We also address the limitations of standard adjacent merging by introducing a **lookback window** ω. While strict adjacency ($\omega = 1$) is susceptible to high-frequency acoustic jitter, our windowed approach ($\omega > 1$) bridges local fluctuations, preserving semantic continuity without compromising the similarity threshold τ.

Operationally, a token h_t is merged into the active group if its cosine similarity with any of the most recent ω tokens exceeds τ; otherwise, the current group is aggregated via mean-pooling.

4.2 Layer-wise Dynamics

We evaluate *Affinity Pooling* on Qwen2-Audio and Kimi-Audio separately to individual layers under the setup in Section 3.1. We fix $\omega = 3$ here as an empirical default, focusing on probing layer-wise redundancy, while deferring sensitivity analysis to Section 5.4. Figure 4 (data in Appendix A.2) illustrates the results, revealing three key phenomena:

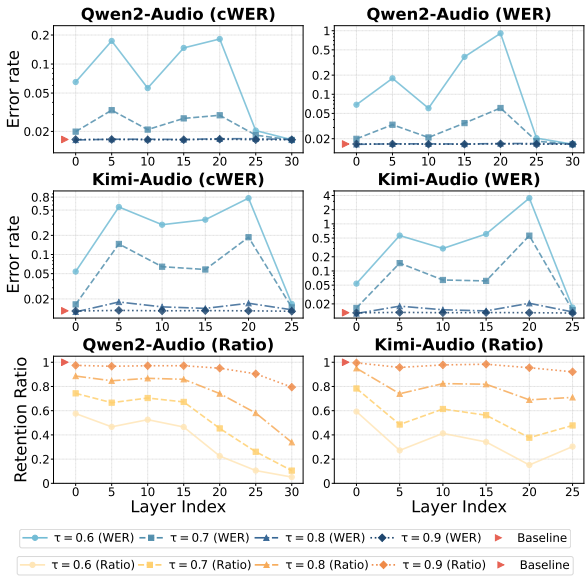


Figure 4: Layer-wise dynamics of *Affinity Pooling* on Qwen2-Audio and Kimi-Audio. We report WER, cWER on log-scale, and retention ratios with $\omega = 3$ across varying thresholds $\tau \in \{0.6, 0.7, 0.8, 0.9\}$.

(1) **Non-monotonic feature stability.** As illustrated in Figure 4, *Affinity Pooling* exhibits a bimodal error profile across both WER and cWER metrics. While representations at the input ($l = 0$) and deep layers ($l \geq 25$) remain robust, intermediate depths degrade significantly, showing distinct error spikes around $l = 5$ and $l = 20$ under aggressive thresholds ($\tau < 0.8$). This sensitivity mirrors the trend observed in the Oracle baseline as discussed in Section 3.2, confirming that intermediate layers undergo critical feature reorganization and are sensitive and best left uncompressed.

(2) **Substantial compression at deep layers.** Retention ratios consistently decrease as depth increases (Figure 4, bottom row). Notably, applying *Affinity Pooling* to Qwen2-Audio at $l = 30$ ($\tau = 0.6$) compresses the sequence to 5.18% of its original length while achieving a WER of 1.64%, slightly outperforming the uncompressed baseline of 1.65%. This result indicates that **deep layers possess significantly higher compressibility than previously observed** in Section 3.2, suggesting that *Affinity Pooling* effectively uncovers the latent redundancy within these representations.

(3) **Superiority over supervised alignment.** Our unsupervised approach outperforms the supervised Oracle at the input level. At $l = 0$ ($\tau = 0.7$), we achieve a lower WER of 1.99% with 74.32% retention, compared to the Oracle's 2.50% WER at 76.08% retention. This suggests that intrinsic similarity captures essential information more effectively than the rigid linguistic boundaries.

4.3 Semantic Granularity of Merged Tokens

Why does a simple cosine similarity-based merging strategy achieve such high compression rates while preserving model performance? To answer this, we analyze the emergent token groups formed during the merging process. Here, we present a representative sample to intuitively illustrate this behavior, while more examples are detailed in Appendix D.2.

Figure 5 illustrates the layer-wise evolution of token aggregation. We observe a transition from fragmented, acoustic-level groupings in shallow layers ($l \leq 5$) to broad semantic abstractions in deep layers ($l \geq 25$). For instance, Kimi-Audio forms continuous token blocks, whereas Qwen2-Audio consolidates larger multi-word blocks, often merging sequences of 4–5 words or more into a single group. Crucially, this structural aggregation preserves fidelity: **both models achieve a WER of 0 on this utterance across all layers visualized in the figure**. These patterns align with recent findings that LLM embeddings operate far below their theoretical information capacity—where a single vector could encode over 1,500 tokens (Kuratov et al., 2025). Our method leverages this by merging these adjacent similar tokens, effectively densifying information without exceeding the vector's capacity.

5 *Affinity Pooling* for Efficient LSLMs

In this section, we apply *Affinity Pooling* as a training-free compression mechanism for LSLMs. We first evaluate the method across multiple downstream tasks, verifying that it maintains high performance despite significant token reduction. To demonstrate practical utility, we report improvements in inference speed and memory consumption on standard hardware. We then benchmark our approach against fixed-budget baselines to highlight its superiority over naive compression methods. Finally, we analyze the impact of key hyperparameters to provide optimal configuration guidelines.

5.1 Performance on Downstream Tasks

We assess the efficacy of *Affinity Pooling* across three diverse speech tasks: Automatic Speech Recognition (ASR), Speech Question Answering (QA), and Speech Translation (ST). Our primary objective is to determine whether the proposed method can reduce computational overhead without compromising task performance.

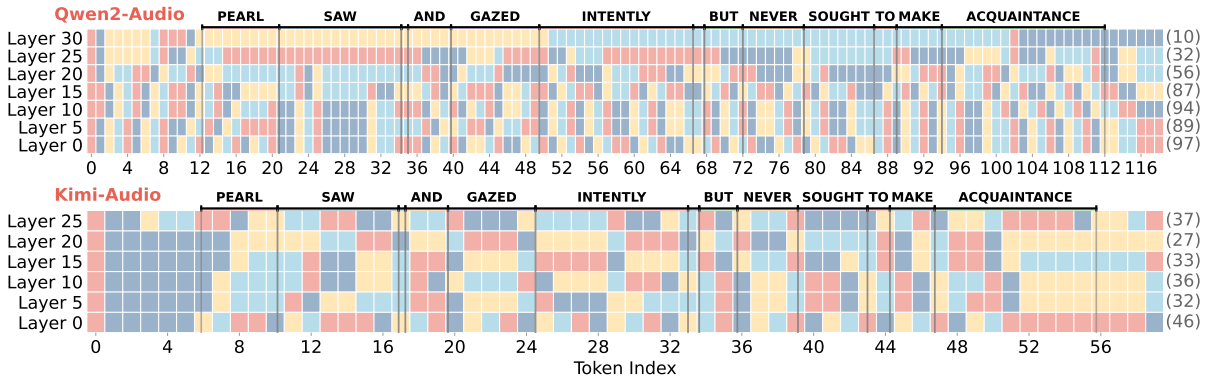


Figure 5: **Visualization of *Affinity Pooling* (τ=0.7, $\omega = 3$) on Qwen2-Audio (top) and Kimi-Audio (bottom).** Colors denote merged token groups, and vertical lines mark word boundaries. The right axis indicates the total number of tokens after compression. **Both models maintain a WER of 0 across all tested layers.**

Table 1: **Main results and ablation studies on Qwen2-Audio.** FRR: Final Retention Ratio. Bold indicates improvement over Vanilla, while underlining denotes the best performance within each setting.

Method	Scope		Efficiency (↓)			ASR (WER ↓)				QA (Acc ↑)				ST (BLEU ↑)		
	l_{in}	l_{deep}	FRR	Pre. GFLOPs	FLOPs Ratio	KES	LSC	LSO	*Avg.*	OBQA	SDQA	TrQA	*Avg.*	en2zh	zh2en	*Avg.*
Vanilla	-	-	100.0	780.94	100.0	3.28	1.65	3.88	2.94	42.64	27.31	21.29	30.41	42.80	23.02	32.91
Setting A: Aggressive (τ_{in}=0.80, τ_{deep}=0.70)																
AP_{in}	✓	-	78.64	612.93	78.49	3.39	**1.64**	**3.80**	**2.94**	**43.74**	26.29	20.90	30.31	42.35	22.69	32.52
AP_{deep}	-	✓	14.30	718.12	91.96	3.31	**1.65**	**3.84**	**2.93**	**42.86**	**27.49**	21.19	**30.51**	42.76	**23.02**	32.89
DAP	✓	✓	14.91	566.30	72.52	3.44	**1.63**	**3.79**	2.95	42.20	**29.66**	20.61	**30.82**	42.29	22.78	32.54
Setting B: Conservative (τ_{in}=0.90, τ_{deep}=0.80)																
AP_{in}	✓	-	93.56	730.28	93.51	**3.26**	**1.65**	**3.81**	**2.91**	**43.52**	**27.67**	**21.58**	**30.92**	**42.84**	**23.06**	**32.95**
AP_{deep}	-	✓	33.29	731.96	93.73	3.31	1.66	**3.83**	**2.93**	**44.62**	26.94	21.09	**30.88**	**42.80**	23.01	**32.91**
DAP	✓	✓	33.76	686.39	87.89	3.29	**1.64**	**3.77**	**2.90**	**42.64**	**27.85**	20.90	**30.46**	**42.86**	22.94	32.90

Experimental Setup. We utilize Qwen2-Audio as the primary testbed. We compare the uncompressed baseline (Vanilla) against three variants of our method: *Affinity Pooling* applied only at the input (AP_{in}), only at a deep layer (AP_{deep}), and the combined approach, *Dual Affinity Pooling* (DAP). To investigate the tradeoff between efficiency and accuracy, we introduce two configurations for the cosine similarity threshold τ: a *Conservative* setting (τ_{in}=0.9, τ_{deep}=0.8) prioritizing performance preservation, and an *Aggressive* setting (τ_{in}=0.8, τ_{deep}=0.7) aiming for maximum compression. For all other hyperparameters, we fix the layer indices at $l_{\text{in}} = 0$ and $l_{\text{deep}} = 29$, and the window sizes at $\omega_{\text{in}} = 1$ and $\omega_{\text{deep}} = 3$. All experiments utilize greedy decoding.

Datasets and Metrics. We evaluate on multiple benchmarks:(1) ASR: KeSpeech (Tang et al., 2021) and LibriSpeech (Panayotov et al., 2015) (WER); (2) QA: OpenBookQA, SDQA (Chen et al., 2024), and SpeechTriviaQA (He et al., 2024) (Accuracy); (3) ST: CoVost2 (Wang et al., 2020) en2zh and zh2en (BLEU). For efficiency, we report prefilling GFLOPs and Final Retention Ratio (FRR), defined as the percentage of audio tokens remaining after all compression stages. Further details on dataset specifications and evaluation protocols are provided in Appendix C.1 and C.2, respectively.

Main Results. Table 1 presents the performance and efficiency results. *Affinity Pooling* achieves substantial computational savings with negligible degradation across all tasks. In the *Aggressive* setting, DAP reduces the FRR to 14.91% and cuts prefilling GFLOPs by 27.48%. Despite this drastic reduction, it maintains comparable WER and BLEU scores to the baseline, while even improving QA accuracy. Notably, under the *Conservative* setting, AP_{in} outperforms the Vanilla baseline on all three task averages, while AP_{deep} and DAP also surpass Vanilla in both ASR and QA tasks. We ver-

Table 2: **Prefilling efficiency on H200.** Time-to-first-token (TTFT, ms), peak memory m, and dynamic increment Δm (GB), where $\mathrm{Spd.} = \mathrm{TTFT}_{\mathrm{Vanilla}}/\mathrm{TTFT}$ and $\mathrm{Sav.} = \Delta m_{\mathrm{Vanilla}}/\Delta m$.

Method	Time Efficiency			Memory Usage		
	TTFT (ms)↓	Spd. ↑	t_{AP} (ms)↓	m (GB)↓	Δm (GB)↓	Sav. ↑
Duration bucket: 40–60 s						
Vanilla	132.24	1.00×	-	33.40	1.99	1.00×
$\mathrm{AP}_{\mathrm{in}}$	117.36	1.13×	0.43	33.10	1.68	1.18×
$\mathrm{AP}_{\mathrm{deep}}$	131.31	1.01×	8.36	32.77	1.36	1.46×
DAP	117.87	1.12×	7.33	32.58	1.17	1.70×
Duration bucket: 20–40 s						
Vanilla	89.90	1.00×	-	32.53	1.15	1.00×
$\mathrm{AP}_{\mathrm{in}}$	83.58	1.08×	0.47	32.39	1.01	1.14×
$\mathrm{AP}_{\mathrm{deep}}$	89.70	1.00×	4.34	32.16	0.78	1.47×
DAP	84.08	1.07×	4.31	32.08	0.70	1.64×
Duration bucket: 0–20 s						
Vanilla	53.98	1.00×	-	31.79	0.41	1.00×
$\mathrm{AP}_{\mathrm{in}}$	54.01	1.00×	0.43	31.76	0.38	1.08×
$\mathrm{AP}_{\mathrm{deep}}$	55.74	0.97×	1.66	31.66	0.28	1.46×
DAP	55.50	0.97×	2.01	31.65	0.27	1.52×

Table 3: **ASR results under different token budgets.** WER (↓). Underlines denote the best performance within each budget, and boldface indicates results better than the Vanilla baseline.

Method	KES	LSC	LSO	*Avg.*
Vanilla	3.28	1.65	3.88	2.94
Budget: 90% Tokens				
speedup	7.85	6.14	18.45	10.81
interpolate	3.52	1.79	4.34	3.22
$\mathrm{AP}_{\mathrm{in}}$	3.84	**1.65**	**3.79**	3.09
Budget: 80% Tokens				
speedup	8.94	6.97	21.8	12.57
interpolate	3.86	1.84	4.25	3.32
$\mathrm{AP}_{\mathrm{in}}$	3.44	1.75	3.92	3.04
Budget: 70% Tokens				
speedup	11.92	11.28	32.98	18.73
interpolate	5.34	2.36	4.83	4.18
$\mathrm{AP}_{\mathrm{in}}$	4.04	2.21	4.42	3.56
Budget: 60% Tokens				
speedup	18.04	19.85	51.34	29.74
interpolate	8.29	3.98	6.96	6.41
$\mathrm{AP}_{\mathrm{in}}$	5.94	4.38	6.78	5.70

ify these findings generalize to Kimi-Audio, with full results in Appendix (Table 8).

We observe that while $\mathrm{AP}_{\mathrm{deep}}$ achieves more aggressive token reduction, $\mathrm{AP}_{\mathrm{in}}$ delivers greater computational savings due to cumulative effects across the entire network. This advantage is threshold-dependent: under *Aggressive* settings, $\mathrm{AP}_{\mathrm{in}}$ reduces FLOPs by 21.51% compared to 8.04% for $\mathrm{AP}_{\mathrm{deep}}$; under *Conservative* settings, both yield comparable efficiency gains (∼6%). Notably, DAP's FRR slightly exceeds $\mathrm{AP}_{\mathrm{deep}}$ alone in the *Aggressive* setting, suggesting that early compression may disrupt some long-range redundancies captured at deeper layers. Nevertheless, DAP offers a favorable balance between efficiency and performance across our evaluation suite.

5.2 Real-World Efficiency

To assess practical deployment viability, we measure inference latency and memory consumption on a single NVIDIA H200 GPU. We utilize the Qwen2-Audio model under the *Aggressive* configuration (τ_{in}=0.8, τ_{deep}=0.7) to establish an upper bound on potential efficiency gains. We curate a test set partitioned into three duration buckets: D_1 (0–20s), D_2 (20–40s), and D_3 (40–60s), with $n = 100$ randomly selected samples per bucket.

Since *Affinity Pooling* specifically optimizes the prompt processing stage, we focus our evaluation on prefilling metrics. We report the time-to-first-token (TTFT), representing the total wall-clock time from input to the first generated token, and explicitly isolate the computational overhead of our algorithm (t_{AP}). For memory, we track the peak VRAM usage (m) and the dynamic memory increment (Δm).

As shown in Table 2, latency gains become more noticeable for longer utterances. Across all three buckets, $\mathrm{AP}_{\mathrm{in}}$ consistently provides the largest wall-clock TTFT speedup, suggesting that early token reduction is a major contributor to end-to-end latency reduction. For long inputs (D_3), all compression scopes yield measurable gains, with DAP reaching up to $\sim 1.1\times$ speedup. In contrast to latency, DAP consistently reduces GPU memory across all duration buckets. The gain is most pronounced for long utterances (D_3), where the dynamic memory increment drops from 1.99 GB to 1.17 GB, corresponding to a $\sim 1.7\times$ memory saving.

5.3 Comparative Analysis

While our previous experiments demonstrate the robustness of deep layers to compression, the fidelity of the initial token selection at the input stage remains the critical bottleneck for information preservation. To rigorously evaluate the effectiveness of our similarity-based method against signal-agnostic approaches, we conduct a controlled experiment under fixed token budgets at the input layer. To ensure a fair comparison, we enforce strict token retention budgets ($K\%$) across all methods, ranging from 90% down to 60% of

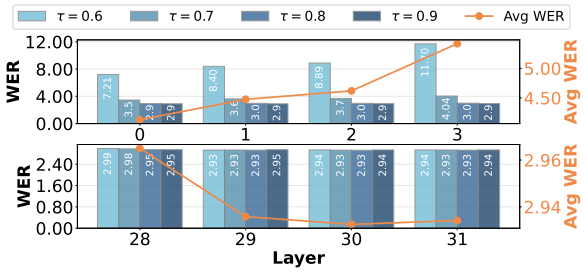


Figure 6: Layer sensitivity of Qwen2-Audio across early ($l \in [0, 3]$, top) and deep ($l \in [28, 31]$, bottom) layers.

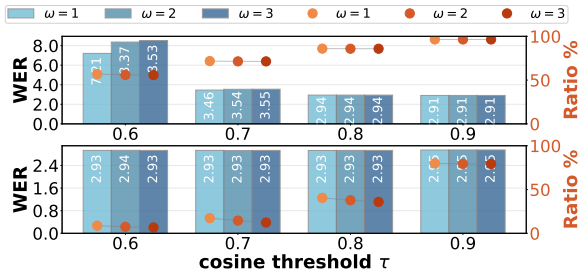


Figure 7: Lookback window ablation of Qwen2-Audio at input ($l = 0$, top) and deep layer ($l = 29$, bottom).

the original sequence length. We compare *Affinity Pooling* against two established baselines: (1) *Signal-level Speedup*, which accelerates the raw audio via time-stretching prior to encoding; and (2) *Linear Interpolation*, which uniformly downsamples the audio embedding sequence $\mathbf{H}_a^{(0)}$.

As detailed in Table 3, signal-agnostic methods suffer rapid degradation as the compression budget tightens. Specifically, *Signal-level Speedup* introduces significant distortion, leading to a drastic increase in WER. While *Linear Interpolation* performs better, it still consistently lags behind our approach. In contrast, *Affinity Pooling* demonstrates superior robustness. At the aggressive 60% budget, our method achieves a mean WER of 5.70%, significantly outperforming other methods. This performance gap highlights a fundamental limitation of fixed-rate compression: speech information is non-uniformly distributed. Rigid downsampling inevitably discards critical phonemic details in dense regions while preserving redundancy in silence. By aggregating tokens based on semantic affinity, our approach aligns better with the information density of the signal, thereby maximizing distinctiveness even under significant compression constraints.

5.4 Parameter Sensitivity

We conduct an ablation study to characterize the sensitivity of our method to its three governing hyperparameters: the application layer l, the similarity threshold τ, and the lookback window ω. All experiments are performed across three ASR tasks, and we report the average results to ensure generalizability. This analysis identifies the optimal operating points for Qwen2-Audio.

Optimal Layer and Threshold. We first examine the impact of the injection point by sweeping across early ($l \in [0, 3]$) and deep layers ($l \in [28, 31]$). As illustrated in Figure 6, the input embedding layer ($l=0$) provides the most favorable tradeoff among the shallow layers, effectively reducing sequence length with minimal impact on WER. However, performance degrades notably when compression is applied to the immediate subsequent layers ($l=[1, 3]$), even at conservative thresholds. In contrast, deep layers exhibit high stability; performance remains robust across a wide range of τ, confirming that the model's final representations can tolerate aggressive compression. These results support an asymmetric configuration: utilizing the input layer for initial reduction and a deep layer for maximizing compression. We observe consistent trends when replicating these experiments on Kimi-Audio (see Appendix B.2).

Impact of Lookback Window. We further investigate the temporal scope ω. As shown in Figure 7, the optimal window size depends on the depth. At the input level ($l=0$), the method is sensitive to wider windows; increasing ω beyond 1 leads to higher WER, indicating that strict adjacency is required to preserve acoustic details. Conversely, at deep layers ($l=29$), a wider lookback ($\omega=3$) improves the compression ratio without compromising accuracy. This justifies the design of DAP, which pairs a local constraint ($\omega=1$) at the input with a wider context ($\omega=3$) at deeper layers.

6 Conclusion

In this work, we explore the potential redundancy of dense tokenization in LSLMs. Through analysis of layer-wise representation evolution, we observe a transition from acoustic details to broader semantic abstractions. Building on these, we introduce *Affinity Pooling* as a training-free method to reduce computational load and memory usage during inference. Our experiments suggest that leveraging intrinsic feature similarity can be an effective alternative to fixed-rate processing. We hope these findings encourage further investigation into dynamic architectures that more closely align computation with the actual semantic content.

Limitations

Scope of Evaluation. While our experiments demonstrate the efficacy of the proposed compression mechanism on semantics-oriented speech tasks, its impact on fine-grained acoustic details remains underexplored.

Alignment Accuracy. Our oracle analysis relies on forced alignment to find word boundaries. However, these boundaries are approximations and may not perfectly match the actual acoustic transitions in natural speech. This could slightly affect the precision of our compression analysis.

Baseline Comparisons. Since token compression for LSLMs is a new field, there are very few open-source methods available for comparison. Therefore, we compared our approach primarily against standard signal processing techniques.

References

Swarup Ranjan Behera, Abhishek Dhiman, Karthik Gowda, and Aalekhya Satya Narayani. 2024. Fastast: Accelerating audio spectrogram transformer via token merging and cross-model knowledge distillation. *Preprint*, arXiv:2406.07676.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. 2023. Token merging: Your vit but faster. *Preprint*, arXiv:2210.09461.

Fan Bu, Yuhao Zhang, Xidong Wang, Benyou Wang, Qun Liu, and Haizhou Li. 2024. Roadmap towards superhuman speech understanding using large language models. *Preprint*, arXiv:2410.13268.

Yiming Chen, Xianghu Yue, Chen Zhang, Xiaoxue Gao, Robby T. Tan, and Haizhou Li. 2024. Voicebench: Benchmarking llm-based voice assistants. *Preprint*, arXiv:2410.17196.

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

Wenqian Cui, Dianzhi Yu, Xiaoqi Jiao, Ziqiao Meng, Guangyan Zhang, Qichao Wang, Yiwen Guo, and Irwin King. 2025. Recent advances in speech language models: A survey. *Preprint*, arXiv:2410.03751.

Tianyu Fu, Tengxuan Liu, Qinghao Han, Guohao Dai, Shengen Yan, Huazhong Yang, Xuefei Ning, and Yu Wang. 2025. Framefusion: Combining similarity and importance for video token reduction on large vision language models. *Preprint*, arXiv:2501.01986.

Chaoqun He, Renjie Luo, Shengding Hu, Yuanqian Zhao, Jie Zhou, Hanghao Wu, Jiajie Zhang, Xu Han, Zhiyuan Liu, and Maosong Sun. 2024. Ultraeval: A lightweight platform for flexible and comprehensive evaluation for llms. *Preprint*, arXiv:2404.07584.

Shengpeng Ji, Yifu Chen, Minghui Fang, Jialong Zuo, Jingyu Lu, Hanting Wang, Ziyue Jiang, Long Zhou, Shujie Liu, Xize Cheng, Xiaoda Yang, Zehan Wang, Qian Yang, Jian Li, Yidi Jiang, Jingzhen He, Yunfei Chu, Jin Xu, and Zhou Zhao. 2024. Wavchat: A survey of spoken dialogue models. *Preprint*, arXiv:2411.13577.

KimiTeam, Ding Ding, Zeqian Ju, Yichong Leng, Songxiang Liu, Tong Liu, Zeyu Shang, Kai Shen, Wei Song, Xu Tan, Heyi Tang, Zhengtao Wang, Chu Wei, Yifei Xin, Xinran Xu, Jianwei Yu, Yutao Zhang, Xinyu Zhou, Y. Charles, and 21 others. 2025. Kimi-audio technical report. *Preprint*, arXiv:2504.18425.

Yuri Kuratov, Mikhail Arkhipov, Aydar Bulatov, and Mikhail Burtsev. 2025. Cramming 1568 tokens into a single vector and back again: Exploring the limits of embedding space capacity. *Preprint*, arXiv:2502.13063.

Taehan Lee and Hyukjun Lee. 2025. *Token Pruning in Audio Transformers: Optimizing Performance and Decoding Patch Importance*. IOS Press.

Tianpeng Li, Jun Liu, Tao Zhang, Yuanbo Fang, Da Pan, Mingrui Wang, Zheng Liang, Zehuan Li, Mingan Lin, Guosheng Dong, Jianhua Xu, Haoze Sun, Zenan Zhou, and Weipeng Chen. 2025. Baichuan-audio: A unified framework for end-to-end speech interaction. *Preprint*, arXiv:2502.17239.

Yuang Li, Yu Wu, Jinyu Li, and Shujie Liu. 2023. Accelerating transducers through adjacent token merging. *Preprint*, arXiv:2306.16009.

Yueqian Lin, Yuzhe Fu, Jingyang Zhang, Yudong Liu, Jianyi Zhang, Jingwei Sun, Hai "Helen" Li, and Yiran Chen. 2025. Speechprune: Context-aware token pruning for speech information retrieval. *Preprint*, arXiv:2412.12009.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. 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.

Kele Shao, Keda Tao, Can Qin, Haoxuan You, Yang Sui, and Huan Wang. 2025a. Holitom: Holistic token merging for fast video large language models. *Preprint*, arXiv:2505.21334.

Kele Shao, Keda Tao, Kejia Zhang, Sicheng Feng, Mu Cai, Yuzhang Shang, Haoxuan You, Can Qin, Yang Sui, and Huan Wang. 2025b. When tokens talk too much: A survey of multimodal long-context token compression across images, videos, and audios. *Preprint*, arXiv:2507.20198.

Zhiyuan Tang, Dong Wang, Yanguang Xu, Jianwei Sun, Xiaoning Lei, Shuaijiang Zhao, Cheng Wen, Xi Tan, Chuandong Xie, Shuran Zhou, Rui Yan, Chenjia Lv, Yang Han, Wei Zou, and Xiangang Li. 2021. Kespeech: An open source speech dataset of mandarin and its eight subdialects. In *NeurIPS Datasets and Benchmarks*.

Keda Tao, Can Qin, Haoxuan You, Yang Sui, and Huan Wang. 2025. Dycoke: Dynamic compression of tokens for fast video large language models. *Preprint*, arXiv:2411.15024.

Changhan Wang, Anne Wu, and Juan Pino. 2020. Covost 2 and massively multilingual speech-to-text translation. *Preprint*, arXiv:2007.10310.

Hankun Wang, Yiwei Guo, Chongtian Shao, Bohan Li, Xie Chen, and Kai Yu. 2025a. Codecslime: Temporal redundancy compression of neural speech codec via dynamic frame rate. *Preprint*, arXiv:2506.21074.

Hualei Wang, Yiming Li, Shuo Ma, Hong Liu, and Xiangdong Wang. 2025b. Listening between the frames: Bridging temporal gaps in large audio-language models. *Preprint*, arXiv:2511.11039.

Zichen Wen, Yifeng Gao, Weijia Li, Conghui He, and Linfeng Zhang. 2025. Token pruning in multimodal large language models: Are we solving the right problem? *Preprint*, arXiv:2502.11501.

Senqiao Yang, Yukang Chen, Zhuotao Tian, Chengyao Wang, Jingyao Li, Bei Yu, and Jiaya Jia. 2024. Visionzip: Longer is better but not necessary in vision language models. *Preprint*, arXiv:2412.04467.

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

Rui-Chen Zheng, Wenrui Liu, Hui-Peng Du, Qinglin Zhang, Chong Deng, Qian Chen, Wen Wang, Yang Ai, and Zhen-Hua Ling. 2025. Say more with less: Variable-frame-rate speech tokenization via adaptive clustering and implicit duration coding. *Preprint*, arXiv:2509.04685.

A Experimental Details

A.1 Detailed Results of Layer-wise Oracle Interventions

Complementing the redundancy analysis presented in Section 3.2, we provide the comprehensive numerical results corresponding to the visual trends in Figure 2. Table 4 and Table 5 detail the performance metrics for Qwen2-Audio and Kimi-Audio, respectively. We report both standard WER and clamped WER (cWER) across all investigated compression operators (Random Drop, Uniform Drop, Uniform Merge) and token retention ratios. These quantitative results substantiate the observed hierarchy of representational density, confirming the distinct stability profiles of deep semantic layers compared to shallow acoustic layers.

A.2 Detailed Results of Similarity-Driven Interventions

To supplement the analysis of layer-wise dynamics in Section 4.2, we present the full numerical data. Table 6 and Table 7 detail the performance for Qwen2-Audio and Kimi-Audio, respectively. We report WER, clamped WER (cWER), and the resulting audio token retention ratio across varying cosine similarity thresholds ($\tau \in \{0.6, 0.7, 0.8, 0.9\}$) and injection layers. These empirical results corroborate the bimodal stability profile discussed in the main text, highlighting the robustness of input and deep-layer representations to aggressive compression compared to the sensitivity observed in intermediate layers.

B Extended Analysis on Kimi-Audio

B.1 Downstream Task Performance

We benchmark the proposed compression algorithm across ASR, QA, and Speech Translation tasks on Kimi-Audio, with results detailed in Table 8. Consistent with the observations on Qwen2-Audio, our method achieves substantial computational savings with negligible impact on semantic preservation.

Specifically, under the *Aggressive* setting, the *Dual Affinity Pooling* (DAP) strategy reduces the Final Retention Ratio (FRR) to $\sim$40.7%, translating to a significant reduction in prefilling GFLOPs. Despite this compression, the model maintains competitive accuracy on ASR and QA benchmarks compared to the Vanilla baseline.

Note on Speech Translation: We observe near-zero BLEU scores for the zh2en translation task across all settings, including the baseline. This performance stems from the intrinsic limitations of the Kimi-Audio base model in zero-shot cross-lingual generation for this specific direction, rather than artifacts introduced by our compression mechanism. We include these results solely for completeness.

B.2 Layer Sensitivity Analysis

We investigate the optimal injection points for compression by conducting a parameter sweep across shallow ($l \in [0, 3]$) and deep ($l \in [24, 27]$) layers on Kimi-Audio. Figure 8 illustrates the WER

Table 4: **Detailed WER results for Qwen2-Audio**. Standard WER (WER) and clamped WER (cWER) (in %) are reported for each operator, token ratio, and layer. The baseline WER is **1.65**.

Operator	Token Ratio	Layer 0		Layer 5		Layer 10		Layer 15		Layer 20		Layer 25		Layer 30	
		WER	cWER	WER	cWER	WER	cWER	WER	cWER	WER	cWER	WER	cWER	WER	cWER
Random Drop	25.67%	177.15	65.89	421.39	76.41	382.52	70.54	454.25	67.11	154.46	26.69	2.20	1.98	1.63	1.63
	47.57%	37.54	24.30	79.14	25.28	73.42	20.28	91.95	18.77	32.87	7.74	1.80	1.80	1.64	1.64
	76.08%	6.33	4.88	9.41	4.84	6.87	3.71	6.92	3.26	4.97	2.55	1.69	1.69	1.65	1.65
	97.73%	1.70	1.70	1.72	1.72	2.05	1.70	1.69	1.69	1.67	1.67	1.66	1.66	1.65	1.65
Uniform Drop	25.67%	141.13	58.21	274.25	61.83	244.44	53.83	300.56	49.31	102.06	18.25	2.38	1.91	1.63	1.63
	47.57%	23.78	17.29	42.49	16.98	38.46	13.39	46.25	12.23	20.78	5.54	1.79	1.79	1.64	1.64
	76.08%	3.23	3.23	3.41	2.84	4.00	2.56	4.28	2.30	2.43	1.97	1.70	1.70	1.65	1.65
	97.73%	1.73	1.73	1.71	1.71	1.69	1.69	1.68	1.68	1.66	1.66	1.66	1.66	1.65	1.65
Uniform Merge	25.67%	183.57	58.21	169.63	56.23	72.25	33.45	90.65	28.56	17.82	5.16	1.83	1.83	1.64	1.64
	47.57%	12.07	11.59	10.95	10.49	5.75	5.72	26.45	10.80	4.22	2.14	1.70	1.70	1.63	1.63
	76.08%	2.50	2.50	2.39	2.39	2.01	2.01	2.19	1.99	1.65	1.65	1.65	1.65	1.63	1.63
	97.73%	1.66	1.66	1.66	1.66	1.64	1.64	1.64	1.64	1.63	1.63	1.64	1.64	1.65	1.65

Table 5: **Detailed WER results for Kimi-Audio**. Standard WER (WER) and clamped WER (cWER) (in %) are reported for each operator, token ratio, and layer. The baseline WER is **1.34**.

Operator	Token Ratio	Layer 0		Layer 5		Layer 10		Layer 15		Layer 20		Layer 25	
		WER	cWER	WER	cWER	WER	cWER	WER	cWER	WER	cWER	WER	cWER
Random Drop	25.67%	74.81	65.80	71.57	64.08	92.47	63.68	186.67	60.51	251.76	55.47	1.90	1.90
	47.57%	26.09	25.53	25.39	24.14	24.64	22.75	25.80	16.50	49.52	15.76	1.59	1.59
	76.08%	4.11	4.11	3.36	3.36	3.21	3.21	2.35	2.35	3.97	2.37	1.35	1.35
	97.73%	1.37	1.37	1.36	1.36	1.36	1.36	1.35	1.35	1.32	1.32	1.32	1.32
Uniform Drop	25.67%	66.73	61.91	64.07	59.58	74.69	57.83	130.33	52.55	199.48	48.80	2.13	1.88
	47.57%	17.66	17.66	20.12	19.46	20.04	18.96	15.40	11.53	23.17	8.76	1.45	1.45
	76.08%	3.11	3.11	2.59	2.59	2.68	2.68	1.98	1.98	1.69	1.69	1.36	1.36
	97.73%	1.43	1.43	1.34	1.34	1.35	1.35	1.32	1.32	1.31	1.31	1.30	1.30
Uniform Merge	25.67%	54.32	51.48	51.52	50.40	50.33	46.91	28.78	22.81	8.09	5.73	1.47	1.47
	47.57%	13.58	13.58	15.24	15.24	14.33	13.54	5.41	5.20	1.98	1.98	1.38	1.38
	76.08%	1.98	1.98	1.94	1.94	1.90	1.90	1.58	1.58	1.40	1.40	1.31	1.31
	97.73%	1.34	1.34	1.35	1.35	1.33	1.33	1.35	1.35	1.31	1.31	1.32	1.32

Table 6: Performance of Qwen2-Audio across different layers and similarity thresholds τ. Metrics include WER, clamped WER (cWER), and token retention ratio (Ratio). The baseline WER is **1.65**.

Layer	$\tau = 0.6$			$\tau = 0.7$			$\tau = 0.8$			$\tau = 0.9$		
	WER	**cWER**	**Ratio**	**WER**	**cWER**	**Ratio**	**WER**	**cWER**	**Ratio**	**WER**	**cWER**	**Ratio**
0	6.85	6.51	57.61%	1.99	1.99	74.32%	1.63	1.63	88.60%	1.65	1.65	97.43%
5	17.82	17.36	46.67%	3.32	3.32	66.62%	1.66	1.66	84.71%	1.65	1.65	96.69%
10	6.03	5.63	52.57%	2.09	2.09	70.46%	1.66	1.66	86.74%	1.64	1.64	97.12%
15	38.83	14.70	46.52%	3.52	2.73	67.23%	1.65	1.65	85.92%	1.64	1.64	97.14%
20	90.98	18.19	22.53%	6.08	2.94	45.36%	1.68	1.68	74.12%	1.66	1.66	95.09%
25	2.04	2.04	10.55%	1.83	1.83	26.08%	1.68	1.68	58.06%	1.64	1.64	90.46%
30	1.64	1.64	5.18%	1.63	1.63	10.51%	1.65	1.65	33.89%	1.64	1.64	79.43%

Table 7: Performance of Kimi-Audio across different layers and similarity thresholds τ. Metrics include WER, clamped WER (cWER), and token retention ratio (Ratio). The baseline WER is **1.34**.

Layer	$\tau = 0.6$			$\tau = 0.7$			$\tau = 0.8$			$\tau = 0.9$		
	WER	**cWER**	**Ratio**	**WER**	**cWER**	**Ratio**	**WER**	**cWER**	**Ratio**	**WER**	**cWER**	**Ratio**
0	5.39	5.39	59.27%	1.64	1.64	78.35%	1.25	1.25	95.03%	1.29	1.29	99.51%
5	56.51	55.91	27.21%	14.68	14.65	48.58%	1.79	1.79	73.99%	1.32	1.32	95.73%
10	30.00	29.44	41.27%	6.43	6.43	61.39%	1.50	1.50	82.36%	1.30	1.30	97.80%
15	60.93	35.30	34.16%	6.15	5.81	56.26%	1.42	1.42	81.85%	1.31	1.31	98.37%
20	351.99	77.06	15.15%	56.22	18.69	37.70%	2.06	1.71	68.95%	1.30	1.30	95.48%
25	1.67	1.67	30.35%	1.47	1.47	47.80%	1.35	1.35	70.98%	1.28	1.28	92.10%

Table 8: **Main results and ablation studies on Kimi-Audio.** FRR: Final Retention Ratio. Bold indicates improvement over Vanilla, while underlining denotes the best performance within each setting.

Method	Scope		Efficiency (↓)			ASR (WER ↓)				QA (Acc ↑)				ST (BLEU ↑)		
	l_{in}	l_{deep}	FRR	Pre. GFLOPs	FLOPs Ratio	KES	LSC	LSO	***Avg.***	OBQA	SDQA	TrQA	***Avg.***	en2zh	zh2en	***Avg.***
Vanilla	-	-	100.0	407.22	100.0	2.71	1.34	2.58	2.21	81.98	36.71	33.79	50.83	2.02	0.0	1.01
Setting A: Aggressive (τ_{in}=0.80, τ_{deep}=0.70)																
AP_{in}	✓	-	86.28	351.18	86.24	2.73	**1.30**	**2.55**	**2.19**	**83.08**	**37.79**	**33.79**	**51.55**	2.01	**0.02**	**1.02**
AP_{deep}	-	✓	48.81	377.39	92.68	2.83	1.48	2.68	2.33	**82.20**	**37.79**	**33.79**	**51.26**	2.01	0.0	**1.01**
DAP	✓	✓	40.71	324.64	79.72	2.85	1.43	2.60	2.29	**82.42**	**37.61**	33.69	**51.24**	1.99	**0.02**	**1.01**
Setting B: Conservative (τ_{in}=0.90, τ_{deep}=0.80)																
AP_{in}	✓	-	96.44	392.69	96.43	2.72	**1.34**	2.59	2.22	**82.64**	**36.71**	**33.79**	**51.05**	2.01	0.0	**1.01**
AP_{deep}	-	✓	71.18	390.42	95.87	2.74	1.36	**2.58**	2.23	**82.64**	**37.61**	33.59	**51.28**	**2.02**	0.0	**1.01**
DAP	✓	✓	68.19	376.14	92.36	2.77	1.36	2.59	2.24	**81.98**	**36.89**	33.50	50.79	2.01	0.0	**1.01**

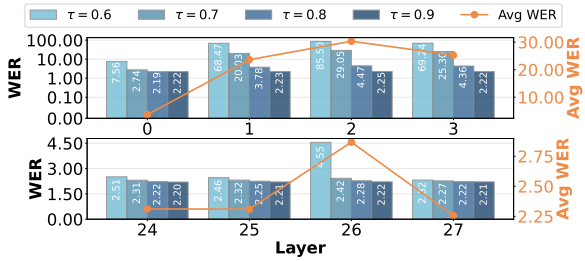


Figure 8: Layer sensitivity of Kimi-Audio across early ($l \in [0, 3]$, top) and deep ($l \in [24, 27]$, bottom) layers. The top plot uses a symlog axis.

dynamics under varying similarity thresholds τ.

Shallow Layers ($l \in [0, 3]$). The input embedding layer ($l = 0$) exhibits superior robustness, effectively balancing token reduction with acoustic fidelity. In contrast, injecting compression at immediate subsequent layers ($l \in [1, 3]$) leads to a sharp degradation in performance, confirming that early feature extraction layers are highly sensitive to structural perturbations.

Deep Layers ($l \in [24, 27]$). Deep representations exhibit remarkable robustness to compression across most operational regimes. Under moderate to conservative thresholds ($\tau \geq 0.7$), all examined layers maintain consistently low WER, corroborating the semantic redundancy hypothesis. However, when applying extremely aggressive compression ($\tau = 0.6$), we observe increased sensitivity at $l = 26$, suggesting that while deep layers generally tolerate substantial token reduction, excessively low similarity thresholds may disrupt critical feature alignments even at these abstract representation levels. This finding reinforces the importance of threshold calibration when targeting deep-layer compression.

C Benchmark and Evaluation Details

C.1 Benchmark Details

To comprehensively evaluate the robustness of our proposed compression algorithm, we conduct experiments across three distinct tasks: Automatic Speech Recognition (ASR), Speech Question Answering (QA), and Speech Translation (ST).

Automatic Speech Recognition We assess ASR performance using three datasets that vary in language and acoustic complexity. For English, we utilize the Librispeech corpus, reporting results on both the *test-clean* (**LSC**) and *test-other* (**LSO**)

splits. While **LSC** (2,620 samples) represents high-quality read speech, **LSO** (2,939 samples) introduces more challenging acoustic environments. To evaluate multilingual generalization, we employ the test set of KeSpeech (**KES**), comprising 5,000 Mandarin samples featuring diverse background noises and rich prosodic features. All ASR tasks are evaluated using WER.

Speech Question Answering We evaluate semantic understanding using three benchmarks. OpenBookQA (**OBQA**) consists of 455 long-form audio clips requiring multi-hop reasoning for single-choice questions. We also utilize SDQA-USA (**SDQA**), which contains 553 real-world spoken queries. Additionally, we incorporate SpeechTriviaQA (**TrQA**), a dataset sourced from the *TwinkStart* repository. **TrQA** comprises 1,020 synthetic speech samples, covering open-domain trivia (e.g., general knowledge and pop culture) to test the model's robustness against synthetic prosody. All QA tasks are evaluated using Accuracy (Acc).

Speech Translation For translation tasks, we utilize the CoVoST2 corpus, a large-scale multilingual dataset derived from Common Voice. We report results on the English-to-Chinese (**en2zh**, 15,531 samples) and Chinese-to-English (**zh2en**, 4,898 samples) directions. Both subsets are characterized by real-world recording conditions, including significant background noise and informal speech. Performance is measured using BLEU scores.

C.2 QA Evaluation Protocol

To assess the semantic accuracy of the Speech Question Answering tasks, we employ a model-based evaluation strategy rather than relying solely on exact string matching, which often penalizes valid paraphrases. We deploy *Qwen3-30B-A3B* as our automated evaluator.

The evaluation process involves a comparative analysis where the evaluator is presented with the original question text, the textual response generated by the LSLMs, and the ground-truth reference answer provided by the dataset. The evaluator is instructed to act as an assistant for audio model responses, specifically tasked with determining the correctness of the generated answer by comparing it against the reference. The prompt explicitly directs the model to accept paraphrasing and synonyms while marking the result as correct only if the core information aligns with the reference. The full prompt template is provided below:

Prompt for QA Evaluation Assistant

```
You are an evaluation assistant for audio model
responses.
## Task
Determine if an audio model's answer
("audio_answer") is correct by comparing with
reference ("ref"). Accept paraphrasing and
synonyms. Mark as correct only if core
information matches.
## Output only true or false
## Evaluate
Question: {question}
Audio Answer: {audio_answer}
Reference: {ref}
```

D Visualizations or Qualitative Results

D.1 Decoding Trajectories under Oracle Interventions

This section investigates the stability of intermediate representations under different compression operators (Random Drop, Uniform Drop, and Uniform Merge). Tables 9 and 10 present representative samples from the Librispeech-test-clean dataset showing the decoding trajectories for Qwen2-Audio and Kimi-Audio, respectively. We observe a vulnerability at intermediate layers where structural perturbations cause divergence, despite the model's ability to recover the ground truth at deeper layers ($l \geq 25$).

Qwen2-Audio (Table 9, we merge Layers 25 and 30 in the table since their decoded outputs are identical for brevity.) The model exhibits severe sensitivity to compression in the middle layers ($l \in [5, 20]$). While Random Drop and Uniform Drop primarily trigger repetition loops, Uniform Merge ($R = 2$) induces a distinct cross-lingual hallucination at Layer 10, generating fluent but unrelated Chinese text. This suggests that the internal representations in the middle layers are in a highly sensitive transitional state. Unlike deeper layers, these features lack the robustness to withstand structural compression, leading to immediate decoding failures like loops and hallucinations.

Kimi-Audio (Table 10). Similar instability is observed in Kimi-Audio. Notably, under Uniform Merge ($R = 1$), the model displays *semantic drift* at Layer 20 (e.g., paraphrasing "ghost" to "spirit" and "vesture" to "veil"). Unlike the repetition loops caused by Random Drop, this suggests that **the merged representations retain high-level semantic content** even when exact lexical alignment

Decoding Trajectories on Qwen2-Audio under Oracle Interventions

Ground Truth: It was strange too that he found an arid pleasure in following up to the end the rigid lines of the doctrines of the church and penetrating into obscure silences only to hear and feel the more deeply his own condemnation.

Random Drop ($R = 2$)

Layer 0: it was too that he an following up to the end the doctrines of the church and to hear and the own <|endoftext|>

Layer 5: it was strange too that he an own own own own own own... Looping

Layer 10: it was strange too that he an and and and and and and and and and and and and ... Looping

Layer 15: it was strange too that he he he he he he he he he he he he he he he he he he ... Looping

Layer 20: it was strange too that he he he he he he he he he he he he he he he he he he he Looping

Layer 25 & Layer 30: it was strange too that he found an arid pleasure in following up to the end the rigid lines of the doctrines of the church and penetrating into obscure silences only to hear and feel the more deeply his own condemnation <|endoftext|>

Uniform Drop ($R = 2$)

Layer 0: it was strange he heard the arduous labours of the church and the obscure toil of the church and the obscure toil of the church and the obscure toil of the church... Looping

Layer 5: it was strange he said he had no desire to hear the end of the story and he was glad to hear that the story was over ... Looping

Layer 10: it was strange he found an arid pleasure in following to the end the obscure of the church and to hear and feel the more own of the narration <|endoftext|>

Layer 15: it was strange too he he found an arid pleasure in following up to the end the the the the the the the the the the ... Looping

Layer 20: it was strange too that he found an arid pleasure in following up to the end the the the the the the the the the the the Looping

Layer 25 & Layer 30: it was strange too that he found an arid pleasure in following up to the end the rigid lines of the doctrines of the church and penetrating into obscure silences only to hear and feel the more deeply his own condemnation <|endoftext|>

Uniform Merge ($R = 2$)

Layer 0: it was strange that he found an arid in the up to the rigid lines of the of the and to and the more own <|endoftext|>

Layer 5: it was strange that he found an arid pleasure in following to the end the lines of the book which he had opened at random and which he had read through to the end <|endoftext|>

Layer 10: 它是一个人的心灵的镜子，它是一个人的心灵的镜子，它是一个人的心灵的镜子，它是一个人的心灵的镜子，它是一个人的心灵的镜子 ... Looping

Layer 15: it was strange too that he found an arid pleasure in following to the end the rigid lines of the doctrines of the church and penetrating into the arid lines of the doctrines of the church and penetrating into the arid lines of ... Looping

Layer 20: it was strange too that he found an arid pleasure in following to the end the rigid lines of the doctrines of the church and penetrating into obscure silences only to hear and feel the more deeply his own condemnation <|endoftext|>

Layer 25 & Layer 30: it was strange too that he found an arid pleasure in following up to the end the rigid lines of the doctrines of the church and penetrating into obscure silences only to hear and feel the more deeply his own condemnation <|endoftext|>

Table 9: Decoding Trajectories for an ASR example on Qwen2-Audio under Oracle Interventions. We consolidate Layers 25 and 30 into a single entry, as their decoded outputs are identical, to save space.

Decoding Trajectories on Kimi-Audio under Oracle Interventions

Ground Truth: A moment before the **ghost** of the ancient kingdom of the danes had looked forth through the vesture of the hazewrapped city

---

Random Drop ($R = 1$)

Layer 0: A ghost of the had for the of the. <|endoftext|>

Layer 5: Before ghost of the had the wraith of the <|endoftext|>

Layer 10: Before ghost of the had of the vesture of <|endoftext|>

Layer 15: The ghost of the ghost of the ghost of the ghost of the ghost of the ghost of the ghost of the ghost of the ghost of ... Looping

Layer 20: A ghost of the ghost of the ghost of the ghost of the ghost of the ghost of the ghost of the ghost of the ghost of the ghost of ... Looping

Layer 25: A moment before the ghost of the ancient kingdom of the danes had looked forth through the vesture of the haze wrapped city <|endoftext|>

Uniform Merge ($R = 1$)

Layer 0: The the of the city. <|endoftext|>

Layer 5: The the kingdom of the danes of the city <|endoftext|>

Layer 10: The ghost of the kingdom had the vesture of the city <|endoftext|>

Layer 15: The kingdom of the danes had the kingdom of the danes had the kingdom of the danes had the kingdom of the danes had the kingdom of the danes ... Looping

Layer 20: A moment before the **spirit** of the kingdom of the danes had passed through the veil of the misty night <|endoftext|>

Layer 25: A moment before the ghost of the ancient kingdom of the danes had looked forth through the vesture of the haze wrapped city <|endoftext|>

Uniform Drop ($R = 1$)

Layer 0: A moment later, the man was dead. <|endoftext|>

Layer 5: A moment the ghost of danes looked through the haze <|endoftext|>

Layer 10: A moment the ghost of the king of denmark had thrown off the vesture of death. <|endoftext|>

Layer 15: A moment the ghost of the ancient kingdom of the danes had looked forth through the haze of the city <|endoftext|>

Layer 20: A moment before the ghost of the ancient kingdom of the danes had looked forth through the vesture of the haze hazed of the danes had looked forth through the vesture of the ... Looping

Layer 25: A moment before the ghost of the ancient kingdom of the danes had looked forth through the vesture of the haze wrapped city <|endoftext|>

Table 10: Decoding Trajectories for an ASR example on Kimi-Audio under Oracle Interventions.

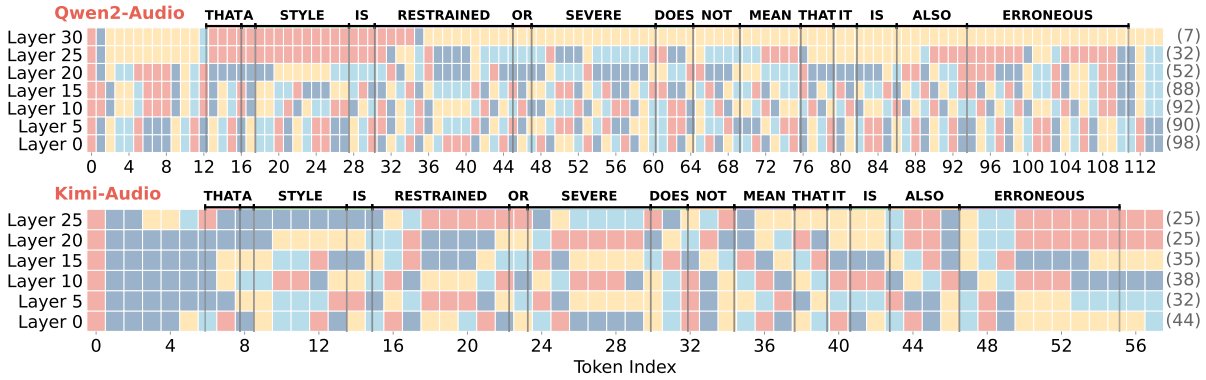
is temporarily lost. In all cases, the deeper layers demonstrate robustness, effectively correcting these intermediate distortions to reproduce the exact ground truth.

D.2 Extended Analysis on Semantic Granularity of Aggregated Tokens

To further substantiate the observations in Section 4.3 regarding the layer-wise evolution of representation density, we provide additional visualizations of *Affinity Pooling* applied to Qwen2-Audio and Kimi-Audio. Figures 9, 10, and 11 illustrate the token aggregation patterns ($\tau = 0.7, \omega = 3$) alongside the corresponding decoded transcripts across varying depths.

Consistent with our main findings, the visualizations reveal a distinct hierarchy in token granularity. In shallow layers ($l \leq 5$), the aggregation groups are fragmented and short. As depth increases, these groups expand significantly. By the deep layers ($l \geq 25$), single tokens frequently span entire phrases or multi-word clauses. For instance, in Figure 9, Qwen2-Audio compresses the input of 115 audio tokens into fewer than 10 tokens at Layer 30 while maintaining a perfect transcript.

The decoding trajectories further highlight the robustness of our similarity-based compression at the input and deep layers, while exposing the sensitivity of intermediate representations. Both models consistently recover the ground truth at the input layer ($l = 0$) and deep layers ($l \geq 25$). This validates our asymmetric design choice in *Dual Affinity Pooling* (DAP). However, we observe transient instability in the middle layers ($l \in [15, 20]$), particularly in Kimi-Audio. As seen in Figure 11, compression at Layer 20 triggers a repetition loop (*be ware of...*"), and in Figure 10, it induces a hallucination (*Ah, now, Nairn...*"). These errors vanish at Layer 25, suggesting that while intermediate layers are structurally fragile to aggregation, the final semantic layers reorganize these features into a highly robust, compressible format. These qualitative results reinforce our quantitative findings: redundancy in LSLMs is not uniform but structurally organized, transitioning from acoustic redundancy in shallow layers to semantic redundancy in deep layers.



Decoding Trajectories

Ground Truth: That a style is restrained or severe does not mean that it is also erroneous.

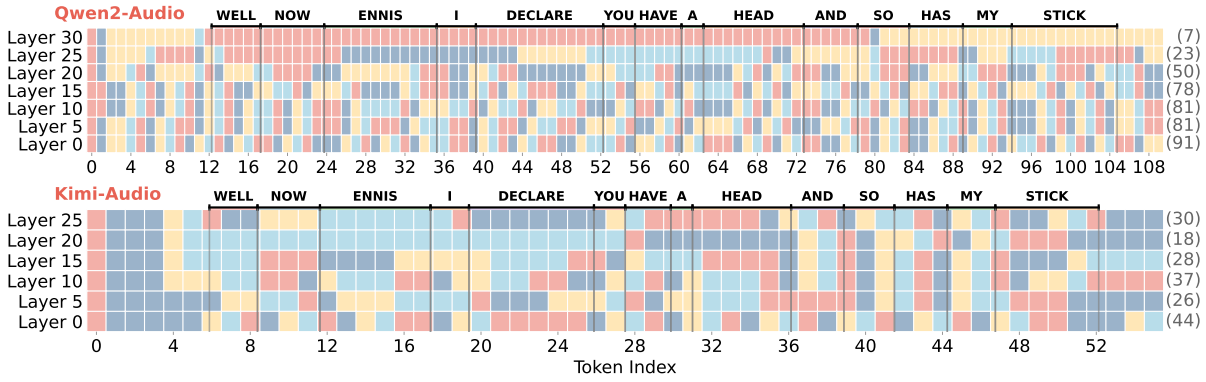
Qwen2-Audio

Layer 0: that a style is restrained or severe does not mean that it is also erroneous <|endoftext|>
Layer 5: that a style is restrained or severe does not mean that it is also erroneous <|endoftext|>
Layer 10: that a style is restrained or severe does not mean that it is also erroneous <|endoftext|>
Layer 15: that a style is restrained or severe does not mean that it is also erroneous <|endoftext|>
Layer 20: that a style is restrained or severe does not mean that it is also erroneous <|endoftext|>
Layer 25: that a style is restrained or severe does not mean that it is also erroneous <|endoftext|>

Kimi-Audio

Layer 0: That a style is restrained or severe does not mean that it is also erroneous. <|endoftext|>
Layer 5: A style restrained or severe does not mean that it is also erroneous. <|endoftext|>
Layer 10: That a style is restrained or severe does not mean that it is also erroneous. <|endoftext|>
Layer 15: A style is restrained or severe does not mean that it is also erroneous. <|endoftext|>
Layer 20: That a style is restrained or severe does not mean that it is also erroneous. <|endoftext|>
Layer 25: That a style is restrained or severe does not mean that it is also erroneous. <|endoftext|>

Figure 9: **Top: Visualization of *Affinity Pooling* (τ=0.7, $\omega = 3$) on Qwen2-Audio and Kimi-Audio**. Colors denote merged token groups, and vertical lines mark word boundaries. The right axis indicates the total number of tokens after compression. **Bottom: ASR transcripts decoded from the compressed representations at the corresponding layers**.



Decoding Trajectories

Ground Truth: Well now ennis i declare you have a head and so has my stick.

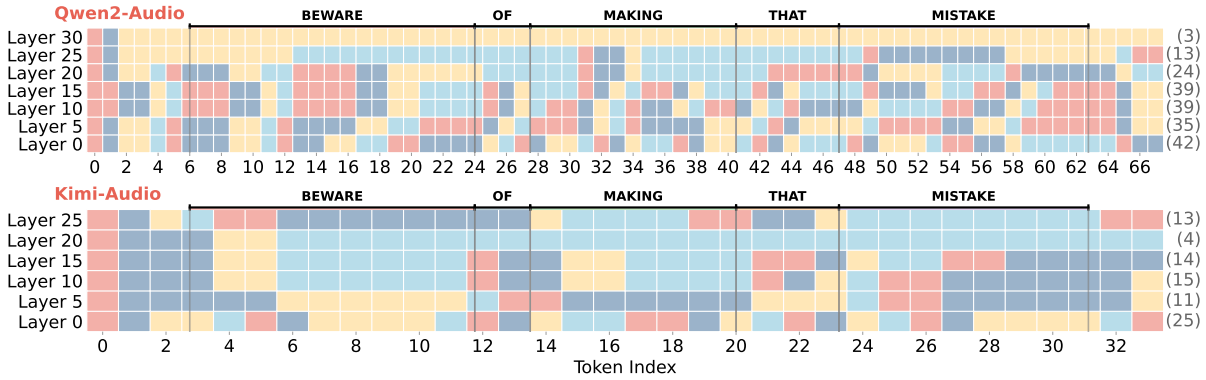
Qwen2-Audio

Layer 0: well now ennis i declare you have a head and so has my stick <|endoftext|>
Layer 5: well now ennis i declare you have a head and so has my stick <|endoftext|>
Layer 10: well now ennis i declare you have a head and so has my stick <|endoftext|>
Layer 15: well now ennis i declare you have a head and so has my stick <|endoftext|>
Layer 20: well now ennis i declare you have a head and so has my stick <|endoftext|>
Layer 25: well now ennis i declare you have a head and so has my stick <|endoftext|>

Kimi-Audio

Layer 0: Well now, Innes, I declare you have a head, and so has my stick. <|endoftext|>
Layer 5: Well now, I declare you have a head, so has my stick. <|endoftext|>
Layer 10: Well now, Innes, I declare you have a head, and so has my stick. <|endoftext|>
Layer 15: Well now, Innes, you have a head, and so has my stick. <|endoftext|>
Layer 20: \"Ah, now, Nairn, I'll have you to remember that I'm a gentleman.\" \"I'll remember, I'll remember, and so has my stick. <|endoftext|>
Layer 25: Well now, Innes, I declare you have a head, and so has my stick. <|endoftext|>

Figure 10: **Top: Visualization of *Affinity Pooling* (τ=0.7, $\omega = 3$) on Qwen2-Audio and Kimi-Audio**. Colors denote merged token groups, and vertical lines mark word boundaries. The right axis indicates the total number of tokens after compression. **Bottom: ASR transcripts decoded from the compressed representations at the corresponding layers**.



Decoding Trajectories

Ground Truth: Beware of making that mistake.

Qwen2-Audio

Layer 0: beware of making that mistake <|endoftext|>
Layer 5: beware of making that mistake <|endoftext|>
Layer 10: beware of making that mistake <|endoftext|>
Layer 15: beware of making that mistake <|endoftext|>
Layer 20: beware of making that mistake <|endoftext|>
Layer 25: beware of making that mistake <|endoftext|>

Kimi-Audio

Layer 0: Beware of making that mistake. <|endoftext|>
Layer 5: Beware of that mistake. <|endoftext|>
Layer 10: Beware of making that mistake. <|endoftext|>
Layer 15: \"Be ware of making that mistake. <|endoftext|>
Layer 20: \"Be ware of be ware of be ware of be ware of be ware of be ware of be ware of be ware of be ware of be ware of be ware of... Looping
Layer 25: \"Beware of making that mistake. <|endoftext|>

Figure 11: **Top: Visualization of *Affinity Pooling* (τ=0.7, $\omega = 3$) on Qwen2-Audio and Kimi-Audio**. Colors denote merged token groups, and vertical lines mark word boundaries. The right axis indicates the total number of tokens after compression. **Bottom: ASR transcripts decoded from the compressed representations at the corresponding layers**.