FASTAV: EFFICIENT TOKEN PRUNING FOR AUDIO-VISUAL LARGE LANGUAGE MODEL INFERENCE

Chaeyoung Jung, Youngjoon Jang, Seungwoo Lee, Joon Son Chung

Korea Advanced Institute of Science and Technology, South Korea

ABSTRACT

In this work, we present FastAV, the first token pruning framework tailored for audio-visual large language models (AV-LLMs). While token pruning has been actively explored in standard large language models (LLMs) and vision-language models (LVLMs), its application to AV-LLMs has received little attention, even though multimodal integration substantially increases their token demands. To address this gap, we introduce a pruning strategy that utilizes attention weights to identify tokens emphasized at different stages and estimates their importance. Building on this analysis, FastAV applies a two-stage pruning strategy: (1) global pruning in intermediate layers to remove broadly less influential tokens, and (2) fine pruning in later layers considering the impact on next token generation. Notably, our method does not rely on full attention maps, which makes it fully compatible with efficient attention mechanisms such as FlashAttention. Extensive experiments demonstrate that FastAV reduces FLOPs by more than 40% on two representative AV-LLMs, while preserving or even improving model performance.

***Index Terms*—** Audio-visual LLM, LLM token pruning, Multimodal LLM

1. INTRODUCTION

Recent advances in large language models (LLMs) [1–4] have driven growing interest in extending them to multimodal settings by incorporating vision, audio, and other modalities to tackle more complex tasks. However, as model sizes and the number of processed tokens grow substantially, both memory consumption and computational cost increase sharply, leading to degraded inference efficiency. To address these challenges, recent studies have explored various strategies to improve model efficiency, ranging from structure-level pruning [5–9] to token-level pruning [10–14], all aiming to reduce overhead while preserving model performance.

Several pruning strategies for LLMs have been proposed and can be categorized as post-training, training-free, and inference-time methods. Post-training methods compress pretrained models via structured pruning, including gradient-guided pruning with LoRA recovery [5] and refined importance metrics [6, 15]. Training-free methods enhance model efficiency without fine-tuning, such as Hessian-based reconstruction [7], activation-preserving criteria [8], and dynamic sparse masks [9]. For inference, LazyLLM [10] dynamically prunes tokens during generation by selectively computing key–value pairs for tokens deemed important, even if they are discarded in earlier layers. Extending inference-time token pruning to large vision-language models (LVLMs), recent studies focus on efficient visual token selection. FastV [11] prunes visual tokens in later layers based on adaptive attention patterns learned in early layers. VTW [12] removes mid-to-late visual tokens using attention-sink and information-migration signals. TopV [13] prunes less important visual tokens using a visual-aware cost function.

Unlike the extensive study of token pruning in LLMs and LVLMs, its application to audio-visual LLMs (AV-LLMs) remains underexplored. AV-LLMs [16–30] handle rich multimodal streams of audio, video, and text, leading to an inevitable increase in token counts. However, it is still unclear how many of these tokens are essential for downstream reasoning. To bridge this gap, we present FastAV, a framework that analyzes AV-LLMs to identify key audio-visual tokens and prune less informative ones at inference, enabling efficient computation without compromising performance.

To gain deeper insights into the behavior of AV-LLMs, we analyze attention rollout [31], which accumulates attention across layers to trace token influence. Higher rollout values indicate stronger influence, highlighting critical tokens. We apply this to two representative AV-LLMs, VideoLLaMA2 [19] and video-SALMONN2 [18]. As shown in Fig.1, our rollout analysis reveals that after tokens pass through the middle layer (layer 14), attention increasingly concentrates on earlier tokens, particularly on the left side of the red line, forming anchor-like patterns [32] in both models. In the first row of Fig. 2, we observe that rollout in early layers such as layer 4 remains uniform, whereas by layer 14 it clearly shifts toward earlier tokens. This rollout pattern then persists through deeper layers, including layer 24.

Motivated by this observation, we apply global pruning at the middle layer to roughly remove less influential tokens. Since this stage already eliminates more than half of the tokens, the remaining ones must be pruned with caution to avoid harming model performance. To further increase inference efficiency while preserving important information, we intro-

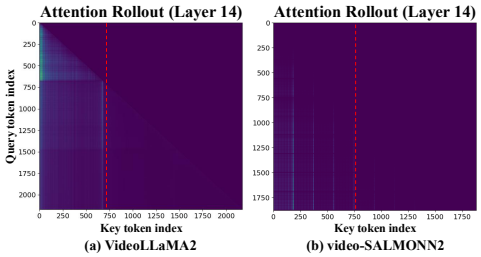

Fig. 1. **Attention rollout at the 14th layer in VideoLLaMA2 [19] and video-SALMONN2 [18].** Accumulated attention concentrates on earlier tokens, highlighting their pivotal role in carrying the most essential information.

duce a fine pruning stage. Following prior studies [33, 34], we estimate token importance using the attention weights of the last query token, and progressively remove the least important tokens at each subsequent layer based on their contribution to the final answer, further reducing computational costs.

Experimental results show that FastAV reduces FLOPs by over 40% in two representative AV-LLMs, VideoLLaMA2 and video-SALMONN2, while maintaining or even improving performances across three datasets. Furthermore, applying FastAV to both AV-LLMs drastically reduces audio tokens (e.g., from 1,496 to 10 in VideoLLaMA2) without degrading performance, highlighting their essential yet compact role and underscoring the need for efficient strategies to leverage multimodal information, particularly audio.

Our contributions are threefold. First, we present FastAV, the first systematic analysis of the role and impact of audio and video tokens in AV-LLMs. Second, we introduce a two-stage pruning strategy that combines global pruning guided by attention rollout with fine grained pruning based on last query token importance. Third, we show that FastAV reduces FLOPs by more than 40% while maintaining or even improving model performance, enabling efficient inference.

2. METHOD

2.1. AV-LLM Inference

For video data, a visual encoder extracts frame-level features into a sequence of M visual tokens $\mathbf{X}^{vis} = [\mathbf{x}_1^{vis}, ..., \mathbf{x}_M^{vis}]$. Similarly, an audio encoder transforms audio signals into U audio tokens $\mathbf{X}^{aud} = [\mathbf{x}_1^{aud}, ..., \mathbf{x}_U^{aud}]$. Text inputs, including a question, are tokenized into E textual tokens $\mathbf{X}^{lang} = [\mathbf{x}_1^{lang}, ..., \mathbf{x}_E^{lang}]$. These modality-specific tokens are concatenated to form a unified sequence $\mathbf{X} = [\mathbf{x}_i]_{i=1}^{K}$, where $K = M + U + E$, which serves as the input to the LLM decoder. During inference, the model generates outputs autoregressively, predicting the next token conditioned on all input modalities and previously generated outputs:

$$\mathbf{y}_t \sim p(\mathbf{y}_t|\mathbf{X}^{vis}, \mathbf{X}^{aud}, \mathbf{X}^{lang}, \mathbf{y}_{<t}), \tag{1}$$

where $\mathbf{y}_t$ is the token at timestep t and $\mathbf{y}_{<t}$ are past tokens.

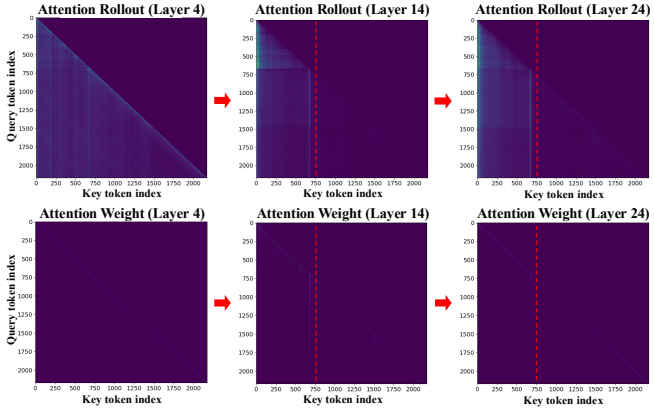

Fig. 2. **Attention rollout and weights across layers in VideoLLaMA2 [19].** Attention rollout reveals a progressive focus on earlier tokens, stabilizing around the middle layers and persisting in deeper layers, whereas raw attention weights alone do not exhibit such a clear pattern.

2.2. FastAV

Analysis of audio-visual token importance via attention rollout. To explore the contribution of multimodal tokens across transformer layers, we use the attention rollout method [31], which tracks how information propagates through the network as depth of the layer increases. Attention rollout aggregates attention weights from the initial to the current layer. Since each transformer layer has multiple attention heads, we first aggregate them by averaging across heads, resulting in a single attention matrix $\mathbf{A}^l \in \mathbb{R}^{n \times n}$ for layer l, where n is the number of tokens. Each row of $\mathbf{A}^l$ represents a token's attention distribution over all tokens. To preserve each token's original representation and prevent the attention rollout from being dominated solely by raw attention weights, we incorporate residual connections by forming a convex combination of $\mathbf{A}^l$ and the identity matrix $\mathbf{I}$:

$$\tilde{\mathbf{A}}^{\,l} = \alpha\, \mathbf{A}^{\,l} + (1 - \alpha)\, \mathbf{I}. \tag{2}$$

Here, α plays the role of balancing residual connections and aggregated attention. Larger values highlight inter-token dependencies, while smaller values reinforce each token's own representation, stabilizing the rollout. The cumulative attention rollout up to layer l is computed as the sequential matrix multiplication of these modified attention matrices:

$$\mathbf{R}^{\,l} = \tilde{\mathbf{A}}^{\,l}\, \tilde{\mathbf{A}}^{\,l-1} \cdots \tilde{\mathbf{A}}^{\,1}. \tag{3}$$

This matrix represents how much each token from the input layer influences every other token after propagating through all attention layers up to l. We use attention rollout to gain deeper insight into the model's behavior, rather than relying solely on raw attention weights. To illustrate its effectiveness, Fig. 2 compares attention rollout with raw attention weights at early (layer 4), middle (layer 14), and late (layer 24) stages of the 28-layer VideoLLaMA2 [19] network. In the first row, showing the attention rollout, we observe that attention gradu-

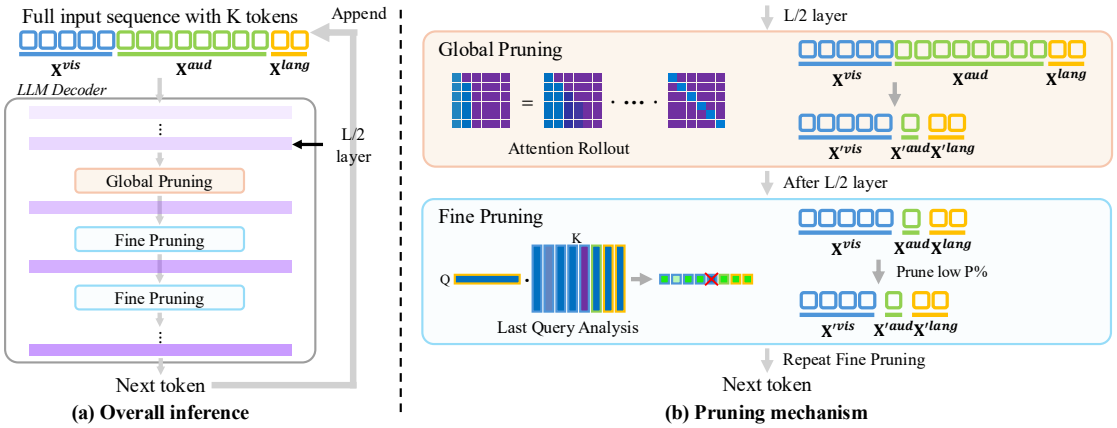

Fig. 3. **Overview of the FastAV framework.** FastAV starts with the full input context and reduces computation through two-stage pruning. In the middle layer, global pruning removes later tokens guided by attention rollout. In subsequent layers, fine-grained pruning discards the least important P% of remaining tokens based on last-query token analysis.

ally concentrates on the earliest tokens starting from the middle layer, as highlighted by the red line. By the late layer, this rollout pattern has stabilized. In contrast, the second row, showing raw attention weights, reveals no clear pattern. This highlights the effectiveness of attention rollout in capturing the model's behavior and guiding global pruning decisions.

Global pruning. Guided by attention rollout, we first perform global pruning to remove tokens in later positions, as attention increasingly concentrates on earlier tokens after the middle layers (L/2), as shown in Fig. 3 (a). For both models, we analyze 100 non-test samples and apply an attention rollout threshold at the middle layer to remove less influential video and audio tokens, typically those occurring beyond position 750 in the sequence. As a result, approximately two-thirds of the later tokens are removed for VideoLLaMA2, while more than half are removed for video-SALMONN2.

Fine pruning. After global pruning, most tokens have already been discarded, so subsequent pruning must be applied carefully to further improve inference efficiency without harming performance. We perform fine pruning in the layers following the middle layer where global pruning is applied, using the last query token in the attention weight, as illustrated in Fig. 3 (b). As in prior works [33, 34], the last query token analysis efficiently identifies important tokens, as it directly influences the next token's prediction without computing the full attention matrix. Let $\mathbf{Q}^l_{\text{last}} \in \mathbb{R}^{h\times 1\times d}$ denote the last query feature at layer l, and $\mathbf{K}^l \in \mathbb{R}^{h\times n\times d}$ represents the key features of the n tokens remaining after global pruning, where h is the number of attention heads and d is the hidden dimension. The importance scores are computed as:

$$\mathbf{s}^l = \text{mean}_h\Big(\text{softmax}(\mathbf{Q}^l_{\text{last}}(\mathbf{K}^l)^\top)\Big), \tag{4}$$

where the mean is taken over attention heads. At each layer following global pruning, tokens with the lowest P% scores are removed, iteratively refining the token set.

3. EXPERIMENTS

3.1. Experimental setup

Baselines. We evaluate our approach using two representative AV-LLMs: VideoLLaMA2 [19] and video-SALMONN2 [18].

Datasets. AVQA [35] dataset contains 57,000 YouTube videos for real-world audio-visual understanding. MUSIC-AVQA [36] includes 45,867 question-answer pairs from 9,288 music videos, focusing on audio-visual reasoning. AVHBench [37] evaluates audio-visual hallucinations and is divided into three subtasks: audio- or video-driven hallucination (AV hallucination), audio-visual matching (AV matching), and audio-visual captioning (AV captioning).

Evaluation protocol. For AVHBench, we report accuracy excluding AV captioning. For AVQA, MUSIC-AVQA, and AV captioning, which have open-ended responses, we follow the GPT-assisted evaluation protocol from VideoLLaMA2[1] and report only the average score, as standard deviations are negligible. FLOPs are measured relative to the original theoretical one (set to 100) as in [11], and latency indicates the time in seconds to generate a single token during a forward pass.

Implementation details. For global pruning in VideoLLaMA2, all video tokens precede the audio tokens, so we keep only the first 10 audio tokens and prune the rest. In video-SALMONN2, video and audio tokens are interleaved at the frame level, so we prune the later frames while retaining the first 4. For fine pruning, we set the pruning ratio P to 20% of the tokens at each layer after the middle layer.

3.2. Experimental Results

Main results. To demonstrate the effectiveness of FastAV, we conduct experiments on VideoLLaMA2 and video-SALMONN2 across three datasets: AVQA, MUSIC-AVQA, and AVHBench. As shown in Table 1, FastAV consistently reduces theoretical FLOPs by over 40%, accelerates inference

[1] https://github.com/DAMO-NLP-SG/VideoLLaMA2/tree/audio_visual

Table 1. Theoretical FLOPs and accuracy on VideoLLaMA2 [19] and video-SALMONN2 [18] across AVQA [35], MUSIC-AVQA [36], and AVHBench [37]. FastAV significantly reduces FLOPs while maintaining comparable accuracy without additional training. NA indicates not applicable, as MUSIC-AVQA contains long videos unsuitable for video-SALMONN2.

Method	FLOPs↓	Latency↓	Memory↓	MUSIC-AVQA↑	AVQA↑	AVHBench		
						AV hallucination↑	AV matching↑	AV captioning↑
VideoLLaMA2 [19]	100	0.43	22G	**81.3**	61.4	77.9	57.8	2.8
w/ FastAV	**56**	**0.32**	**19G**	81.2	**62.3**	**78.2**	**69.0**	**2.9**
video-SALMONN2 [18]	100	0.44	28G	NA	57.6	64.5	**50.8**	**3.2**
w/ FastAV	**58**	**0.29**	**21G**	NA	**58.4**	**64.8**	50.7	3.1

Table 2. Comparison of global pruning strategies with VideoLLaMA2 on AVHBench. Low informative pruning performs best, highlighting the value of attention rollout.

Method	FLOPs	AVHBench		
		AV hallucination	AV matching	Avg
Vanilla	100	77.9	57.8	70.7
Random	65	77.2	54.2	69.0
Top attentive		76.1	51.7	67.4
Low attentive		77.5	57.8	70.5
Top informative		72.3	50.9	64.7
Low informative (Ours)		**78.7**	**67.7**	**74.5**

Table 3. Comparison of fine pruning strategies with VideoLLaMA2 on AVHBench. Low attentive pruning achieves the best performance, demonstrating its effectiveness.

Method	FLOPs	AVHBench		
		AV hallucination	AV matching	Avg
Vanilla	100	77.9	57.8	70.7
Random	56	76.1	54.9	68.5
Top attentive		74.5	52.8	66.8
Low attentive (Ours)		**78.2**	**69.0**	**74.9**

by approximately 30%, and lowers memory consumption, all while maintaining strong performance across tasks. Notably, on VideoLLaMA2, accuracy on the AV matching task improves by over 10%, suggesting that removing more than 99% of audio tokens may enhance multimodal understanding. FastAV significantly enhances computational efficiency, maintaining model performance without additional training.

Global pruning strategy. To assess the effectiveness of our global pruning method, we compare several strategies on VideoLLaMA2 using AVHBench in Table 2 without applying fine pruning. Vanilla represents the original model inference without pruning, and random pruning removes tokens arbitrarily while keeping FLOPs constant. In contrast, pruning top attentive tokens based on last-query attention weights significantly degrades performance, since high-attention tokens often carry critical information [33, 34]. Pruning low attentive tokens preserves vanilla performance, but our rollout-based low-informative strategy outperforms it, demonstrating that attention rollout captures token importance more effectively than raw attention for global pruning. Additionally, removing highly informative tokens leads to the worst performance, highlighting their essential role in our approach.

Fine pruning strategy. Table 3 compares fine pruning strategies on VideoLLaMA2 using AVHBench. The low-attentive approach consistently outperforms random pruning, demonstrating its effectiveness as the second stage of FastAV.

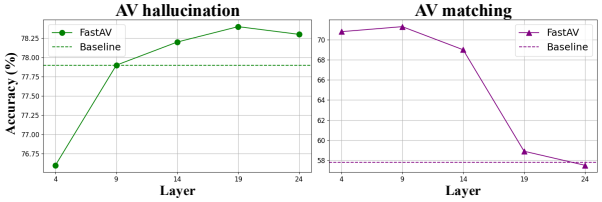

Fig. 4. Layerwise accuracy of VideoLLaMA2 on AVHBench subtasks. The middle layer are chosen to balance performance between AV hallucination and AV matching tasks.

Table 4. Theoretical FLOPs and accuracy of VideoLLaMA2 under different pruning ratios P. FastAV significantly reduces FLOPs as pruning increases, with a 20% pruning ratio achieving the best accuracy at low FLOPs.

P (%)	FLOPs↓	AVHBench		
		AV hallucination	AV Matching	Avg
0	65	**78.7**	67.7	74.5
10	59	78.3	68.3	74.7
20 (Ours)	56	78.2	**69.0**	**74.9**
30	**54**	78.3	68.5	74.8

Pruning layer selection. To identify the optimal starting layer for two-stage pruning, we conduct layer-wise experiments on VideoLLaMA2 using AVHBench. Guided by attention rollout analysis, we select a middle layer (layer 14) as the pruning starting point. As shown in Fig. 4, pruning in the early layers degrades performance on the AV hallucination task, whereas pruning from the middle layer preserves performance and can even improve results across all tasks.

Pruning percentage P. To select an appropriate pruning ratio for fine pruning, we conduct experiments on VideoLLaMA2 using AVHBench. Table 4 reports results with pruning ratios increased in 10% increments, showing that higher pruning ratios further reduce FLOPs. A 0% pruning ratio corresponds to applying only global pruning, while a 20% pruning ratio reduces FLOPs by approximately 44% compared to the original inference and achieves the highest average performance.

4. CONCLUSION

We introduce FastAV, the first token pruning framework specifically designed for AV-LLMs, combining global pruning via attention rollout with fine pruning using last-query analysis to remove less informative tokens while preserving crucial information. Experiments show it reduces FLOPs by over 40%, enabling efficient processing of long, complex multimodal inputs without degrading performance.

5. ACKNOWLEDGEMENTS

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2023-00212845, Multimodal Speech Processing for Human-Computer Interaction).

References

[1] Josh Achiam, Steven Adler, et al., "Gpt-4 technical report," *arXiv:2303.08774*, 2023.

[2] Tom Brown et al., "Language models are few-shot learners," in *Proc. NeurIPS*, 2020.

[3] Yiheng Liu et al., "Summary of chatgpt-related research and perspective towards the future of large language models," *Meta-Radiology*, p. 100017, 2023.

[4] Romal Thoppilan et al., "Lamda: Language models for dialog applications," *arXiv:2201.08239*, 2022.

[5] Xinyin Ma et al., "Llm-pruner: On the structural pruning of large language models," in *Proc. NeurIPS*, 2023.

[6] Jialong Guo et al., "Slimllm: Accurate structured pruning for large language models," in *Proc. ICML*, 2023.

[7] Elias Frantar and Dan Alistarh, "Sparsegpt: Massive language models can be accurately pruned in one-shot," in *Proc. ICML*, 2023.

[8] Mingjie Sun et al., "A simple and effective pruning approach for large language models," in *Proc. ICLR*, 2024.

[9] Yuxin Zhang et al., "Dynamic sparse no training: Training-free fine-tuning for sparse llms," in *Proc. ICLR*, 2024.

[10] Qichen Fu et al., "Lazyllm: Dynamic token pruning for efficient long context llm inference," *arXiv:2407.14057*, 2024.

[11] Liang Chen et al., "An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models," in *Proc. ECCV*, 2024.

[12] Zhihang Lin et al., "Boosting multimodal large language models with visual tokens withdrawal for rapid inference," in *Proc. AAAI*, 2025.

[13] Cheng Yang et al., "Topv: Compatible token pruning with inference time optimization for fast and low-memory multimodal vision language model," in *Proc. CVPR*, 2025.

[14] Taehan Lee and Hyukjun Lee, "Token pruning in audio transformers: Optimizing performance and decoding patch importance," *arXiv:2504.01690*, 2025.

[15] Yingtao Zhang et al., "Plug-and-play: An efficient post-training pruning method for large language models," in *Proc. ICLR*, 2024.

[16] Sanjoy Chowdhury et al., "Meerkat: Audio-visual large language model for grounding in space and time," in *Proc. ECCV*, 2024.

[17] Guangzhi Sun et al., "video-salmonn: Speech-enhanced audio-visual large language models," in *Proc. ICML*, 2024.

[18] Changli Tang, Yixuan Li, et al., "video-salmonn 2: Captioning-enhanced audio-visual large language models," *arXiv:2506.15220*, 2025.

[19] Zesen Cheng, Sicong Leng, et al., "Videollama 2: Advancing spatial-temporal modeling and audio understanding in video-llms," *arXiv*, 2024.

[20] Qilang Ye, Zitong Yu, et al., "Cat: Enhancing multimodal large language model to answer questions in dynamic audio-visual scenarios," in *Proc. ECCV*, 2024.

[21] Jun Zhan, Junqi Dai, et al., "Anygpt: Unified multimodal llm with discrete sequence modeling," *arXiv:2402.12226*, 2024.

[22] Jiasen Lu, Christopher Clark, et al., "Unified-io 2: Scaling autoregressive multimodal models with vision language audio and action," in *Proc. CVPR*, 2024.

[23] Jiaming Han et al., "Imagebind-llm: Multi-modality instruction tuning," *arXiv:2309.03905*, 2023.

[24] Chenyang Lyu, Minghao Wu, et al., "Macaw-llm: Multi-modal language modeling with image, audio, video, and text integration," *arXiv:2306.09093*, 2023.

[25] Hang Zhang et al., "Video-llama: An instruction-tuned audio-visual language model for video understanding," in *Proc. EMNLP*, 2023.

[26] Sihan Chen et al., "Vast: A vision-audio-subtitle-text omni-modality foundation model and dataset," in *Proc. NeurIPS*, 2023.

[27] Yixuan Su, Tian Lan, Huayang Li, Jialu Xu, Yan Wang, and Deng Cai, "Pandagpt: One model to instruction-follow them all," *arXiv:2305.16355*, 2023.

[28] Zijia Zhao et al., "Chatbridge: Bridging modalities with large language model as a language catalyst," *arXiv:2305.16103*, 2023.

[29] Jiaming Han et al., "Onellm: One framework to align all modalities with language," in *Proc. CVPR*, 2024.

[30] Chaeyoung Jung et al., "Fork-merge decoding: Enhancing multimodal understanding in audio-visual large language models," *arXiv:2505.20873*, 2025.

[31] Samira Abnar and Willem Zuidema, "Quantifying attention flow in transformers," in *Proc. ACL*, 2020.

[32] Qidong Huang, Xiaoyi Dong, Pan Zhang, et al., "Opera: Alleviating hallucination in multi-modal large language models via over-trust penalty and retrospection-allocation," in *Proc. CVPR*, 2024.

[33] Woomin Song, Seunghyuk Oh, et al., "Hierarchical context merging: Better long context understanding for pre-trained llms," in *Proc. ICLR*, 2024.

[34] Chaeyoung Jung et al., "Avcd: Mitigating hallucinations in audio-visual large language models through contrastive decoding," *arXiv:2505.20862*, 2025.

[35] Pinci Yang et al., "Avqa: A dataset for audio-visual question answering on videos," in *Proc. ACM MM*, 2022.

[36] Guangyao Li et al., "Learning to answer questions in dynamic audio-visual scenarios," in *Proc. CVPR*, 2022.

[37] Kim Sung-Bin, Oh Hyun-Bin, et al., "Avhbench: A cross-modal hallucination benchmark for audio-visual large language models," in *Proc. ICLR*, 2025.