RAMP: Regime-Aware Merge–Prune Token Compression for ASR-Oriented Audio-Language Models

Jaehan Jung **Taehan Lee** **Hyukjun Lee***
Sogang University, South Korea
{jhjung22, alpaca, hyukjunl}@sogang.ac.kr

Abstract

ASR-oriented audio-language models (ALMs) encode speech into long audio token sequences for LLM decoding, substantially increasing the inference cost of transcription. This motivates token compression methods that reduce audio token sequence length while preserving the acoustic evidence needed for transcription. However, existing compression methods often inherit vision-language criteria that are poorly matched to temporally ordered audio representations, or require attention-map access outside the standard FlashAttention path. Through a multi-axis geometric analysis, we show that ALM audio representations exhibit two distinct representation regimes: a locally redundant encoder regime and a dispersed post-projector regime. Motivated by this geometric shift, we propose **RAMP** (**R**egime-**A**ware **M**erge–**P**rune), a training-free method for FlashAttention-compatible audio token compression that applies bipartite L_2 merging in the encoder and centroid-distance-guided pruning after the projector. Across seven ASR benchmarks and two primary ALMs, RAMP achieves the lowest average WER multiplier among the evaluated training-free compression baselines at every evaluated retention ratio. At the most aggressive setting ($r = 0.4$) on Audio-Flamingo-3, RAMP attains a $2.36\times$ average WER multiplier, compared with $4.26\times$ for the best non-RAMP baseline, while reducing time-to-first-token (TTFT) by 24.8% over Vanilla.

1 Introduction

Audio-language models (ALMs) encode audio input as token sequences for LLM decoding (Tang et al., 2024; Deshmukh et al., 2023; Gong et al., 2024; Ghosh et al., 2024; Chu et al., 2024; Lu et al., 2025; Xu et al., 2025; Fixie AI, 2025; Ghosh et al., 2025), often producing hundreds to thousands of tokens per utterance. These long audio token sequences substantially increase LLM prefill cost, making audio token compression a practical concern for ASR-oriented ALM inference, where the model is prompted to generate transcriptions and compression must preserve acoustic evidence needed for word-level recognition. Existing audio token compression methods often inherit design choices from vision-language compression. Fundamentally, token compression must balance removing redundancy (Bolya et al., 2023; Li et al., 2023b) with preserving informative tokens (Alvar et al., 2025; Zhang et al., 2025a). FlashAttention (Dao et al., 2022) accelerates exact attention without materializing the full attention matrix, so attention maps are not directly exposed along the standard execution path. Attention-dependent methods (Chen et al., 2024; Zhang et al., 2025b) rely on these maps, making attention map access a deployment constraint: a practical compression method should improve the accuracy–latency trade-off without reconstructing attention maps, disabling optimized kernels, or introducing backend-specific modifications. Attention-free methods (Bolya et al., 2023; Wen et al., 2025; Alvar et al., 2025; Zhang et al., 2025a; Jiang et al., 2025; Ma et al., 2026) avoid this issue, but are primarily designed for visual redundancy patterns, leaving it unclear whether visually motivated criteria transfer to ASR-oriented audio representations. Although recent audio-side methods (Li et al., 2023b; Lu et al., 2024; Lin et al., 2025; Gibier et al., 2026; Xiang et al., 2026) analyze or exploit redundancy in audio token sequences, they do not explicitly characterize how the representation geometry changes across the audio encoder–projector boundary, where acoustic hidden states are transformed into LLM-aligned audio tokens. This raises a stage-specific question: *should we apply a single compression criterion throughout an ALM, or should we match different operators to the representation geometry of each stage?*

*Corresponding author.

Findings of the Association for Computational Linguistics: EMNLP 2026, pages 34268–34288
October 24-29, 2026 ©2026 Association for Computational Linguistics

We argue that stage-agnostic compression overlooks a structural property of ALM representations. Through a multi-axis analysis combining intrinsic dimension, adjacent-token cosine similarity, anisotropy, and a token-level centroid distance D_C, we find that encoder representations are locally redundant, whereas post-projector LLM-aligned tokens become more dispersed. This contrast suggests that a single compression operator may not be equally effective across stages. We therefore propose **RAMP** (**R**egime-**A**ware **M**erge–**P**rune), a training-free two-stage compression method that combines encoder-side merging with post-projector pruning. RAMP applies bipartite L_2 merging at evenly spaced encoder layers and centroid-distance-guided greedy pruning at the projector output, aligning operator placement with the stage-wise geometry revealed by our representation analysis. Because all scores are computed from hidden states and projected tokens rather than attention maps, RAMP preserves the standard FlashAttention execution path without modification.

Our main contributions are as follows:

- We present a multi-axis geometric analysis of ALM audio representations, identifying two distinct regimes: a locally redundant encoder regime and a more dispersed post-projector regime.
- We propose **RAMP**, a training-free audio token compression method for FlashAttention-compatible deployment, built on a stage-wise operator placement principle: merging in the encoder and pruning after the projector.
- We evaluate RAMP across seven ASR benchmarks, three retention ratios, and two ALMs, analyzing operator placement, metric choice, FlashAttention-compatible latency, and token-rate generalization. At 40% token retention on Audio-Flamingo-3, RAMP achieves a $2.36\times$ average WER multiplier compared with $4.26\times$ for the best non-RAMP baseline and reduces TTFT by 24.8% over Vanilla.

2 Related Work

2.1 Multimodal Large Language Models

Multimodal large language models (MLLMs) extend pretrained LLMs to non-text modalities using modality-specific encoders and projectors that convert visual or audio inputs into token sequences for the language model. This architecture underlies representative vision-language models (Li et al., 2023a; Dai et al., 2023; Liu et al., 2023; Zhu et al., 2024) and audio-language models (Xu et al., 2025; Ghosh et al., 2025; Fixie AI, 2025). Because these non-text token sequences can reach hundreds to thousands of tokens, they substantially increase LLM-side computation and memory costs, motivating token compression.

2.2 Token Compression for VLMs

Vision token compression methods are commonly categorized into attention-dependent and attention-free approaches. **Attention-dependent** methods score tokens using attention maps. FastV (Chen et al., 2024) and SparseVLM (Zhang et al., 2025b) rely on LLM attention scores to select informative visual tokens. Because these methods require access to attention maps, they are incompatible with FlashAttention path unless attention maps are reconstructed or the inference backend is modified. **Attention-free** methods instead rely on feature-space criteria. ToMe (Bolya et al., 2023) merges similar tokens, and DART (Wen et al., 2025) prunes tokens based on duplication. DivPrune (Alvar et al., 2025), CDPruner (Zhang et al., 2025a), and G-Prune (Jiang et al., 2025) select tokens based on diversity, instruction-conditioned diversity, and graph-based semantic similarity, respectively, while ApET (Ma et al., 2026) uses approximation-error-guided pruning. However, these criteria are primarily designed for visual tokens, and whether they transfer effectively to temporally ordered audio tokens remains unclear.

2.3 Token Compression for ALMs

Compared with vision-side token compression, audio token compression for ALMs remains less explored. A-ToMe (Li et al., 2023b) performs encoder-side merging by combining adjacent audio tokens based on self-attention key similarity. FastAdaSP (Lu et al., 2024) uses weighted token merging for SpeechLM, while Affinity Pooling (Xiang et al., 2026) applies similarity-based merging at decoder-side input and deep layers. In contrast, SpeechPrune (Lin et al., 2025) uses speech-text relevance and approximated attention signals for token pruning, while Segmentwise Pruning (Gibier et al., 2026) applies temporally constrained Top-K selection. Although these methods show that both merging and pruning can reduce audio token sequences, they do not explicitly analyze how repre-

sentation geometry changes when acoustic encoder hidden states are transformed into post-projector LLM-aligned audio tokens. This gap motivates RAMP's stage-specific design, which applies merging in the encoder and pruning after the projector.

3 Methodology

3.1 ALM Preliminaries

An ALM consists of an audio encoder, a multimodal projector, and an LLM decoder. The audio encoder converts an input audio signal into acoustic hidden states $\mathbf{H} = \{\mathbf{h}_1, \ldots, \mathbf{h}_{N_H}\}$. The projector maps $\mathbf{H}$ into the LLM embedding space and may reduce the temporal length, producing post-projector audio tokens $\mathbf{A} = \{\mathbf{a}_1, \ldots, \mathbf{a}_{N_A}\}$, where typically $N_A \leq N_H$. The resulting audio tokens are concatenated with text prompt tokens $\mathbf{T} = \{\mathbf{t}_1, \ldots, \mathbf{t}_{N_T}\}$ according to a model-specific input template and passed to the LLM decoder for autoregressive generation.

3.2 Representation Analysis Tools

Prior audio token compression methods often rely primarily on similarity-based redundancy signals (Li et al., 2023b; Xiang et al., 2026). In contrast, we use four complementary geometric quantities to characterize how audio representations evolve from encoder hidden states to post-projector LLM-aligned tokens.

Intrinsic Dimension. We estimate intrinsic dimension via TwoNN (Facco et al., 2017), which infers the effective dimensionality of sampled token representations from the ratio between first- and second-nearest-neighbor distances. Intrinsic dimension has been used to characterize how transformer representations evolve across layers (Valeriani et al., 2023; Razzhigaev et al., 2024). A lower intrinsic dimension indicates that token representations vary along fewer effective degrees of freedom.

Adjacent-token Cosine Similarity. Adjacent-token cosine similarity quantifies local similarity between temporally neighboring audio tokens, serving as a measure of temporal smoothness.

Anisotropy. Anisotropy measures the extent to which embeddings are directionally concentrated in the representation space (Ethayarajh, 2019; Cai et al., 2021). We measure anisotropy at the utterance level by computing the average pairwise cosine similarity over distinct token pairs within each utterance. High anisotropy indicates that tokens within an utterance lie in a narrow directional cone, whereas low anisotropy corresponds to a more dispersed token space.

Centroid Distance (D_C). To capture token-level angular deviation from the per-utterance centroid, we define D_C for a token sequence $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ as

$$D_C(\mathbf{x}_i) = 1 - \cos(\mathbf{x}_i, \bar{\mathbf{x}}) \tag{1}$$

where $\bar{\mathbf{x}} = \frac{1}{N}\sum_{j=1}^{N} \mathbf{x}_j$ denotes the per-utterance centroid. Tokens with high D_C are angularly distant from the centroid, whereas tokens with low D_C remain close to it. Unlike anisotropy, which summarizes utterance-level directional concentration, D_C provides token-level deviation scores; we summarize the upper tail of these scores using the 95th percentile, denoted as p_{95}.

3.3 Multi-axis Representation Analysis

Figure 1 characterizes audio representations across audio encoder layers and the projector output for Audio-Flamingo-3 and Qwen2.5-Omni-7B on LibriSpeech *test-clean*. It reports the four geometric quantities: (a) intrinsic dimension, (b) adjacent-token cosine similarity, (c) anisotropy, and (d) p_{95} of D_C. Throughout this section, paired numerical values are reported in this model order.

Encoder: locally redundant representations. Across both models, the encoder layers exhibit consistent trends across the four metrics. First, intrinsic dimension decreases with depth, suggesting that deeper layers have fewer effective degrees of freedom. Second, adjacent-token cosine similarity stays high across most layers, reflecting strong local temporal smoothness. Third, anisotropy remains high across most layers and decreases toward the encoder output, indicating that encoder tokens remain directionally concentrated before projection. Finally, p_{95} of D_C stays modest across most layers and increases toward the encoder output, suggesting that the encoder contains only a limited tail of high-D_C tokens. Together, these trends characterize encoder tokens as locally smooth and directionally concentrated.

Projector output: dispersed LLM-aligned representations. Projection consistently changes all four metrics. Intrinsic dimension increases from the last sampled encoder layer to the projector output, from 4.2 to 6.7 and from 4.8 to 8.4. Adjacent-token cosine similarity drops to 0.52 and 0.51, showing that the strong local smoothness observed in the encoder is largely lost. Anisotropy also decreases

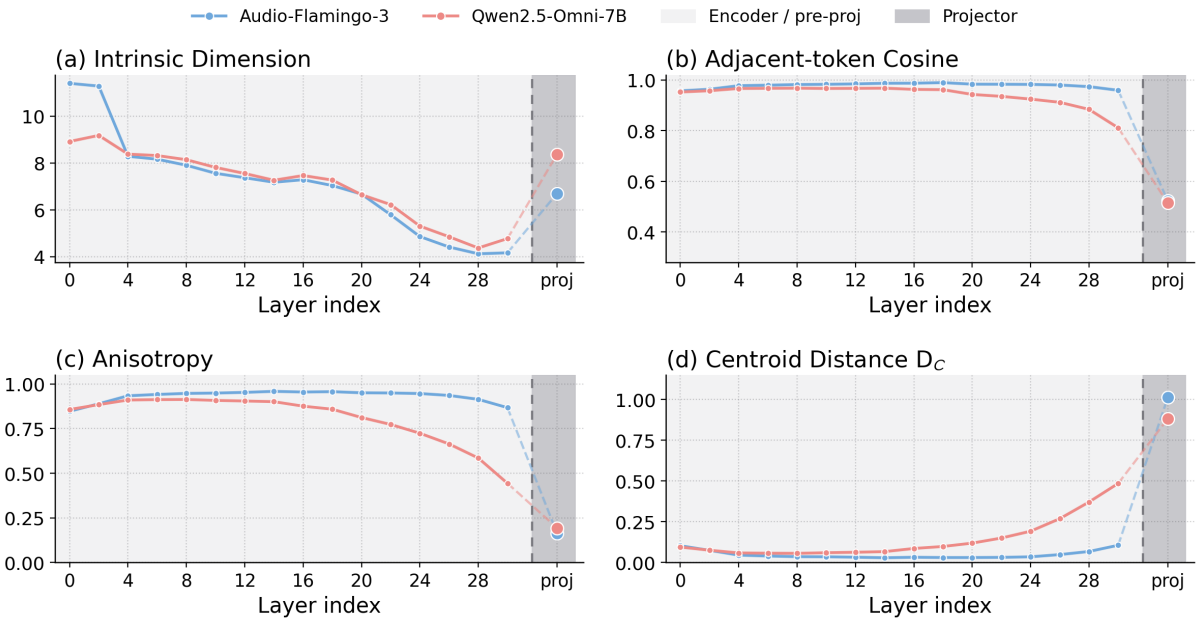


Figure 1: Layer-wise representation analysis for Audio-Flamingo-3 and Qwen2.5-Omni-7B on the full LibriSpeech *test-clean* set. We report (a) intrinsic dimension, (b) adjacent-token cosine similarity, (c) anisotropy, and (d) p_{95} of D_C across audio encoder layers and the projector output.

to 0.16 and 0.19, indicating that projected tokens become less directionally concentrated. The D_C tail expands, with p_{95} increasing from 0.11 to 1.01 and from 0.49 to 0.88. Together, these shifts show that projection maps locally smooth encoder representations into a more dispersed LLM-aligned space with an expanded high-D_C tail.

Two distinct regimes. These measurements reveal two structurally distinct representation regimes. The encoder is locally smooth and directionally concentrated, whereas the projector output is more dispersed and has an expanded high-D_C tail. This contrast motivates stage-specific operator placement rather than a single stage-agnostic criterion.

3.4 RAMP: Regime-Aware Merge–Prune Compression

RAMP compresses audio tokens in two stages: encoder-side merging followed by post-projector pruning. Figure 2 shows the overall pipeline.

Stage 1: encoder-side merging. Motivated by the locally redundant encoder regime, we apply bipartite L_2-based token merging at four evenly spaced encoder layers, using $\{7, 15, 23, 31\}$ (zero-indexed) for 32-layer encoders. The encoder-side reduction budget is distributed uniformly across these selected merge layers. At each merge layer, tokens are split into source and target sets using alternating temporal positions. Each source token is matched to its nearest target under L_2 distance, and the matched pairs with the smallest distances are merged until the per-layer reduction budget is reached. Targets are updated via size-weighted averaging following ToMe (Bolya et al., 2023), while unmerged source tokens are preserved.

Although matching is not explicitly restricted to adjacent pairs, the encoder regime is locally smooth and directionally concentrated, making nearest-neighbor matching suitable for identifying redundant tokens without imposing an adjacency constraint. Because these redundant tokens carry overlapping acoustic evidence, merging can reduce sequence length while preserving information from both merged tokens. We rank merge candidates by L_2 distance rather than cosine similarity because cosine similarities among locally redundant encoder tokens are saturated, whereas L_2 preserves magnitude-sensitive variation that better separates merge candidates. We compare merge scoring metrics (L_2 vs. cosine) and the use of an adjacency constraint in Appendix C. After each merge layer, we sort merged and unmerged tokens by temporal indices to preserve the temporal order of audio tokens.

Stage 2: post-projector pruning. In the post-projector regime, adjacent-token similarity is substantially lower than in the encoder, making nearest-neighbor merging less reliable. Moreover, averag-

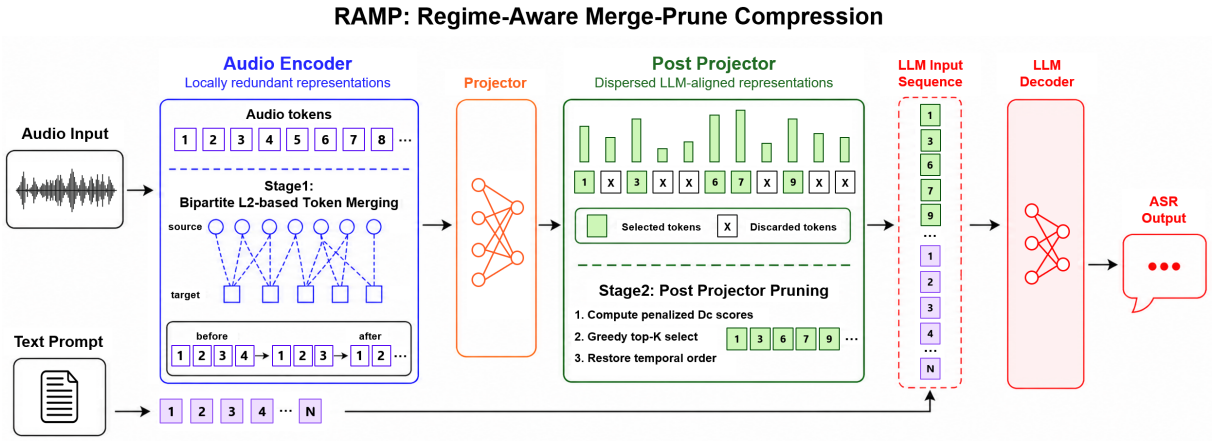


Figure 2: Overview of RAMP. RAMP applies encoder-side merging to exploit local redundancy and post-projector pruning to select a centroid-distance-guided subset of audio tokens before LLM decoding.

ing dispersed LLM-aligned embeddings may suppress angularly distinctive information carried by individual tokens. We therefore use pruning-based token selection, which retains or removes tokens without averaging. For scoring, we use D_C as a geometry-preserving selection criterion in the dispersed post-projector regime. Since projection expands the high-D_C tail and reduces local redundancy, retaining centroid-distinctive tokens helps preserve the angular coverage in the LLM-aligned token space under a fixed token budget. We compare D_C-based selection with low-D_C, random, and uniform alternatives in Appendix C.

During greedy selection, we penalize candidates that are temporally close to already selected tokens. Let i and j denote temporal position indices in the post-projector token sequence, and let t denote the greedy selection step. For candidate token i at step t, we define the penalized score as

$$\tilde{s}_i^{(t)} = D_C(\mathbf{a}_i) - \sum_{j \in \mathcal{S}_{t-1}} \lambda \exp\left(-\frac{(i-j)^2}{2\sigma^2}\right) \quad (2)$$

where $\mathcal{S}_{t-1}$ is the set of tokens selected in previous steps, $\lambda = 1 - r_{\text{proj}}$, and $\sigma = 2$. The Gaussian penalty discourages selecting tokens near previously selected ones without imposing a hard exclusion window. We then select the candidate with the highest penalized score:

$$i_t = \arg\max_{i \notin \mathcal{S}_{t-1}} \tilde{s}_i^{(t)} \quad (3)$$

We repeat this selection until the projector-side token budget is reached. The selected tokens are then sorted by their original temporal indices before being passed to the LLM.

Two-stage budget split. Given a final retention ratio r, we allocate the total token reduction between the encoder and projector stages using $\beta \in [0, 1]$:

$$r_{\text{enc}} = 1 - \beta(1 - r), \qquad r_{\text{proj}} = \frac{r}{r_{\text{enc}}} \quad (4)$$

Here, r_{enc} denotes the Stage 1 retention ratio, and r_{proj} denotes the Stage 2 retention ratio applied at the projector output, so the final retention ratio is $r_{\text{enc}} r_{\text{proj}} = r$. Appendix B analyzes the sensitivity to β and suggests that lower post-projector token rates favor a more encoder-heavy allocation. Based on this observation, we use $\beta = 0.4$ for the two 25 Hz ALMs and $\beta = 0.8$ for the lower-rate 6.25 Hz model, fixing each value across datasets and retention ratios.

4 Experiments

4.1 Experimental Setup

Models. Our main evaluation uses Audio-Flamingo-3 (Ghosh et al., 2025) and Qwen2.5-Omni-7B (Xu et al., 2025), both operating at a 25 Hz audio token rate. For generalization, we further evaluate Ultravox v0.6 Llama-3.1-8B (Fixie AI, 2025), which uses a lower 6.25 Hz audio token rate and a different LLM family, in Appendix A.

Datasets. We evaluate on seven widely used English ASR benchmarks covering diverse speech domains: LibriSpeech *test-clean* and *test-other* (Panayotov et al., 2015), SPGISpeech (O'Neill et al., 2021), TEDLIUM (Hernandez et al., 2018), GigaSpeech (Chen et al., 2021), Common Voice 15 (Ardila et al., 2020), and VoxPopuli (Wang et al., 2021).

Baselines. We compare RAMP against six training-free compression baselines: A-ToMe (Li

Method		LS-clean	LS-other	SPGI	TED	Giga	CV	VP	Avg. × ↓
Vanilla		1.51	3.03	1.73	3.31	9.90	8.20	5.70	—
Retention ratio $r = 0.6$									
A-ToMe	Interspeech 23	2.08	4.03	2.17	4.38	11.42	8.26	6.59	1.23×
DART	EMNLP 25	2.30	4.18	2.97	6.38	13.39	7.90	6.60	1.43×
CDPruner	NeurIPS 25	7.49	8.85	8.86	12.20	19.27	9.11	12.10	3.12×
DivPrune	CVPR 25	6.55	7.58	4.89	9.23	16.86	8.86	8.98	2.40×
G-Prune	AAAI 25	38.61	43.84	30.53	36.46	44.83	60.33	49.45	12.75×
ApET	CVPR 26	37.65	39.70	21.88	33.76	38.89	53.67	45.31	11.33×
RAMP		**1.81**	**3.53**	**1.93**	**3.93**	**10.34**	**7.45**	**5.97**	**1.10×**
Retention ratio $r = 0.5$									
A-ToMe	Interspeech 23	4.00	6.49	3.55	6.80	15.23	10.06	8.33	1.87×
DART	EMNLP 25	4.48	7.00	5.02	9.47	17.50	8.30	8.04	2.18×
CDPruner	NeurIPS 25	15.43	16.10	15.84	20.50	27.54	12.30	19.17	5.50×
DivPrune	CVPR 25	13.18	14.87	9.22	15.51	23.29	11.47	13.36	4.25×
G-Prune	AAAI 25	57.54	62.26	43.93	45.97	59.16	76.20	65.76	17.82×
ApET	CVPR 26	59.86	62.92	39.05	56.77	56.49	72.09	65.42	18.02×
RAMP		**2.52**	**4.51**	**2.58**	**5.17**	**12.34**	**8.09**	**6.67**	**1.37×**
Retention ratio $r = 0.4$									
A-ToMe	Interspeech 23	12.18	16.12	9.90	14.83	25.65	14.05	14.49	4.35×
DART	EMNLP 25	12.15	15.34	10.36	14.87	25.20	11.60	13.05	4.26×
CDPruner	NeurIPS 25	28.71	28.78	27.13	32.64	39.27	19.79	30.87	9.41×
DivPrune	CVPR 25	27.83	30.33	19.43	27.54	35.35	20.10	23.07	8.29×
G-Prune	AAAI 25	73.90	75.56	56.42	58.15	72.39	90.67	74.47	22.21×
ApET	CVPR 26	90.12	87.25	57.78	77.46	76.72	93.39	80.46	25.50×
RAMP		**5.36**	**8.44**	**5.36**	**8.88**	**17.83**	**9.47**	**8.44**	**2.36×**

Table 1: Main results on Audio-Flamingo-3. Benchmark columns report WER (%), and Avg. × denotes the mean of dataset-level WER ratios relative to Vanilla across seven benchmarks (lower is better). **Bold** and underline indicate the best and second-best compression methods within each retention-ratio group, respectively.

et al., 2023b) is applied within the audio encoder, ApET (Ma et al., 2026) is applied to encoder hidden states before projection, and the pruning criteria of DART (Wen et al., 2025), CDPruner (Zhang et al., 2025a), DivPrune (Alvar et al., 2025), and G-Prune (Jiang et al., 2025) are adapted to post-projector audio tokens. All baselines and RAMP use the same audio token retention ratio r, and compressed tokens are temporally sorted before being passed to the LLM decoder, enabling a controlled comparison under the same token budget. We use each baseline in a direct training-free setting under this shared budget and temporal-order constraint, without method-specific audio tuning or placement search. Attention-dependent compressors are excluded from the main comparison because they require explicit attention-map access, forcing a separate path outside the standard FlashAttention-enabled deployment setting.

Metrics. We report word error rate (WER) as the primary ASR metric. For inference efficiency, we measure time-to-first-token (TTFT) on an RTX PRO 4500 using vLLM 0.20.1 (Kwon et al., 2023). TTFT includes the audio forward pass (audio encoding, multimodal projection, and compression), followed by LLM prefill before autoregressive decoding. All evaluations use greedy decoding.

4.2 Main Results

Tables 1 and 2 report WER across seven ASR benchmarks and three retention ratios for Audio-Flamingo-3 and Qwen2.5-Omni-7B. For each compression method, Avg. × measures the average dataset-level WER multiplier relative to Vanilla, so lower values indicate less WER degradation under compression.

RAMP best preserves WER under compression. RAMP obtains the lowest Avg. × at every retention ratio on both ALMs. On Audio-Flamingo-3, RAMP achieves 1.10×, 1.37×, and 2.36× at $r = 0.6$, 0.5, and 0.4, respectively, compared with 1.23×, 1.87×, and 4.26× for the best non-RAMP baseline. On Qwen2.5-Omni-7B, RAMP achieves 1.22×, 1.77×, and 3.65×, compared with 1.40×, 2.33×, and 5.14× for the best non-RAMP baseline. RAMP is the best compression method in nearly all benchmark-level settings; the main exception is Common Voice 15 on Qwen2.5-Omni-7B, where it remains second-best across retention ratios. Among the stronger non-RAMP baselines, A-ToMe and DART remain competitive under moderate compression, but their gap from RAMP widens as the token budget tightens. This suggests that RAMP's regime-aware placement is especially beneficial under aggressive compression.

Method		LS-clean	LS-other	SPGI	TED	Giga	CV	VP	Avg. × ↓
Vanilla		1.67	3.27	2.22	3.40	9.89	7.69	6.00	—
Retention ratio $r = 0.6$									
A-ToMe	Interspeech 23	2.54	4.82	3.27	5.00	11.96	10.23	7.91	1.40×
DART	EMNLP 25	4.26	6.10	3.82	6.07	14.48	**8.64**	8.46	1.70×
CDPruner	NeurIPS 25	8.78	11.82	8.51	9.71	18.06	14.21	10.08	2.99×
DivPrune	CVPR 25	5.60	6.97	6.00	8.07	14.93	10.48	9.10	2.14×
G-Prune	AAAI 25	10.27	14.25	11.28	8.97	17.27	39.15	16.16	3.97×
ApET	CVPR 26	35.61	41.06	22.99	31.89	43.26	20.16	18.04	9.09×
RAMP		**2.15**	**4.26**	**2.80**	**4.38**	**11.02**	9.00	**6.86**	**1.22**×
Retention ratio $r = 0.5$									
A-ToMe	Interspeech 23	5.26	8.63	6.25	7.75	15.67	14.84	11.35	2.33×
DART	EMNLP 25	8.11	10.16	7.03	9.02	19.16	**11.40**	12.06	2.74×
CDPruner	NeurIPS 25	20.03	22.54	18.63	18.77	27.66	24.60	17.35	5.96×
DivPrune	CVPR 25	11.00	13.21	11.90	13.49	20.96	15.82	14.21	3.79×
G-Prune	AAAI 25	19.01	23.71	21.30	15.52	24.43	54.37	27.34	6.70×
ApET	CVPR 26	59.14	59.17	36.82	52.32	57.40	31.68	28.58	14.31×
RAMP		**3.80**	**6.69**	**4.53**	**5.87**	**13.45**	11.67	**8.42**	**1.77**×
Retention ratio $r = 0.4$									
A-ToMe	Interspeech 23	17.47	22.04	17.48	18.50	27.47	28.14	23.06	5.83×
DART	EMNLP 25	15.73	18.72	16.25	15.92	27.95	**18.75**	21.21	5.14×
CDPruner	NeurIPS 25	38.08	41.61	36.82	35.25	43.52	43.81	33.37	11.16×
DivPrune	CVPR 25	24.80	28.23	25.65	25.21	33.53	30.19	26.67	7.74×
G-Prune	AAAI 25	33.07	39.15	37.62	28.37	37.75	69.75	45.36	11.07×
ApET	CVPR 26	65.36	71.64	51.04	68.75	68.89	46.78	44.35	17.81×
RAMP		**10.04**	**13.80**	**11.33**	**11.27**	**20.19**	19.66	**13.79**	**3.65**×

Table 2: Main results on Qwen2.5-Omni-7B. Benchmark columns report WER (%), and Avg. × denotes the mean of dataset-level WER ratios relative to Vanilla across seven benchmarks (lower is better). **Bold** and underline indicate the best and second-best compression methods within each retention-ratio group, respectively.

4.3 Inference Efficiency

Table 3 reports Audio-Flamingo-3 inference latency at $r = 0.4$, decomposed into audio forward time, LLM prefill time, and TTFT. Since all compression methods use the same final audio token retention ratio, their LLM-side prefill times are nearly identical, and TTFT differences mainly come from method-specific audio-forward overhead. RAMP achieves the lowest TTFT among the compared methods at 42.4 ms, corresponding to a 24.8% reduction over Vanilla. Importantly, RAMP achieves this efficiency with a substantially lower Avg. × than the strongest non-RAMP baseline at the same retention ratio (2.36× vs. 4.26×). Thus, RAMP's main advantage is a stronger accuracy–latency trade-off under the same final token budget, rather than a large latency margin over every compression method.

Method	Audio forward		LLM prefill		TTFT	
	ms	Δ% ↑	ms	Δ% ↑	ms	Δ% ↑
Vanilla	16.7	—	39.7	—	56.4	—
A-ToMe	17.9	-7.2%	**27.1**	**+31.7%**	45.0	+20.2%
DART	16.7	0.0%	27.2	+31.5%	43.9	+22.2%
CDPruner	22.1	-32.3%	27.2	+31.5%	49.3	+12.6%
DivPrune	17.9	-7.2%	27.2	+31.5%	45.1	+20.0%
G-Prune	15.8	+5.4%	27.3	+31.2%	43.1	+23.6%
ApET	16.6	+0.6%	27.2	+31.5%	43.8	+22.3%
RAMP	**15.3**	**+8.4%**	**27.1**	**+31.7%**	**42.4**	**+24.8%**

Table 3: Inference latency of Audio-Flamingo-3 at $r = 0.4$ on 2000 randomly sampled VoxPopuli examples. Δ% denotes relative speedup over Vanilla for each component; positive values indicate speedup and negative values indicate slowdown.

5 Ablation Studies

5.1 Regime-Aware Operator Placement

To test the importance of placing merging in the encoder regime and pruning in the post-projector regime, we compare RAMP with one-stage variants, same-operator two-stage variants, and a swapped-operator variant. Across all variants, merging uses the same bipartite L_2 matching with size-weighted averaging as RAMP, and pruning uses the same D_C-based greedy selection with the temporal penalty. One-stage variants apply the full retention ratio at a single stage, while all two-stage variants use the same β-based budget split as RAMP to isolate the effect of operator placement. Full placement results are provided in Appendix D.

Table 4 shows that the one-stage variants obtain only partial gains: encoder-only merging and post-projector-only pruning reach 2.82× and 2.79×, re-

Placement	Avg. × ↓
Vanilla	—
One-stage variants	
Encoder merge only	2.82×
Post-projector prune only	2.79×
Two-stage variants	
Merge → Merge	9.54×
Prune → Prune	4.08×
Prune → Merge (Swap)	13.18×
Merge → Prune (RAMP)	**2.36×**

Table 4: Operator placement ablation on Audio-Flamingo-3 at $r = 0.4$. For two-stage variants, placement is written in encoder → post-projector order. Avg. × denotes the mean of dataset-level WER ratios relative to Vanilla across seven benchmarks (lower is better).

Method	FA enabled		FA disabled	
	ms ↓	Δ% ↑	ms ↓	Δ% ↑
Vanilla	16.7	—	27.0	—
A-ToMe	17.9	-7.2%	**23.1**	**+14.4%**
DART	16.7	0.0%	27.1	-0.4%
CDPruner	22.1	-32.3%	32.4	-20.0%
DivPrune	17.9	-7.2%	28.2	-4.4%
G-Prune	15.8	+5.4%	26.2	+3.0%
ApET	16.6	+0.6%	27.2	-0.7%
RAMP	**15.3**	**+8.4%**	23.2	+14.1%

Table 5: Audio forward latency of Audio-Flamingo-3 at $r = 0.4$ on 2000 randomly sampled VoxPopuli examples, with and without FlashAttention (FA). Δ% denotes relative speedup over Vanilla within the same FA condition; positive values indicate speedup and negative values indicate slowdown.

spectively, each exploiting only one regime while leaving the other regime's structure unexploited. RAMP leverages both regimes by combining the two operators in a regime-matched order, achieving the lowest Avg. × among all variants at 2.36×: merging removes redundancy in the locally smooth encoder regime, while pruning preserves centroid-distinctive projected tokens in the dispersed post-projector regime. The remaining two-stage placements show that using two stages alone is not sufficient; when the operator assignment is not matched to the corresponding representation regimes, the mismatch cost outweighs the benefit of adding another compression stage, raising Avg. × to 4.08×–13.18×. The swapped placement causes the largest degradation, reaching 13.18×, further highlighting the importance of assigning each operator to the appropriate representation regime.

5.2 FlashAttention-Compatible Deployment

We analyze audio-forward latency with FlashAttention (FA) enabled and disabled. Table 5 reports speedup over Vanilla within each FA condition. Across methods, enabling FA substantially reduces audio-forward latency, confirming that FA is a key part of the optimized inference path. RAMP improves over Vanilla in both settings: it reduces latency from 16.7 ms to 15.3 ms with FA enabled and from 27.0 ms to 23.2 ms with FA disabled. The comparison also shows that latency gains measured without FA do not always transfer to the FA-enabled path. For example, A-ToMe is faster than Vanilla without FA (23.1 ms vs. 27.0 ms), but becomes slower than Vanilla with FA (17.9 ms vs. 16.7 ms). In contrast, RAMP remains faster than Vanilla in both settings and achieves the lowest latency among all evaluated methods in the standard FA-enabled setting. This supports our deployment motivation: RAMP preserves the optimized FA path without materializing attention maps, enabling compression gains without sacrificing backend compatibility.

6 Conclusion

We introduce **RAMP**, a training-free audio token compression method designed for FlashAttention-compatible deployment of ASR-oriented ALMs. RAMP is motivated by a multi-axis representation analysis showing that ALM audio representations exhibit two distinct regimes: a locally redundant encoder regime and a dispersed post-projector regime. Based on this observation, RAMP applies bipartite L_2 merging in the encoder and centroid-distance-guided pruning after the projector, using only hidden states and projected tokens rather than attention maps. Across seven ASR benchmarks, three retention ratios, and two primary ALMs, RAMP achieves the lowest average WER multiplier among the evaluated training-free compression baselines. At the aggressive 40% token-retention setting on Audio-Flamingo-3, RAMP improves the average WER multiplier from 4.26× for the strongest non-RAMP baseline to 2.36×, while reducing TTFT by 24.8% over Vanilla and achieving the lowest TTFT among the compared methods. Together with the operator-placement ablation and FlashAttention latency analysis, these results suggest that ASR-oriented audio token compression benefits from matching compression operators to stage-wise representation regimes while preserving compatibility with standard FlashAttention-based deployment.

Limitations

Adaptation of vision-derived baselines. We adapt vision-derived baselines to audio tokens without audio-specific hyperparameter tuning or placement search. Our goal is to evaluate direct training-free transfer of their original token-scoring criteria under a shared token budget and temporal-order constraint, rather than to develop audio-specific variants of each baseline. Thus, the results should not be interpreted as the best achievable performance of these baselines after audio-specific adaptation.

Encoder architecture generalization. Our evaluation focuses on ALMs built on Whisper-style encoders. Conformer-based encoders introduce stronger locality and relative-position biases through convolutional modules and relative positional encoding. Because token compression changes the effective spacing and neighborhood structure of intermediate tokens, applying RAMP inside such encoders may require Conformer-aware extensions.

Decoder-preserving deployment. RAMP compresses audio tokens before LLM decoding and leaves the decoder unchanged, preserving compatibility with optimized serving backends such as vLLM. Decoder-internal compression could provide additional savings, but may require backend-specific modifications whose overhead can offset algorithmic gains. Integrating decoder-internal compression with modern serving backends remains an important direction for future work.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2026-25599056).

References

Saeed Ranjbar Alvar, Gursimran Singh, Mohammad Akbari, and Yong Zhang. 2025. DivPrune: Diversity-based visual token pruning for large multimodal models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9392–9401.

Rosana Ardila, Megan Branson, Kelly Davis, Michael Kohler, Josh Meyer, Michael Henretty, Reuben Morais, Lindsay Saunders, Francis Tyers, and Gregor Weber. 2020. Common Voice: A massively-multilingual speech corpus. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4218–4222, Marseille, France. European Language Resources Association.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. 2023. Token merging: Your ViT but faster. In *International Conference on Learning Representations*.

Xingyu Cai, Jiaji Huang, Yuchen Bian, and Kenneth Church. 2021. Isotropy in the contextual embedding space: Clusters and manifolds. In *International Conference on Learning Representations*.

Guoguo Chen, Shuzhou Chai, Guan-Bo Wang, Jiayu Du, Wei-Qiang Zhang, Chao Weng, Dan Su, Daniel Povey, Jan Trmal, Junbo Zhang, Mingjie Jin, Sanjeev Khudanpur, Shinji Watanabe, Shuaijiang Zhao, Wei Zou, Xiangang Li, Xuchen Yao, Yongqing Wang, Zhao You, and Zhiyong Yan. 2021. GigaSpeech: An evolving, multi-domain ASR corpus with 10,000 hours of transcribed audio. In *Interspeech 2021*, pages 3670–3674. ISCA.

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. 2024. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *Computer Vision – ECCV 2024*, pages 19–35, Cham. Springer Nature Switzerland.

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.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale N Fung, and Steven Hoi. 2023. InstructBLIP: Towards general-purpose vision-language models with instruction tuning. In *Advances in Neural Information Processing Systems*, volume 36, pages 49250–49267. Curran Associates, Inc.

Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems*, volume 35, pages 16344–16359. Curran Associates, Inc.

Soham Deshmukh, Benjamin Elizalde, Rita Singh, and Huaming Wang. 2023. Pengi: An audio language model for audio tasks. In *Advances in Neural Information Processing Systems*, volume 36, pages 18090–18108. Curran Associates, Inc.

Kawin Ethayarajh. 2019. How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 55–65, Hong Kong, China. Association for Computational Linguistics.

Elena Facco, Maria d'Errico, Alex Rodriguez, and Alessandro Laio. 2017. Estimating the intrinsic dimension of datasets by a minimal neighborhood information. *Scientific Reports*, 7(1):12140.

Fixie AI. 2025. Ultravox v0.6. https://huggingface.co/fixie-ai/ultravox-v0_6-llama-3_1-8b. Hugging Face model card. Accessed: 2026-05-25.

Sreyan Ghosh, Arushi Goel, Jaehyeon Kim, Sonal Kumar, Zhifeng Kong, Sang-gil Lee, Chao-Han Yang, Ramani Duraiswami, Dinesh Manocha, Rafael Valle, and Bryan Catanzaro. 2025. Audio Flamingo 3: Advancing audio intelligence with fully open large audio language models. In *Advances in Neural Information Processing Systems*, volume 38, pages 41819–41886. Curran Associates, Inc.

Sreyan Ghosh, Sonal Kumar, Ashish Seth, Chandra Kiran Reddy Evuru, Utkarsh Tyagi, S Sakshi, Oriol Nieto, Ramani Duraiswami, and Dinesh Manocha. 2024. GAMA: A large audio-language model with advanced audio understanding and complex reasoning abilities. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 6288–6313, Miami, Florida, USA. Association for Computational Linguistics.

Marcel Gibier, Pierre Serrano, Olivier Boeffard, Raphaël Duroselle, and Jean-François Bonastre. 2026. Segmentwise pruning in audio-language models. In *ICASSP 2026 - 2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 15322–15326.

Yuan Gong, Hongyin Luo, Alexander Liu, Leonid Karlinsky, and James R Glass. 2024. Listen, think, and understand. In *International Conference on Learning Representations*, volume 2024, pages 18516–18545.

François Hernandez, Vincent Nguyen, Sahar Ghannay, Natalia Tomashenko, and Yannick Estève. 2018. TED-LIUM 3: Twice as much data and corpus repartition for experiments on speaker adaptation. In *Speech and Computer*, pages 198–208, Cham. Springer International Publishing.

Yutao Jiang, Qiong Wu, Wenhao Lin, Wei Yu, and Yiyi Zhou. 2025. What kind of visual tokens do we need? training-free visual token pruning for multi-modal large language models from the perspective of graph. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(4):4075–4083.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pages 611–626. ACM.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023a. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 19730–19742. PMLR.

Yuang Li, Yu Wu, Jinyu Li, and Shujie Liu. 2023b. Accelerating transducers through adjacent token merging. In *Proceedings of Interspeech 2023*, pages 1379–1383.

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. In *2025 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. In *Advances in Neural Information Processing Systems*, volume 36, pages 34892–34916. Curran Associates, Inc.

Ke-Han Lu, Zhehuai Chen, Szu-Wei Fu, Chao-Han Huck Yang, Jagadeesh Balam, Boris Ginsburg, Yu-Chiang Frank Wang, and Hung-Yi Lee. 2025. Developing instruction-following speech language model without speech instruction-tuning data. In *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.

Yichen Lu, Jiaqi Song, Chao-Han Huck Yang, and Shinji Watanabe. 2024. FastAdaSP: Multitask-adapted efficient inference for large speech language model. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 440–451, Miami, Florida, US. Association for Computational Linguistics.

Qiankun Ma, Ziyao Zhang, Haofei Wang, Zhen Song, Jie Chen, and Hairong Zheng. 2026. ApET: Approximation-error guided token compression for efficient VLMs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 26306–26316.

Patrick K. O'Neill, Vitaly Lavrukhin, Somshubra Majumdar, Vahid Noroozi, Yuekai Zhang, Oleksii Kuchaiev, Jagadeesh Balam, Yuliya Dovzhenko, Keenan Freyberg, Michael D. Shulman, Boris Ginsburg, Shinji Watanabe, and Georg Kucsko. 2021. SPGISpeech: 5,000 Hours of Transcribed Financial Audio for Fully Formatted End-to-End Speech Recognition. In *Interspeech 2021*, pages 1434–1438.

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.

Anton Razzhigaev, Matvey Mikhalchuk, Elizaveta Goncharova, Ivan Oseledets, Denis Dimitrov, and Andrey Kuznetsov. 2024. The shape of learning: Anisotropy and intrinsic dimensions in transformer-based models. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 868–874, St. Julian's, Malta. Association for Computational Linguistics.

S Sakshi, Utkarsh Tyagi, Sonal Kumar, Ashish Seth, Ramaneswaran Selvakumar, Oriol Nieto, Ramani Duraiswami, Sreyan Ghosh, and Dinesh Manocha. 2025. MMAU: A massive multi-task audio understanding and reasoning benchmark. In *International Conference on Learning Representations*, volume 2025, pages 84929–84964.

Changli Tang, Wenyi Yu, Guangzhi Sun, Xianzhao Chen, Tian Tan, Wei Li, Lu Lu, Zejun Ma, and Chao Zhang. 2024. SALMONN: Towards generic hearing abilities for large language models. In *International Conference on Learning Representations*, volume 2024, pages 16607–16629.

Lucrezia Valeriani, Diego Doimo, Francesca Cuturello, Alessandro Laio, Alessio Ansuini, and Alberto Cazzaniga. 2023. The geometry of hidden representations of large transformer models. In *Advances in Neural Information Processing Systems*, volume 36, pages 51234–51252. Curran Associates, Inc.

Changhan Wang, Morgane Riviere, Ann Lee, Anne Wu, Chaitanya Talnikar, Daniel Haziza, Mary Williamson, Juan Pino, and Emmanuel Dupoux. 2021. 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, Online. Association for Computational Linguistics.

Zichen Wen, Yifeng Gao, Shaobo Wang, Junyuan Zhang, Qintong Zhang, Weijia Li, Conghui He, and Linfeng Zhang. 2025. Stop looking for "important tokens" in multimodal language models: Duplication matters more. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 9961–9980, Suzhou, China. Association for Computational Linguistics.

Bajian Xiang, Tingwei Guo, Xuan Chen, and Yang Han. 2026. Do we need distinct representations for every speech token? unveiling and exploiting redundancy in large speech language models. In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 15069–15087, San Diego, California, United States. Association for Computational Linguistics.

Jin Xu, Zhifang Guo, Jinzheng He, Hangrui Hu, Ting He, Shuai Bai, Keqin Chen, Jialin Wang, Yang Fan, Kai Dang, Bin Zhang, Xiong Wang, Yunfei Chu, and Junyang Lin. 2025. Qwen2.5-Omni technical report. *Preprint*, arXiv:2503.20215.

Qizhe Zhang, Mengzhen Liu, Lichen Li, Ming Lu, Yuan Zhang, Junwen Pan, Qi She, and Shanghang Zhang. 2025a. Beyond attention or similarity: Maximizing conditional diversity for token pruning in MLLMs. In *Advances in Neural Information Processing Systems*, volume 38, pages 25438–25468. Curran Associates, Inc.

Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis A Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, and Shanghang Zhang. 2025b. SparseVLM: Visual token sparsification for efficient vision-language model inference. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 74840–74857. PMLR.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. 2024. MiniGPT-4: Enhancing vision-language understanding with advanced large language models. In *International Conference on Learning Representations*, volume 2024, pages 18378–18394.

A Generalization across Token Rate and LLM Family

To evaluate whether RAMP's gains persist beyond the two primary 25 Hz ALMs evaluated in Section 4.2, we additionally evaluate Ultravox v0.6 Llama-3.1-8B. Unlike Audio-Flamingo-3 and Qwen2.5-Omni-7B, Ultravox v0.6 Llama-3.1-8B operates at a substantially lower 6.25 Hz audio token rate and uses a different LLM family. This setting changes both the token rate and the LLM backbone, allowing us to test whether the representation regimes identified in Section 3.3, and the resulting merge–prune design, remain effective under these architectural differences.

Figure 3 extends the layer-wise representation analysis in Section 3.3 to Ultravox v0.6 Llama-3.1-8B. The same qualitative encoder-to-projector shift is observed even at 6.25 Hz: encoder representations remain relatively locally smooth and directionally concentrated, whereas post-projector tokens become more dispersed in the LLM-aligned space. This suggests that the two-regime structure motivating encoder-side merging followed by post-projector pruning may generalize beyond Audio-Flamingo-3 and Qwen2.5-Omni-7B.

Table 6 reports ASR results on Ultravox v0.6 Llama-3.1-8B. Because Ultravox v0.6 Llama-3.1-8B already operates at a lower 6.25 Hz token rate, its projected audio tokens are temporally coarser than those of the 25 Hz models. This makes the post-projector token budget more fragile, since pruning after projection removes tokens from an already downsampled sequence. We therefore use an encoder-heavy budget split, $\beta = 0.8$, which shifts more reduction to the denser pre-projector encoder sequence and reduces reliance on post-projector pruning. Under this low-rate setting,

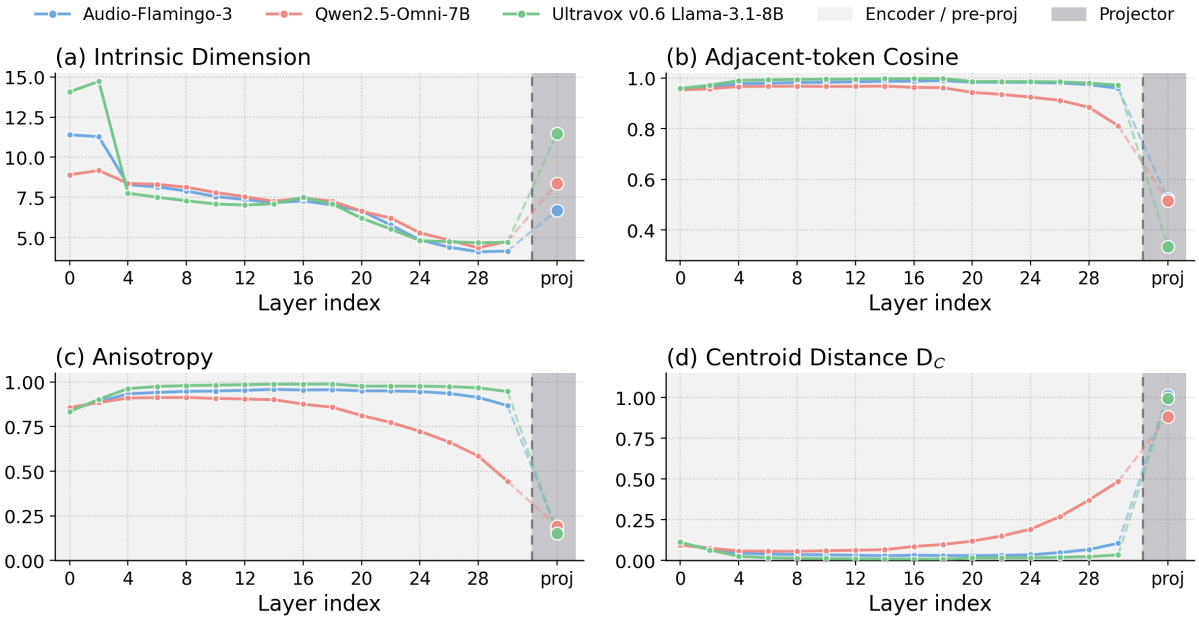


Figure 3: Layer-wise representation analysis for Audio-Flamingo-3, Qwen2.5-Omni-7B, and Ultravox v0.6 Llama-3.1-8B on the full LibriSpeech *test-clean* set. We report (a) intrinsic dimension, (b) adjacent-token cosine similarity, (c) anisotropy, and (d) p_{95} of D_C across audio encoder layers and the projector output.

Method	LS-clean	LS-other	SPGI	TED	Giga	CV	VP	Avg. × ↓
Vanilla	2.23	4.44	3.66	3.13	9.91	9.89	8.69	—
Retention ratio $r = 0.6$								
A-ToMe Interspeech 23	7.97	11.88	8.08	9.66	17.21	13.61	12.82	2.30×
DART EMNLP 25	32.85	34.59	26.27	31.65	35.31	26.32	37.89	7.20×
CDPruner NeurIPS 25	35.42	36.75	32.50	32.30	39.06	30.60	38.17	7.83×
DivPrune CVPR 25	33.28	34.45	28.43	32.35	36.62	22.12	37.79	7.29×
G-Prune AAAI 25	86.70	92.54	74.06	77.31	77.44	101.22	75.79	18.77×
ApET CVPR 26	30.29	33.89	27.94	29.98	36.62	26.92	41.37	7.09×
RAMP	**5.89**	**8.69**	**5.78**	**6.28**	**15.40**	**11.84**	**11.15**	**1.75×**
Retention ratio $r = 0.5$								
A-ToMe Interspeech 23	17.20	20.40	15.65	18.13	26.00	19.88	19.61	4.18×
DART EMNLP 25	42.06	43.41	33.62	39.08	42.76	34.14	45.08	9.04×
CDPruner NeurIPS 25	45.08	45.88	42.52	42.08	48.37	38.41	47.03	9.97×
DivPrune CVPR 25	41.99	43.24	34.42	39.89	43.21	29.44	44.81	9.03×
G-Prune AAAI 25	102.01	101.19	84.52	83.21	85.83	113.71	83.72	21.14×
ApET CVPR 26	43.63	44.87	37.77	39.27	46.74	39.71	49.60	9.57×
RAMP	**10.76**	**13.90**	**10.00**	**12.16**	**21.77**	**14.91**	**15.68**	**2.87×**
Retention ratio $r = 0.4$								
A-ToMe Interspeech 23	33.52	37.57	30.46	33.50	41.14	34.19	34.73	7.73×
DART EMNLP 25	51.19	53.53	43.31	47.85	51.71	44.08	52.91	11.13×
CDPruner NeurIPS 25	55.46	56.47	53.87	53.16	58.49	49.85	56.54	12.39×
DivPrune CVPR 25	50.60	51.67	41.85	47.44	50.53	36.99	52.52	10.83×
G-Prune AAAI 25	110.99	103.96	90.72	87.12	90.81	122.80	87.93	22.50×
ApET CVPR 26	54.95	60.04	49.33	49.20	57.69	53.88	58.22	12.19×
RAMP	**24.00**	**25.39**	**21.21**	**23.68**	**35.01**	**22.74**	**26.65**	**5.53×**

Table 6: Generalization results on Ultravox v0.6 Llama-3.1-8B. Because Ultravox v0.6 Llama-3.1-8B operates at a lower 6.25 Hz post-projector token rate, RAMP is evaluated with the encoder-heavy $\beta = 0.8$ setting. Benchmark columns report WER (%), and Avg. × denotes the mean of dataset-level WER ratios relative to Vanilla across seven benchmarks (lower is better). **Bold** and underline indicate the best and second-best compression methods within each retention-ratio group, respectively.

β	LS-clean	LS-other	SPGI	TED	Giga	CV	VP	Avg. $\times\downarrow$
Vanilla	1.51	3.03	1.73	3.31	9.90	8.20	5.70	—
Retention ratio $r = 0.6$								
$\beta = 0.2$	**1.78**	3.59	1.96	4.03	10.42	<u>7.47</u>	<u>6.03</u>	<u>1.11×</u>
$\beta = 0.4$	1.81	<u>3.53</u>	**1.93**	<u>3.93</u>	**10.34**	**7.45**	**5.97**	**1.10×**
$\beta = 0.6$	<u>1.80</u>	**3.45**	<u>1.95</u>	3.94	<u>10.38</u>	7.73	6.10	**1.10×**
$\beta = 0.8$	1.83	3.55	1.98	**3.89**	10.53	7.79	6.15	<u>1.11×</u>
Retention ratio $r = 0.5$								
$\beta = 0.2$	<u>2.57</u>	4.86	2.72	5.45	12.51	8.17	**6.54**	1.42×
$\beta = 0.4$	**2.52**	**4.51**	**2.58**	<u>5.17</u>	**12.34**	<u>8.09</u>	<u>6.67</u>	**1.37×**
$\beta = 0.6$	2.63	<u>4.66</u>	<u>2.60</u>	**4.98**	<u>12.48</u>	**7.98**	6.70	<u>1.39×</u>
$\beta = 0.8$	2.66	4.84	2.81	5.45	13.20	<u>8.09</u>	6.74	1.45×
Retention ratio $r = 0.4$								
$\beta = 0.2$	<u>5.49</u>	<u>8.69</u>	5.78	9.34	<u>17.98</u>	10.16	**8.43**	<u>2.46×</u>
$\beta = 0.4$	**5.36**	**8.44**	**5.36**	**8.88**	**17.83**	**9.47**	<u>8.44</u>	**2.36×**
$\beta = 0.6$	5.60	8.94	<u>5.51</u>	<u>9.32</u>	18.62	9.66	8.75	<u>2.46×</u>
$\beta = 0.8$	6.32	9.29	6.21	9.82	19.64	<u>9.63</u>	9.08	2.65×

Table 7: Sensitivity to the β budget-split parameter of RAMP on Audio-Flamingo-3. Benchmark columns report WER (%), and Avg. $\times$ denotes the mean of dataset-level WER ratios relative to Vanilla across seven benchmarks (lower is better). **Bold** and <u>underline</u> indicate the best and second-best β within each retention-ratio group, respectively; the highlighted row marks the fixed $\beta = 0.4$ setting used in the main results.

β	LS-clean	LS-other	SPGI	TED	Giga	CV	VP	Avg. $\times\downarrow$
Vanilla	1.67	3.27	2.22	3.40	9.89	7.69	6.00	—
Retention ratio $r = 0.6$								
$\beta = 0.2$	<u>2.19</u>	<u>4.36</u>	**2.77**	**4.23**	<u>11.06</u>	9.20	<u>6.87</u>	<u>1.23×</u>
$\beta = 0.4$	**2.15**	**4.26**	<u>2.80</u>	4.38	**11.02**	**9.00**	**6.86**	**1.22×**
$\beta = 0.6$	2.28	**4.26**	2.91	<u>4.37</u>	11.48	<u>9.04</u>	7.00	1.25×
$\beta = 0.8$	2.48	4.61	3.12	4.57	11.63	9.27	7.22	1.32×
Retention ratio $r = 0.5$								
$\beta = 0.2$	**3.70**	**6.66**	**4.36**	**5.68**	**13.17**	12.47	**8.29**	**1.75×**
$\beta = 0.4$	<u>3.80</u>	<u>6.69</u>	<u>4.53</u>	<u>5.87</u>	<u>13.45</u>	**11.67**	<u>8.42</u>	<u>1.77×</u>
$\beta = 0.6$	4.17	7.14	4.97	6.27	13.93	<u>11.85</u>	8.73	1.88×
$\beta = 0.8$	4.98	7.63	5.70	6.80	14.62	12.26	9.37	2.07×
Retention ratio $r = 0.4$								
$\beta = 0.2$	**9.95**	<u>14.75</u>	**11.09**	<u>11.37</u>	**19.82**	21.06	**13.75**	<u>3.69×</u>
$\beta = 0.4$	<u>10.04</u>	**13.80**	<u>11.33</u>	**11.27**	<u>20.19</u>	**19.66**	<u>13.79</u>	**3.65×**
$\beta = 0.6$	11.98	15.68	12.67	12.67	21.61	<u>19.71</u>	14.46	4.08×
$\beta = 0.8$	13.71	17.40	14.44	13.63	23.60	20.29	16.20	4.54×

Table 8: Sensitivity to the β budget-split parameter of RAMP on Qwen2.5-Omni-7B. Benchmark columns report WER (%), and Avg. $\times$ denotes the mean of dataset-level WER ratios relative to Vanilla across seven benchmarks (lower is better). **Bold** and <u>underline</u> indicate the best and second-best β within each retention-ratio group, respectively; the highlighted row marks the fixed $\beta = 0.4$ setting used in the main results.

RAMP achieves the lowest Avg. $\times$ across all retention ratios. At $r = 0.6$, 0.5, and 0.4, RAMP achieves Avg. $\times$ values of $1.75\times$, $2.87\times$, and $5.53\times$, respectively, compared with $2.30\times$, $4.18\times$, and $7.73\times$ for the strongest non-RAMP baseline, A-ToMe. The gap between RAMP and A-ToMe widens as the retention ratio decreases, indicating that RAMP remains more stable under tighter compression budgets. Together, these results show that RAMP remains the strongest compression method even in a lower-rate ALM with a different LLM family when using a token-rate-aware budget split.

β	LS-clean	LS-other	SPGI	TED	Giga	CV	VP	Avg. $\times\downarrow$
Vanilla	2.23	4.44	3.66	3.13	9.91	9.89	8.69	—
Retention ratio $r = 0.6$								
$\beta = 0.2$	9.52	12.72	9.93	10.94	19.37	12.68	13.44	2.59×
$\beta = 0.4$	7.12	9.78	7.79	8.66	17.69	12.58	12.21	2.11×
$\beta = 0.6$	**5.89**	**7.85**	6.00	6.92	15.84	12.34	11.84	1.78×
$\beta = 0.8$	**5.89**	8.69	**5.78**	**6.28**	**15.40**	**11.84**	**11.15**	**1.75×**
Retention ratio $r = 0.5$								
$\beta = 0.2$	19.35	22.82	18.26	21.09	28.47	17.69	19.63	4.64×
$\beta = 0.4$	17.60	19.45	16.73	20.19	27.12	17.05	18.79	4.27×
$\beta = 0.6$	14.14	16.62	13.16	14.55	24.14	15.80	18.16	3.49×
$\beta = 0.8$	**10.76**	**13.90**	**10.00**	**12.16**	**21.77**	**14.91**	**15.68**	**2.87×**
Retention ratio $r = 0.4$								
$\beta = 0.2$	31.15	35.02	30.35	33.20	41.07	25.74	30.48	7.29×
$\beta = 0.4$	33.96	34.65	30.51	34.14	40.41	26.93	30.69	7.51×
$\beta = 0.6$	32.37	33.83	27.86	29.18	38.54	24.80	31.65	7.01×
$\beta = 0.8$	**24.00**	**25.39**	**21.21**	**23.68**	**35.01**	**22.74**	**26.65**	**5.53×**

Table 9: Sensitivity to the β budget-split parameter of RAMP on Ultravox v0.6 Llama-3.1-8B. Benchmark columns report WER (%), and Avg. $\times$ denotes the mean of dataset-level WER ratios relative to Vanilla across seven benchmarks (lower is better). **Bold** and underline indicate the best and second-best β within each retention-ratio group, respectively; the highlighted row marks the encoder-heavy $\beta = 0.8$ setting used in the generalization results.

B Analysis of Budget Split β

RAMP uses the budget-split parameter β to allocate the total token-reduction budget between encoder-side merging and post-projector pruning for a fixed final retention ratio. A larger β shifts more reduction to the encoder stage, while a smaller β shifts more reduction to the post-projector stage. We sweep $\beta \in \{0.2, 0.4, 0.6, 0.8\}$ to examine how this allocation behaves across model architectures and audio token rates.

Tables 7 and 8 report the results for the two primary 25 Hz ALMs. On Audio-Flamingo-3, RAMP performs strongly across budget splits, with $\beta = 0.4$ giving the best or tied-best Avg. $\times$ across all retention ratios. On Qwen2.5-Omni-7B, $\beta = 0.2$ and $\beta = 0.4$ are consistently the two strongest settings. We therefore use $\beta = 0.4$ as a single shared setting for the two 25 Hz ALMs, providing a strong common choice without per-dataset or per-ratio tuning. Table 9 reports the corresponding sweep for Ultravox v0.6 Llama-3.1-8B. Unlike the 25 Hz ALMs, Ultravox v0.6 Llama-3.1-8B operates at a substantially lower 6.25 Hz post-projector token rate, making its projected audio sequence temporally coarser. In this setting, assigning more reduction to the denser encoder sequence is more effective, and $\beta = 0.8$ achieves the lowest Avg. $\times$ at every evaluated retention ratio.

Across the evaluated models, these results suggest a practical allocation principle based on post-projector token rate. The two 25 Hz models, corresponding to approximately 40 ms per projected token, favor moderate encoder-side allocation, whereas the 6.25 Hz model, corresponding to approximately 160 ms per projected token, favors the encoder-heavy $\beta = 0.8$ setting. As the projected sequence becomes temporally coarser, more of the compression budget should be shifted to the denser encoder regime.

C Empirical Validation of Merge and Prune Criteria

Table 10 provides empirical support for the metric choices used in the two stages of RAMP on Audio-Flamingo-3. Each variant modifies only one stage while keeping the other stage fixed to the RAMP default, allowing us to isolate the effect of the encoder-side merge metric and the post-projector selection score.

For encoder-side merging, L_2-based matching is consistently more effective than cosine-based alternatives. Non-adjacent cosine matching performs worse, especially under stronger compression, indicating that unrestricted cosine similarity does not provide a reliable merge signal in the encoder. Adjacent cosine improves over non-adjacent

Stage	Variant	LS-clean	LS-other	SPGI	TED	Giga	CV	VP	Avg. × ↓
	Vanilla	1.51	3.03	1.73	3.31	9.90	8.20	5.70	—
	Retention ratio $r = 0.6$								
Merge	Non-adj. cosine	2.33	4.24	2.42	4.83	11.65	7.82	6.50	1.30×
Merge	Adjacent L_2	**1.78**	**3.47**	1.96	4.04	10.39	7.70	6.07	**1.10**×
Merge	Adjacent cosine	1.88	3.79	2.09	4.29	10.80	7.69	6.32	1.16×
Prune	Low-D_C	7.93	10.40	5.00	9.55	17.58	15.09	16.02	2.98×
Prune	Random	8.07	9.93	6.51	10.12	17.30	10.66	10.81	2.91×
Prune	Uniform	2.72	4.77	2.50	5.00	11.86	8.59	7.45	1.41×
	RAMP	1.81	3.53	**1.93**	**3.93**	**10.34**	**7.45**	**5.97**	**1.10**×
	Retention ratio $r = 0.5$								
Merge	Non-adj. cosine	5.52	8.17	4.85	8.63	16.32	9.32	8.51	2.29×
Merge	Adjacent L_2	2.59	4.65	2.66	5.18	12.75	8.15	**6.56**	1.40×
Merge	Adjacent cosine	3.01	5.48	3.17	6.37	13.90	8.85	7.24	1.62×
Prune	Low-D_C	20.71	22.53	12.86	21.82	29.14	22.78	30.96	6.62×
Prune	Random	17.07	19.04	13.62	18.35	25.96	15.60	17.96	5.53×
Prune	Uniform	6.00	8.52	4.45	8.24	16.53	10.26	10.12	2.36×
	RAMP	**2.52**	**4.51**	**2.58**	**5.17**	**12.34**	**8.09**	6.67	**1.37**×
	Retention ratio $r = 0.4$								
Merge	Non-adj. cosine	15.47	18.88	12.66	18.07	27.65	13.15	14.15	5.16×
Merge	Adjacent L_2	5.72	**8.44**	5.60	9.38	18.22	9.71	8.49	2.45×
Merge	Adjacent cosine	7.43	10.97	7.30	11.55	20.75	11.66	9.96	3.07×
Prune	Low-D_C	44.39	44.49	31.66	41.62	49.18	39.35	54.97	13.48×
Prune	Random	32.35	34.07	26.09	32.79	39.33	26.19	31.67	10.05×
Prune	Uniform	16.06	19.35	12.13	18.22	28.34	14.75	17.63	5.33×
	RAMP	**5.36**	**8.44**	**5.36**	**8.88**	**17.83**	**9.47**	**8.44**	**2.36**×

Table 10: Empirical validation of merge metrics and pruning scores on Audio-Flamingo-3. Each variant changes one stage while keeping the other stage fixed to the RAMP default. For **Merge** variants, post-projector pruning is fixed to high-D_C selection with the temporal penalty; for **Prune** variants, encoder-side merging is fixed to L_2 matching. Benchmark columns report WER (%), and Avg. × denotes the mean of dataset-level WER ratios relative to Vanilla across seven benchmarks (lower is better). **Bold** and underline indicate the best and second-best variant within each retention-ratio group, respectively.

cosine, but remains consistently worse than L_2-based matching. Moreover, adjacent L_2 remains close to RAMP, suggesting that the key factor is the L_2 merge criterion itself, while a strict adjacency constraint is not necessary. This supports our choice of non-adjacent L_2 matching: in the locally smooth encoder regime, cosine similarity is highly saturated, whereas L_2 retains magnitude-sensitive variation that better distinguishes redundant merge candidates.

For post-projector pruning, D_C-based selection is more effective than the alternatives. Selecting low-D_C tokens performs worse, showing that centroid-near tokens alone do not preserve the dispersed post-projector representation. Random and uniform selection are also consistently worse than RAMP, especially at lower retention ratios, indicating that the pruning stage benefits from incorporating geometry-aware token-level scores.

Overall, these results support the stage-specific metric choices used by RAMP. Encoder-side merging benefits from L_2-based redundancy estimation, while post-projector pruning benefits from using D_C as a geometry-aware selection signal in the dispersed LLM-aligned token space.

D Extended Operator Placement Ablations

Tables 11, 12, and 13 provide extended operator placement results across all benchmarks, retention ratios, and evaluated ALMs. They complement the operator placement ablation in Section 5.1 by showing whether the same placement principle holds beyond the main model and compression setting.

On Audio-Flamingo-3, the one-stage variants remain relatively close to RAMP, especially under moderate compression. Encoder-only merging performs well at $r = 0.6$ and $r = 0.5$, while post-projector-only pruning becomes slightly stronger

Placement	LS-clean	LS-other	SPGI	TED	Giga	CV	VP	Avg. × ↓
Vanilla	1.51	3.03	1.73	3.31	9.90	8.20	5.70	—
Retention ratio $r = 0.6$								
Encoder merge only	**1.80**	3.57	1.95	4.02	10.65	7.88	6.10	1.12×
Post-projector prune only	1.83	3.69	2.07	4.11	10.93	7.95	6.05	1.14×
Merge → Merge	7.84	10.81	4.70	9.28	17.84	12.56	11.89	2.81×
Prune → Prune	2.39	4.53	2.90	5.09	12.27	8.21	6.47	1.38×
Prune → Merge (Swap)	15.63	18.86	8.74	14.31	23.09	18.52	17.04	4.79×
Merge → Prune (RAMP)	1.81	**3.53**	**1.93**	**3.93**	**10.34**	**7.45**	**5.97**	**1.10×**
Retention ratio $r = 0.5$								
Encoder merge only	2.81	5.00	2.68	5.31	12.90	8.27	7.11	1.46×
Post-projector prune only	2.82	5.35	3.20	5.96	13.24	8.94	6.80	1.56×
Merge → Merge	18.35	21.18	10.92	17.36	27.08	17.47	18.90	5.55×
Prune → Prune	4.41	7.23	5.22	8.08	15.88	9.65	7.78	2.13×
Prune → Merge (Swap)	31.17	33.57	18.24	25.30	34.79	28.58	28.13	8.83×
Merge → Prune (RAMP)	**2.52**	**4.51**	**2.58**	**5.17**	**12.34**	**8.09**	**6.67**	**1.37×**
Retention ratio $r = 0.4$								
Encoder merge only	7.18	10.17	6.18	9.76	19.11	10.88	10.37	2.82×
Post-projector prune only	5.97	10.22	7.27	10.42	19.43	11.22	8.69	2.79×
Merge → Merge	32.11	34.54	22.75	29.84	39.70	23.07	29.33	9.54×
Prune → Prune	9.70	14.46	11.34	15.60	24.08	13.02	11.71	4.08×
Prune → Merge (Swap)	44.03	46.88	33.42	40.14	47.92	37.12	38.72	13.18×
Merge → Prune (RAMP)	**5.36**	**8.44**	**5.36**	**8.88**	**17.83**	**9.47**	**8.44**	**2.36×**

Table 11: Extended operator placement ablation on Audio-Flamingo-3 across seven benchmarks. For two-stage variants, placement is written in encoder → post-projector order. Benchmark columns report WER (%), and Avg. × denotes the mean of dataset-level WER ratios relative to Vanilla across seven benchmarks (lower is better). **Bold** and underline indicate the best and second-best placement within each retention-ratio group, respectively.

Placement	LS-clean	LS-other	SPGI	TED	Giga	CV	VP	Avg. × ↓
Vanilla	1.67	3.27	2.22	3.40	9.89	7.69	6.00	—
Retention ratio $r = 0.6$								
Encoder merge only	2.37	4.48	2.99	4.45	11.57	9.47	7.43	1.30×
Post-projector prune only	2.23	4.54	2.84	**4.32**	11.26	9.49	**6.80**	1.25×
Merge → Merge	13.96	16.37	8.27	11.91	22.35	17.21	11.17	3.85×
Prune → Prune	3.65	6.03	3.85	5.49	13.07	9.81	7.26	1.60×
Prune → Merge (Swap)	17.96	21.52	10.81	14.64	25.71	20.89	13.25	4.86×
Merge → Prune (RAMP)	**2.15**	**4.26**	**2.80**	4.38	**11.02**	**9.00**	6.86	**1.22×**
Retention ratio $r = 0.5$								
Encoder merge only	4.38	7.31	5.25	6.72	14.84	12.69	9.89	2.00×
Post-projector prune only	4.01	7.16	4.69	5.97	**13.32**	13.10	8.62	1.85×
Merge → Merge	27.22	29.83	18.30	23.37	34.97	29.92	19.89	7.33×
Prune → Prune	8.78	12.09	8.12	9.89	18.21	13.84	10.13	2.98×
Prune → Merge (Swap)	35.80	40.03	23.92	28.91	41.55	37.05	24.87	9.45×
Merge → Prune (RAMP)	**3.80**	**6.69**	**4.53**	**5.87**	13.45	**11.67**	**8.42**	**1.77×**
Retention ratio $r = 0.4$								
Encoder merge only	12.51	17.15	13.71	13.82	23.61	21.68	17.70	4.45×
Post-projector prune only	11.03	15.64	12.30	11.77	20.41	23.05	15.14	4.00×
Merge → Merge	46.38	49.63	35.81	41.05	51.49	43.24	36.95	12.59×
Prune → Prune	23.85	27.85	20.65	21.87	31.25	25.86	19.50	6.90×
Prune → Merge (Swap)	57.70	61.89	44.97	52.38	60.77	51.15	42.77	15.58×
Merge → Prune (RAMP)	**10.04**	**13.80**	**11.33**	**11.27**	**20.19**	**19.66**	**13.79**	**3.65×**

Table 12: Extended operator placement ablation on Qwen2.5-Omni-7B across seven benchmarks. For two-stage variants, placement is written in encoder → post-projector order. Benchmark columns report WER (%), and Avg. × denotes the mean of dataset-level WER ratios relative to Vanilla across seven benchmarks (lower is better). **Bold** and underline indicate the best and second-best placement within each retention-ratio group, respectively.

Placement	LS-clean	LS-other	SPGI	TED	Giga	CV	VP	Avg. × ↓
Vanilla	2.23	4.44	3.66	3.13	9.91	9.89	8.69	—
Retention ratio $r = 0.6$								
Encoder merge only	8.87	11.76	8.75	9.93	18.20	13.70	12.99	2.41×
Post-projector prune only	23.53	25.78	17.91	22.11	27.52	17.89	28.60	5.17×
Merge → Merge	22.30	25.10	21.48	22.07	28.74	29.16	24.12	5.31×
Prune → Prune	6.62	9.44	6.92	9.00	16.25	13.00	11.24	2.02×
Prune → Merge (Swap)	23.20	28.11	22.11	23.58	29.97	31.13	23.28	5.59×
Merge → Prune (RAMP)	**5.89**	**8.69**	**5.78**	**6.28**	**15.40**	**11.84**	**11.15**	**1.75×**
Retention ratio $r = 0.5$								
Encoder merge only	18.32	21.58	16.72	18.50	27.08	20.03	20.13	4.37×
Post-projector prune only	30.97	33.66	26.00	29.88	35.03	23.27	33.54	6.84×
Merge → Merge	35.83	38.64	34.34	34.54	40.69	45.16	38.50	8.33×
Prune → Prune	11.95	14.82	11.39	14.00	23.11	16.21	**15.28**	3.14×
Prune → Merge (Swap)	37.54	42.85	36.11	36.58	42.73	49.52	38.19	8.82×
Merge → Prune (RAMP)	**10.76**	**13.90**	**10.00**	**12.16**	**21.77**	**14.91**	15.68	**2.87×**
Retention ratio $r = 0.4$								
Encoder merge only	35.81	38.51	31.68	32.70	42.14	33.97	33.89	7.92×
Post-projector prune only	42.95	46.10	38.80	40.88	46.63	35.26	43.05	9.50×
Merge → Merge	52.14	55.43	50.33	49.53	54.98	62.69	52.51	11.91×
Prune → Prune	25.56	30.37	24.62	27.87	35.60	26.39	27.66	6.20×
Prune → Merge (Swap)	54.18	59.95	52.96	52.08	56.82	67.38	56.64	12.57×
Merge → Prune (RAMP)	**24.00**	**25.39**	**21.21**	**23.68**	**35.01**	**22.74**	**26.65**	**5.53×**

Table 13: Extended operator placement ablation on Ultravox v0.6 Llama-3.1-8B across seven benchmarks. For two-stage variants, placement is written in encoder → post-projector order. Benchmark columns report WER (%), and Avg. × denotes the mean of dataset-level WER ratios relative to Vanilla across seven benchmarks (lower is better). **Bold** and underline indicate the best and second-best placement within each retention-ratio group, respectively.

at $r = 0.4$. This suggests that both the encoder and post-projector stages contain exploitable structure, but using only one side of this structure provides only partial gains. RAMP achieves the lowest Avg. × at all retention ratios by combining the two stage-specific operators in the matched order.

On Qwen2.5-Omni-7B, post-projector-only pruning remains competitive across retention ratios. This indicates that the projected token space provides useful selection signals for this model. However, pruning alone does not exploit the locally redundant encoder regime, and its gap from RAMP becomes larger under stronger compression. RAMP again achieves the lowest Avg. × at every retention ratio by first reducing encoder-side redundancy through merging and then selecting projected tokens after the projector.

On Ultravox v0.6 Llama-3.1-8B, the pattern differs from the two 25 Hz ALMs. Because Ultravox v0.6 Llama-3.1-8B operates at a lower 6.25 Hz token rate, Prune → Prune becomes relatively competitive compared with the one-stage variants. This suggests that the preferred secondary placement can depend on token rate and model architecture. Nevertheless, Prune → Prune remains worse than RAMP, showing that pruning-heavy placement alone does not fully replace the benefit of combining encoder-side merging with post-projector pruning.

Overall, the best non-RAMP placement varies across models, but RAMP remains consistently best across all evaluated ALMs and retention ratios. These results support the central placement principle: the benefit comes not from using two stages alone, but from matching each compression operator to the representation regime where it is most appropriate.

E Beyond ASR: Audio Understanding Evaluation

To examine whether RAMP preserves performance beyond ASR-focused benchmarks, we evaluate Audio-Flamingo-3 on MMAU and MMAU-mini (Sakshi et al., 2025), which cover broader audio understanding across Sound, Music, and Speech categories. Table 14 shows that RAMP remains robust beyond transcription-focused evaluation. On the full MMAU benchmark, RAMP achieves the highest Avg. score among all com-

Method		MMAU				MMAU-mini			
		Sound	Music	Speech	Avg. ↑	Sound	Music	Speech	Avg. ↑
Vanilla		79.83	75.23	67.00	74.02	82.28	75.75	68.17	75.40
Retention ratio $r = 0.6$									
A-ToMe	Interspeech 23	80.17	75.07	66.43	73.89	81.98	74.85	67.87	74.90
DART	EMNLP 25	**80.50**	75.17	67.10	74.26	81.68	74.25	**69.97**	75.30
CDPruner	NeurIPS 25	80.00	75.33	65.60	73.64	**82.58**	74.55	66.37	74.50
DivPrune	CVPR 25	80.13	75.33	66.33	73.93	81.38	**76.65**	67.87	75.30
G-Prune	AAAI 25	75.83	70.63	57.47	67.98	80.48	73.95	58.86	71.10
ApET	CVPR 26	79.20	74.60	56.33	70.04	81.08	75.45	59.76	72.10
RAMP		79.60	**75.67**	**67.77**	**74.34**	81.68	76.35	68.47	**75.50**
Retention ratio $r = 0.5$									
A-ToMe	Interspeech 23	80.03	75.40	66.57	74.00	82.28	75.15	67.57	75.00
DART	EMNLP 25	**80.07**	75.00	66.37	73.81	**82.58**	76.05	**69.07**	**75.90**
CDPruner	NeurIPS 25	80.00	**75.43**	65.00	73.48	81.68	75.15	65.17	74.00
DivPrune	CVPR 25	79.67	**75.43**	65.80	73.63	81.38	**76.65**	67.57	75.20
G-Prune	AAAI 25	74.07	70.17	55.17	66.47	78.68	73.35	57.36	69.80
ApET	CVPR 26	79.10	74.37	54.20	69.22	81.98	74.85	54.65	70.50
RAMP		79.67	74.97	**67.40**	**74.01**	81.08	75.45	68.47	75.00
Retention ratio $r = 0.4$									
A-ToMe	Interspeech 23	79.77	75.47	66.27	73.83	**82.28**	75.75	67.27	75.10
DART	EMNLP 25	79.43	75.40	65.80	73.54	81.68	75.45	66.07	74.40
CDPruner	NeurIPS 25	**79.83**	75.00	63.17	72.67	81.08	76.05	63.36	73.50
DivPrune	CVPR 25	79.40	**75.57**	64.80	73.26	**82.28**	**76.65**	68.17	**75.70**
G-Prune	AAAI 25	72.77	70.00	53.63	65.47	77.18	71.26	55.26	67.90
ApET	CVPR 26	79.23	74.57	52.23	68.68	81.68	74.85	53.45	70.00
RAMP		79.37	75.17	**67.30**	**73.94**	79.88	76.05	**69.67**	75.20

Table 14: MMAU and MMAU-mini results on Audio-Flamingo-3. For each benchmark, task columns report accuracy (%) on the Sound, Music, and Speech categories, and Avg. denotes the overall accuracy (higher is better). **Bold** and underline indicate the best and second-best compression methods within each retention-ratio group, respectively.

pression methods at every retention ratio, slightly exceeding Vanilla at $r = 0.6$ (74.34 vs. 74.02) and remaining within 0.08 points even at $r = 0.4$ (73.94 vs. 74.02). On MMAU-mini, RAMP remains competitive with Vanilla and the strongest compression baselines, achieving Avg. scores of 75.50, 75.00, and 75.20 compared with 75.40 for Vanilla. These results provide complementary evidence that, while our primary focus remains ASR-oriented compression, RAMP largely preserves broader audio-understanding performance under compression.

F Failure Analysis

We analyze compression-induced catastrophic regressions for Audio-Flamingo-3 at retention ratio $r = 0.4$. A sample is counted as a catastrophic failure under Vanilla if the Vanilla output has at least 10 word errors and a sample-level WER of at least 1.0. A sample is counted as a compression-induced catastrophic regression if it is not catastrophic under Vanilla, but the compressed output has a sample-level WER of at least 1.0 and at least twice the Vanilla sample-level WER. This relative criterion focuses the analysis on additional degradation introduced by compression, rather than failures already present under Vanilla. Each regression is assigned one primary tag using the rule-based taxonomy in Table 15. When multiple criteria match, we assign the primary tag according to a fixed priority order. This primary-tag analysis summarizes dominant failure modes rather than providing an exhaustive multi-label annotation of every error.

F.1 Regression Counts and Failure Modes

Table 16 reports the number and primary-tag breakdown of compression-induced catastrophic regressions, summed over all seven ASR benchmarks. RAMP produces the fewest regressions, with 82 cases. By contrast, DivPrune, A-ToMe, CDPruner, and DART produce 179–508 regressions, while G-Prune and ApET exceed 4,900 regressions.

Among RAMP's remaining regressions, 73 of 82 cases (89.0%) are assigned to `short_ref`. A simi-

Tag	Meaning
`refusal`	Empty output or very short label-like non-transcriptional output.
`repetition`	Degenerate repeated-token or repeated-phrase looping output.
`short_ref`	Low-recall error on a short reference with at most six words, after excluding refusal, repetition, and long ungrounded outputs.
`other`	Remaining failures, including ungrounded long outputs, truncation, length blow-up, numeric mismatch, proper-noun hallucination, or residual substitutions.

Table 15: Failure taxonomy for compression-induced catastrophic regressions. Tags are applied in the listed precedence order.

Method	Total	Ref.	Rep.	Short	Other
RAMP	82	1	1	73	7
DivPrune	179	4	1	141	33
A-ToMe	198	6	2	167	23
CDPruner	218	13	4	169	32
DART	508	67	3	410	28
G-Prune	4,941	98	139	2,216	2,488
ApET	5,064	78	263	1,857	2,866

Table 16: Compression-induced catastrophic regressions on Audio-Flamingo-3 at $r = 0.4$, summed over seven ASR benchmarks. Columns report the total count and primary-tag breakdown; Ref., Rep., and Short denote `refusal`, `repetition`, and `short_ref`, respectively.

lar `short_ref`-dominant pattern appears among baselines with relatively low regression counts, including A-ToMe (84.3%), DART (80.7%), DivPrune (78.8%), and CDPruner (77.5%). This indicates that many remaining severe regressions occur on short references, where one or two word errors can already yield high sample-level WER. In contrast, G-Prune and ApET have lower `short_ref` shares of 44.8% and 36.7%, respectively, because more of their failures fall into `other`. This residual category includes ungrounded long outputs, unstable length behavior, numeric mismatches, proper-noun hallucinations, and remaining substitutions.

F.2 Teacher-Forced Reference-Token Probability

To complement the WER-based catastrophic-regression analysis at $r = 0.4$, we further analyze the probability assigned to reference transcript tokens under teacher forcing. This auxiliary diagnostic is not used as the primary evaluation metric, but helps assess whether aggressive compression reduces the probability assigned to the reference tokens. For each utterance, we compute the average probability assigned to the reference tokens.

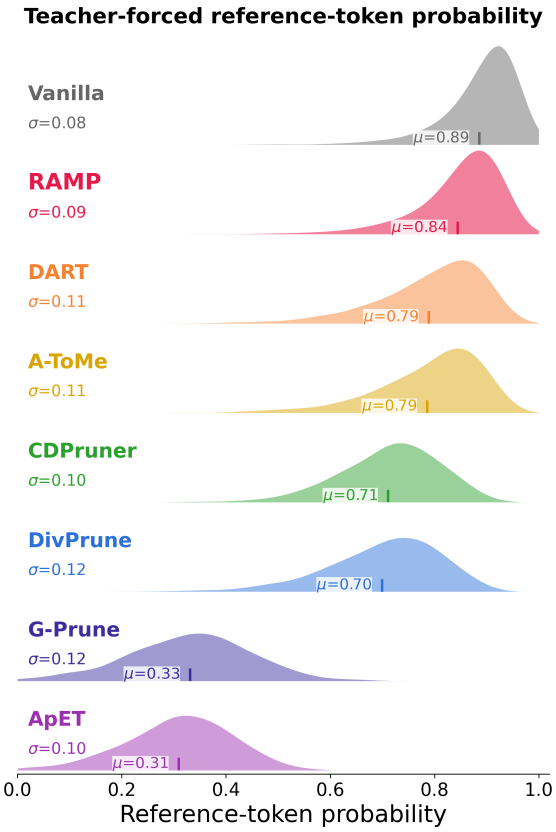


Figure 4: Distribution of teacher-forced reference-token probability for Audio-Flamingo-3 at $r = 0.4$ on the full LibriSpeech *test-clean* set.

Figure 4 shows the resulting distribution on Audio-Flamingo-3 at $r = 0.4$. Among the compression methods, RAMP achieves the highest mean reference-token probability and remains closest to Vanilla. In contrast, several baselines shift the distribution toward lower probabilities, suggesting that they assign lower probability to the reference tokens. This provides additional diagnostic evidence that RAMP better preserves reference-token probability under aggressive compression, consistent with the lower catastrophic-regression counts in Table 16.

F.3 Representative Regressions

Table 17 shows representative catastrophic regressions, including baseline-only regressions where RAMP matches the reference and shared failures where RAMP also regresses. Part A illustrates that several baselines collapse into label-like predictions, output repetitions, short-reference substitutions, or other unstable outputs, whereas RAMP avoids these regressions on the same audio. Part B

shows that RAMP can still regress through repeated phrases, short-reference errors, or difficult residual transcription failures, although such cases are rare compared with the baselines.

In addition to the case-level examples, we observe a cross-sample recurring-output pattern: the same verbatim predictions recur across many distinct audio inputs under several baselines. Examples include prompt echoes such as "transcribe the audio", refusal-like outputs such as "i am sorry", and single-token collapses such as "and". This pattern suggests that severe compression can weaken acoustic grounding and increase reliance on generic, prompt-like, or degenerate outputs.

F.4 Caveats

This analysis uses conservative thresholds and therefore focuses on catastrophic regressions rather than moderate WER degradation. Some failure patterns, such as label-like or refusal-like outputs, also appear occasionally in Vanilla AF3 outputs; this suggests that compression can amplify pre-existing model weaknesses rather than introducing all such behaviors from scratch. Finally, `short_ref` does not mean that short utterances generally fail under compression. Rather, it indicates that one or two word errors can already yield sample-level WER at or above 100% for very short references.

Reference	Prediction
Part A. Baseline-only regressions *(RAMP matches reference)*	
`refusal`	
"so it is like back to school it is christmas"	**DART:** "neutral"
"what is your favorite beverage"	**ApET:** "happy"
"lush ferns grow on the lofty rocks"	**G-Prune:** *empty output*
`repetition`	
"and love be false"	**G-Prune:** "and `e` `e` `e` `e` `e` `e` …"
"other engineers joined later"	**ApET:** "others are `the` `the` `the` `the` `the` …"
"where are the keys"	**ApET:** "we are `the ones who are going to be` `the ones who are going to be` …"
`short_ref`	
"children play badminton on the lawn"	**DivPrune:** "she did not play minton"
"did he trip him"	**A-ToMe:** "triple"
"chutes and ladders"	**DART:** "shoes louder"
`other`	
"all were portraits"	**ApET:** "the song was released on the album the new yorker"
"general purpose tape library series"	**CDPruner:** "in the parker is tape release series"
"it has to be"	**DART:** "it has a very strong sense of place"
Part B. Shared failures *(RAMP also regresses)*	
`repetition`	
"yes that is how close that dave chappelle was chris dopey ah said because jesse told me to have fun papa g be best"	**RAMP:** "so `that is how close` `that is how close` …" **DivPrune:** "yes `how close that is` `how close that is` …" **A-ToMe:** "yes `that is how` `that is how` `that is how` …"
`short_ref`	
"grumpy yoda"	**RAMP:** "grampyoda" **A-ToMe:** "grampy" **DART:** "grand prix"
`other`	
"the slogan lower rates better service was coined"	**RAMP:** "fezlogan lowers barrel service what is kind" **DivPrune:** "fezloghlan lowers his rifle skimming" **CDPruner:** "feast of all saints what is kind"
"asked zarathustra vehemently what doest thou here"	**RAMP:** "as t wast thou vehement what saint thou" **G-Prune:** "as their meaty word"

Table 17: Representative compression-induced catastrophic regressions for Audio-Flamingo-3 at $r = 0.4$. Part A shows baseline-only regressions where RAMP matches the reference; Part B shows shared failures where RAMP also regresses. Predictions are shown after text normalization; boxed spans highlight repeated tokens or phrases, and … indicates truncated text.