Fork-Merge Decoding: Enhancing Multimodal Understanding in Audio-Visual Large Language Models

Anonymous ACL submission

Abstract

The goal of this work is to enhance balanced multimodal understanding in audio-visual large language models (AV-LLMs) by addressing modality bias without additional training. In current AV-LLMs, audio and video features are typically processed jointly in the decoder. While this strategy facilitates unified multimodal understanding, it may introduce modality bias, where the model disproportionately relies on a single dominant modality. In this work, we provide an in-depth analysis of this phenomenon and propose Fork-Merge Decoding (FMD) as a targeted solution. FMD is a simple, training-free inference strategy that rectifies this imbalance without architectural modifications. Specifically, FMD isolates audio and visual inputs in the early decoding layers (*fork*) to enforce modality-specific reasoning, before integrating them for joint interpretation (*merge*). This two-stage mechanism effectively mitigates the dominance by ensuring balanced contributions from both modalities. Experiments on VideoLLaMA2, video-SALMONN, and Qwen2.5-Omni demonstrate that FMD consistently enhances multimodal understanding across diverse benchmarks.

1 Introduction

Recent advancements in large language models (LLMs) have demonstrated superior performance in various text-centric tasks such as problem solving, translation, and summarization (Achiam et al., 2023; Brown et al., 2020; Liu et al., 2023b; Thoppilan et al., 2022; Wei et al., 2022a,b; Zhao et al., 2023a). Building on these successes in text processing, LLMs have evolved to handle additional modalities, including images (Alayrac et al., 2022; Chen et al., 2023a; Dai et al., 2023; Huang et al., 2023; Li et al., 2023a; Liu et al., 2023a; Yu et al., 2024; Zhang et al., 2024; Zhu et al., 2023b), videos (Lin et al., 2024; Maaz et al., 2024), and audio (Huang et al., 2024; Rubenstein et al., 2023; Tang et al., 2024), giving rise to multimodal LLMs (MLLMs). These models process diverse modalities through separate encoders and integrate their outputs within a decoder language model, achieving remarkable performance across a wide range of tasks. Among these, audio-visual LLMs (AV-LLMs) are particularly notable for their ability to jointly integrate visual and auditory information, supporting more sophisticated reasoning and achieving closer alignment with human multimodal perception (Cheng et al., 2024; Sun et al., 2024; Zhang et al., 2023; Xu et al., 2025).

To effectively leverage pretrained LLM decoders in AV-LLMs, various fusion strategies have been proposed to integrate audio and visual information. One common approach (Cheng et al., 2024; Chowdhury et al., 2024; Lyu et al., 2023; Panagopoulou et al., 2023; Ye et al., 2024; Zhang et al., 2023; Zhao et al., 2023b; Xu et al., 2025) is token-wise fusion, where audio and visual features are extracted by separate encoders and then concatenated along the sequence dimension before being fed into the decoder as a continuous input sequence. Several studies (Chowdhury et al., 2024; Ye et al., 2024) additionally introduce adapter modules that facilitate interaction between audio and visual features before they are passed into the LLM. Another approach (Han et al., 2023; Su et al., 2023; Sun et al., 2024) is channel-wise fusion, in which modality-specific features are concatenated along the channel dimension to form a unified representation. In most current AV-LLM architectures, the decoder receives both audio-visual inputs simultaneously, which raises a potential concern: if the model finds one modality easier to interpret—perhaps due to better alignment with its pretraining objectives—it may over-rely on that modality, sometimes leading to hallucinations (Leng et al., 2024a).

To investigate this possibility, we begin by analyzing the attention weight distributions over audio-visual inputs using 100 samples from the AVH-

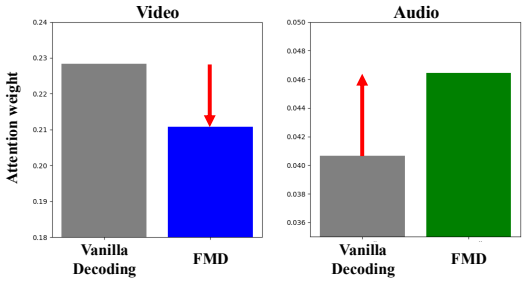


Figure 1: **Attention weight analysis in VideoLLaMA2 on the AVHBench dataset.** We analyze 100 samples and examine the attention weights from the last decoder layer, focusing on the final token of the question. Attention is disproportionately allocated to video inputs over audio, revealing a modality bias. Our proposed FMD method reduces this gap by encouraging more balanced contributions from both modalities.

Bench dataset (Sung-Bin et al., 2025). Our analysis examines the final decoder layer of VideoLLaMA2 (Cheng et al., 2024), focusing on the attention weights of the last token. Since the token is critical for predicting the next token, we use it to quantify the relative attention allocated to each modality. As shown in Figure 1, the vanilla decoding setup, which reflects the default inference behavior of the model, exhibits a clear bias toward video inputs, with attention concentrated on visual features over audio. This aligns with findings from recent studies (Guan et al., 2024; Leng et al., 2024a; Nishimura et al., 2024; Wang et al., 2024), which report that MLLM decoders often exhibit modality bias. These studies highlight that such imbalances can lead to hallucinations or flawed reasoning, emphasizing the need to mitigate modality bias for more balanced multimodal understanding.

To address this issue, we propose Fork-Merge Decoding (FMD), which is a simple yet effective strategy that enhances multimodal understanding without altering the AV-LLM architecture or requiring additional training. The core idea of FMD is to divide the decoding phase into two stages: a *fork* phase and a *merge* phase, designed to improve both unimodal and multimodal understanding. In the *fork* phase, the original multimodal input is split into two unimodal branches by zeroing out either the visual or auditory modality while retaining the text question. Each branch is processed independently through the initial layers of the pretrained AV-LLM, producing modality-specific hidden representations without requiring additional full forward passes. In the *merge* phase, these representations are combined and passed through the remaining decoder layers. This separation enables the model to first attend to unimodal cues in isolation before integrating them for complementary multimodal understanding. As shown in Figure 1, FMD reduces the attention weight on video inputs by 14% and increase the weight on audio inputs by 7%. This adjustment balances the modality bias while still preserving the attributes of the pretrained model. Building on this, since recent AV-LLMs commonly adopt either token-wise or channel-wise concatenation for audio-visual fusion, we propose a generalized decoding strategy that is compatible with both fusion methods. This unified approach improves performance across a variety of models, regardless of their fusion mechanism.

Furthermore, we evaluate the effectiveness of FMD by applying it to three recent AV-LLMs: VideoLLaMA2 (Cheng et al., 2024) and Qwen2.5-Omni (Xu et al., 2025) (token-wise fusion) and video-SALMONN (Sun et al., 2024) (channel-wise fusion). Applying FMD leads to consistent performance improvements in all baselines across five widely used audio-visual benchmarks: AVHBench (Sung-Bin et al., 2025), AVQA (Yang et al., 2022), MUSIC-AVQA (Li et al., 2022), AVUT (Yang et al., 2025), and WorldSense (Hong et al., 2025). Notably, FMD enhances performance not only in tasks that emphasize a single modality but also in tasks that require balanced reasoning across both modalities. These results show that FMD enhances multimodal understanding by fully utilizing information from each modality.

2 Preliminaries

Input processing. Most existing AV-LLMs (Chen et al., 2023b; Cheng et al., 2024; Chowdhury et al., 2024; Lyu et al., 2023; Panagopoulou et al., 2023; Sun et al., 2024; Ye et al., 2024; Zhan et al., 2024; Zhang et al., 2023; Zhao et al., 2023b) process audio and video separately before feeding them into an LLM decoder. Visual inputs are encoded frame by frame into spatial embeddings, while audio signals are mapped to semantic representations capturing acoustic and prosodic cues. Textual instructions or questions are tokenized and embedded by a tokenizer of language model.

For token-wise fusion models, video frames are transformed into M embeddings $\boldsymbol{x}^v = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M\}$ and audio into N embeddings $\boldsymbol{x}^a = \{\boldsymbol{x}_{M+1}, \ldots, \boldsymbol{x}_{M+N}\}$. These are concatenated with L text embeddings $\boldsymbol{x}^l = \{\boldsymbol{x}_{M+N+1}, \ldots, \boldsymbol{x}_{M+N+L}\}$ to form the final sequence $\boldsymbol{x} = \boldsymbol{x}^v \oplus \boldsymbol{x}^a \oplus \boldsymbol{x}^l$ of length $M +$

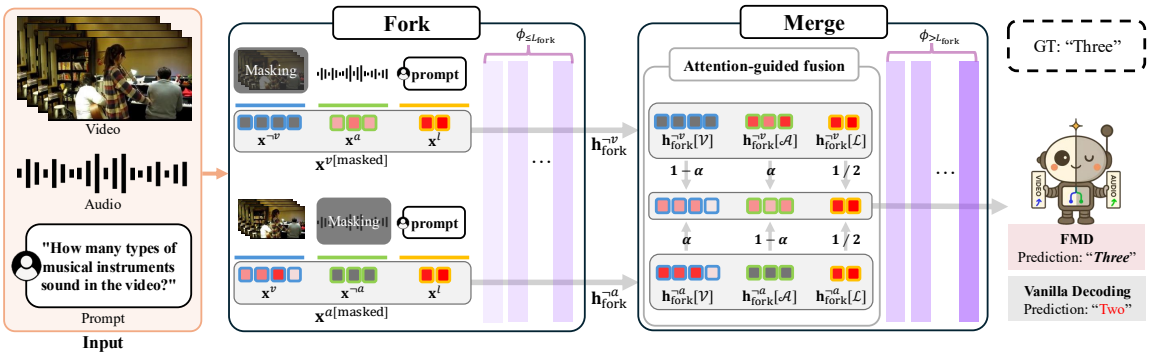


Figure 2: **Overview of the Fork-Merge Decoding pipeline.** The AV-LLM takes video frames, an audio waveform, and a question prompt as input. In the *fork* phase, FMD masks one modality while preserving the question, enabling independent reasoning. After L_{fork} decoder layers, the *merge* phase combines $\boldsymbol{h}_{\text{fork}}^{\neg v}$ and $\boldsymbol{h}_{\text{fork}}^{\neg a}$ with an attention-derived weight α, and the merged representation is processed by the remaining layers to generate answers with balanced multimodal understanding.

$N + L$, omitting instruction tokens for simplicity. For channel-wise fusion models, visual and audio features are projected to a fixed length U embeddings and then concatenated along the channel dimension to form joint audio-visual embeddings, $\boldsymbol{x}_i^{av} = [\boldsymbol{x}_i^v; \boldsymbol{x}_i^a]$. The resulting set $\boldsymbol{x}_i^{av} = \{\boldsymbol{x}_1^{av}, \ldots, \boldsymbol{x}_U^{av}\}$ is combined with L text embeddings via token-wise concatenation, $\boldsymbol{x} = \{\boldsymbol{x}_1^{av}, \ldots, \boldsymbol{x}_U^{av}, \boldsymbol{x}_{U+1}^l, \ldots, \boldsymbol{x}_{U+L}^l\}$, which is then fed into the LLM decoder.

Decoding. Both token-wise and channel-wise fusion-based AV-LLMs generate outputs from the input sequence $\boldsymbol{x}$ using an autoregressive decoding strategy, where each token is predicted by attending to previously generated tokens under a causal mask. At each decoding step t, the model generates the next token $\boldsymbol{y}_t$ conditioned on the input sequence $\boldsymbol{x}$, which includes video, audio, and text prompts, as well as the previously generated tokens $\boldsymbol{y}_{<t}$, with $\boldsymbol{y}_t \sim p(\boldsymbol{y}_t|\boldsymbol{x}, \boldsymbol{y}_{<t}) \propto \exp\left(\text{logit}(\boldsymbol{y}_t|\boldsymbol{x}, \boldsymbol{y}_{<t})\right)$.

3 Fork-Merge Decoding

This section introduces Fork-Merge Decoding (FMD) for *token-wise* (Section 3.1) and *channel-wise* (Section 3.2) fusion in AV-LLMs. An overview is illustrated in Figure 2.

3.1 Decoding with token-wise fusion in AV-LLMs

Input masking strategy. In token-wise fusion models like VideoLLaMA2 (Cheng et al., 2024), the input sequence is composed of $\boldsymbol{x} = \boldsymbol{x}^v \oplus \boldsymbol{x}^a \oplus \boldsymbol{x}^l$, where $\boldsymbol{x}^v$, $\boldsymbol{x}^a$, and $\boldsymbol{x}^l$ denote visual, audio, and language embeddings, respectively. To enable modality-specific processing for audio and video while preserving the textual question, we create two masked input variants as follows:

$$\begin{aligned} \boldsymbol{x}^{v[\text{masked}]} &= \boldsymbol{x}^{\neg v} \oplus \boldsymbol{x}^a \oplus \boldsymbol{x}^l, \\ \boldsymbol{x}^{a[\text{masked}]} &= \boldsymbol{x}^v \oplus \boldsymbol{x}^{\neg a} \oplus \boldsymbol{x}^l, \end{aligned} \tag{1}$$

where $\boldsymbol{x}^{\neg v}$ and $\boldsymbol{x}^{\neg a}$ denote the modality-masked embeddings for vision and audio, respectively. These are obtained by zeroing out the corresponding video frames or audio waveforms at the input. This preserves original embedding shapes and positions while removing content information.

Fork processing. To encourage independent understanding over each modality, each masked sequence is separately processed through the first L_{fork} transformer layers of the decoder ϕ:

$$\begin{aligned} \boldsymbol{h}_{\text{fork}}^{\neg v} &= \phi_{\leq L_{\text{fork}}}(\boldsymbol{x}^{v[\text{masked}]}), \\ \boldsymbol{h}_{\text{fork}}^{\neg a} &= \phi_{\leq L_{\text{fork}}}(\boldsymbol{x}^{a[\text{masked}]}), \end{aligned} \tag{2}$$

where $\boldsymbol{h}_{\text{fork}}^{\neg v}$ and $\boldsymbol{h}_{\text{fork}}^{\neg a}$ denote the intermediate hidden states obtained from the vision-masked and audio-masked inputs, respectively. This design ensures that the model does not observe both audio and visual inputs simultaneously in the early stages, allowing it to focus on how each modality individually relates to the textual prompt. By reasoning over each modality in isolation, the model is less likely to become biased toward the more dominant or easier-to-interpret modality.

Merge processing. After the L_{fork} layers, the hidden states $\boldsymbol{h}_{\text{fork}}^{\neg v}$ and $\boldsymbol{h}_{\text{fork}}^{\neg a}$ are fused and passed through the remaining transformer layers. Each hidden state has a sequence length of $M + N + L$, corresponding to the visual, audio, and text tokens. The fusion is performed by summing the corresponding embeddings at each modality position,

and is formally defined as follows:

$$\begin{aligned}\boldsymbol{h}_{\text{merge}}[\mathcal{V}] &= (1-\alpha)\cdot \boldsymbol{h}^{\neg v}_{\text{fork}}[\mathcal{V}] + \alpha \cdot \boldsymbol{h}^{\neg a}_{\text{fork}}[\mathcal{V}],\\ \boldsymbol{h}_{\text{merge}}[\mathcal{A}] &= \alpha\cdot \boldsymbol{h}^{\neg v}_{\text{fork}}[\mathcal{A}] + (1-\alpha)\cdot \boldsymbol{h}^{\neg a}_{\text{fork}}[\mathcal{A}],\\ \boldsymbol{h}_{\text{merge}}[\mathcal{L}] &= \tfrac{1}{2}\left(\boldsymbol{h}^{\neg v}_{\text{fork}}[\mathcal{L}] + \boldsymbol{h}^{\neg a}_{\text{fork}}[\mathcal{L}]\right),\end{aligned} \tag{3}$$

where $\mathcal{V} = [1{:}M]$, $\mathcal{A} = [M{+}1{:}M{+}N]$, and $\mathcal{L} = [M{+}N{+}1{:}M{+}N{+}L]$ denote the index ranges corresponding to the visual, audio, and language embeddings, respectively. The final merged representation $\boldsymbol{h}_{\text{merge}}$ is then constructed by concatenating the modality-specific segments along the token dimension as $\boldsymbol{h}_{\text{merge}} = \boldsymbol{h}_{\text{merge}}[\mathcal{V}] \oplus \boldsymbol{h}_{\text{merge}}[\mathcal{A}] \oplus \boldsymbol{h}_{\text{merge}}[\mathcal{L}]$. Here, α is a fusion weight for unmasked segments, reflecting their relative contribution when combining masked and unmasked segments. We refer to this approach as attention-guided fusion.

Attention-guided fusion. To determine the fusion weight α, we use the attention matrix $\mathbf{A}^{\text{final}} \in \mathbb{R}^{T\times T}$ from the final transformer layer. Specifically, we focus on the attention vector of the last token, $\mathbf{a}_{\text{last}} = \mathbf{A}^{\text{final}}_{T,:} \in \mathbb{R}^{T}$, which is critical in next-token prediction and follows the approach in prior studies (Huo et al., 2025; Song et al., 2024). By leveraging the two masked branches $\boldsymbol{h}^{\neg v}_{\text{fork}}$ and $\boldsymbol{h}^{\neg a}_{\text{fork}}$, we compute the attention-based contributions of unmasked segments by summing the attention mass over the corresponding regions relative to the total mass, which are then used as α:

$$\alpha = \frac{\sum_{i\in\mathcal{A}} \mathbf{a}^{v[\text{masked}]}_{\text{last}}[i] + \sum_{i\in\mathcal{V}} \mathbf{a}^{a[\text{masked}]}_{\text{last}}[i]}{\sum_{i\in\mathcal{V}\cup\mathcal{A}} (\mathbf{a}^{v[\text{masked}]}_{\text{last}}[i] + \mathbf{a}^{a[\text{masked}]}_{\text{last}}[i])}, \tag{4}$$

where $\mathbf{a}^{v[\text{masked}]}_{\text{last}}$ and $\mathbf{a}^{a[\text{masked}]}_{\text{last}}$ represent the attention weight distributions of the final token from $\boldsymbol{h}^{\neg v}_{\text{fork}}$ and $\boldsymbol{h}^{\neg a}_{\text{fork}}$, respectively. The final attention weight α is then obtained as the fraction of unmasked attention over the total attention in Eq. 4 and is used to interpolate between $\boldsymbol{h}^{\neg v}_{\text{fork}}$ and $\boldsymbol{h}^{\neg a}_{\text{fork}}$ in Eq. 3. This attention-guided fusion ensures that structurally aligned hidden states are preserved, while allowing the more informative modality to be emphasized. It enables a seamless merging scheme without disrupting the architectural integrity of pre-trained AV-LLMs. Instead of computing α for every test instance, we estimate it from a small random subset of N samples from AVHBench (Sung-Bin et al., 2025). This fixed value exhibits high stability across different sample sizes and domains, demonstrating that a single representative α suffices (see Section A.5 in our Supp).

Decoding. The merged hidden state is then forwarded through the remaining transformer layers: $\boldsymbol{h}_{\text{final}} = \phi_{>L_{\text{fork}}}(\boldsymbol{h}_{\text{merge}})$, producing the final prediction logits. By delaying modality fusion to deeper layers, where individual representations become semantically richer through the *fork* phase, our method enhances multimodal understanding while mitigating issues caused by modality imbalance.

3.2 Decoding with channel-wise fusion in AV-LLMs

Input masking strategy. In channel-wise fusion models such as video-SALMONN (Sun et al., 2024), the input sequence is structured as $\boldsymbol{x} = \{\boldsymbol{x}^{av}_1, \ldots, \boldsymbol{x}^{av}_U, \boldsymbol{x}^{l}_{U+1}, \ldots, \boldsymbol{x}^{l}_{U+L}\}$, where $\boldsymbol{x}^{av}$ denotes audio-visual embeddings and $\boldsymbol{x}^{l}$ corresponds to language (prompt) embeddings. Here, U and L indicate the number of audio-visual and language sequence elements, respectively. To allow for modality-specific processing, we construct two masked variants of the input:

$$\boldsymbol{x}^{v[\text{masked}]} = \boldsymbol{x}^{a\neg v} \oplus \boldsymbol{x}^{l}, \quad \boldsymbol{x}^{a[\text{masked}]} = \boldsymbol{x}^{\neg av} \oplus \boldsymbol{x}^{l}, \tag{5}$$

where $\boldsymbol{x}^{a\neg v}$ and $\boldsymbol{x}^{\neg av}$ denote the video-masked and audio-masked embeddings, respectively, obtained by zeroing out the corresponding inputs before they are passed into the decoder layers.

Fork-merge decoding. $\boldsymbol{h}^{\neg v}_{\text{fork}}$ and $\boldsymbol{h}^{\neg a}_{\text{fork}}$ are obtained by passing the masked inputs $\boldsymbol{x}^{v[\text{masked}]}$ and $\boldsymbol{x}^{a[\text{masked}]}$ through the L_{fork} layers (refer to Eq. 2). To compute the merged hidden state $\boldsymbol{h}_{\text{merge}}$, we perform element-wise addition over the audio-visual embedding from both branches, based on the assumption that they capture complementary information. Since each branch processes modality-masked inputs (i.e., audio-masked for visual features and vice versa), their combination is expected to yield a more complete representation. Additionally, mean pooling is applied over the question prompt embedding positions to maintain consistency:

$$\boldsymbol{h}_{\text{merge}}[i] = \begin{cases} \boldsymbol{h}^{\neg v}_{\text{fork}}[i] + \boldsymbol{h}^{\neg a}_{\text{fork}}[i], & \text{if } i \le U,\\ \tfrac{1}{2}\left(\boldsymbol{h}^{\neg v}_{\text{fork}}[i] + \boldsymbol{h}^{\neg a}_{\text{fork}}[i]\right), & \text{if } i > U.\end{cases} \tag{6}$$

We do not apply attention-guided fusion in the channel-wise setting, as the hidden states do not disentangle audio and visual embeddings along time axis. Subsequently, decoding is then continued by forwarding the merged hidden state $\boldsymbol{h}_{\text{merge}}$ through the remaining decoder layers to produce the final

Table 1: **Comparison of audio-visual understanding performance.** FMD improves VideoLLaMA2, video-SALMONN, and Qwen2.5-Omni over their vanilla decoding strategies across all benchmarks, particularly on AV matching (AV Mat.) and AV captioning (AV Cap.). Qwen2.5-Omni results on MUSIC-AVQA, AVUT, and WorldSense are marked N/A due to out-of-memory constraints.

Model	Decoding	AVHBench				AVQA	MUSIC -AVQA	AVUT	WorldSense
		A→V	V→A	AV Mat.	AV Cap.				
VideoLLaMA2	Vanilla	80.02	77.03	57.75	2.84	60.23	81.30	44.44	29.41
	FMD	**80.45**	**77.52**	**59.01**	**2.95**	**61.46**	**81.50**	**45.13**	**29.91**
video-SALMONN	Vanilla	68.69	62.39	49.46	1.83	28.20	44.48	32.36	56.20
	FMD	**70.51**	**65.41**	**54.77**	**2.01**	**40.46**	**50.60**	**32.85**	**57.83**
Qwen2.5-Omni	Vanilla	80.77	71.20	77.45	3.25	86.49	N/A	N/A	N/A
	FMD	**81.30**	**71.77**	**78.22**	**3.36**	**86.61**	N/A	N/A	N/A

output logits. Notably, FMD provides a unified framework that seamlessly integrates input masking with modality-specific processing in separate decoder branches and merged decoding, making it broadly applicable to diverse AV-LLMs regardless of their fusion strategies.

4 Experiments

We conduct extensive experiments to evaluate the effectiveness of the proposed Fork-Merge Decoding (FMD). We first describe the experimental setup, including baselines, datasets, evaluation protocols, and implementation details (Section 4.1). We then present quantitative results to demonstrate the effectiveness of our approach (Section 4.2), followed by a deeper analysis of FMD (Section 4.3). Qualitative examples are provided in Supp. B.

4.1 Experimental setup

Baselines. We evaluate our approach using three representative AV-LLMs: VideoLLaMA2 (Cheng et al., 2024), video-SALMONN (Sun et al., 2024), and Qwen2.5-Omni (Xu et al., 2025).

Datasets and evaluation protocol. The AVHBench (Sung-Bin et al., 2025) is the first benchmark specifically designed to assess audio-visual hallucinations in AV-LLMs. It comprises four subtasks: audio-driven video hallucination (A→V), video-driven audio hallucination (V→A), audio-visual matching (AV matching), and audio-visual captioning (AV captioning). The AVQA (Yang et al., 2022) dataset contains 57,000 YouTube videos for evaluating real-world audio-visual understanding. MUSIC-AVQA (Li et al., 2022) offers 45,867 QA pairs from 9,288 music performance videos, focusing on fine-grained audio-visual reasoning such as identifying sound sources and temporally aligning auditory and visual cues. AVUT (Yang et al., 2025) and WorldSense (Hong et al., 2025) datasets evaluate video comprehension with an emphasis on auditory cues and general multimodal video understanding, respectively.

As VideoLLaMA2 use fixed sparse sampling (e.g., 8 frames), we filter AVUT and WorldSense to include only videos under 60 seconds to preserve information density. For the layer-wise analysis in Section 4.3, we curate a validation set comprising 200 samples from each AVHBench task (A→V, V→A, and AV matching), reserving the remaining data for evaluation. For the AVHBench dataset, the three binary (yes/no) tasks except AV captioning, we report classification accuracy. For AVQA, MUSIC-AVQA, AVUT, WorldSense and AV captioning, which involve open-ended responses, we follow the GPT-assisted evaluation protocol from the official VideoLLaMA2 implementation[1].

Implementation details. For attention-guided fusion, we set the weighting parameter α as described in Section 3.1, using $N = 100$ randomly sampled examples from the AVHBench validation set: $\alpha = 0.8$ for VideoLLaMA2 and $\alpha = 0.9$ for Qwen2.5-Omni. The number of layers used for the *fork* phase L_{fork} is chosen to be roughly one-seventh of the total decoding layers: the 4th layer for VideoLLaMA2 and Qwen2.5-Omni (28 layers) and the 6th layer for video-SALMONN (40 layers). The rationale for these *fork* layer selections is further analyzed in Section 4.3.

4.2 Quantitative analysis

Comparison with vanilla decoding. To verify the effectiveness of our proposed FMD method, we evaluate it with VideoLLaMA2, video-SALMONN, and Qwen2.5-Omni on five datasets: AVHBench, AVQA, MUSIC-AVQA, AVUT, and WorldSense. As shown in Table 1, applying FMD consistently improves performance over vanilla de-

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

Table 2: **Comparison of decoding methods on AVHBench dataset using VideoLLaMA2.** We compare vanilla decoding, DoLa, VCD, SID and two FMD variants (Gaussian noise injection and zero-out masking). Zero-out masking achieves the highest overall accuracy.

Decoding	Designed for	AVHBench		
		A→V	V→A	AV Mat.
Vanilla	-	80.02	77.03	57.75
DoLa	LLM	69.34	63.44	48.33
VCD	VLM	75.96	69.67	52.52
SID	VLM	78.53	72.82	53.52
FMD w/ noise	AV-LLM	79.17	**78.03**	57.76
FMD w/ zero-out	AV-LLM	**80.45**	77.52	**59.01**

coding, which represents the original model inference, across all tasks without any training. Notably, the gains are more pronounced in the AV captioning task (evaluated on a 5-point scale), which requires generating long and sophisticated answers, with relative improvements ranging from 3.4% to 9.8% across models. Among the models, video-SALMONN benefits the most, achieving a remarkable 12.26% increase on the AVQA dataset. This is particularly significant because video-SALMONN has not been trained on AVQA or MUSIC-AVQA, yet FMD still enhances its zero-shot inference. These results highlight the robustness and strong generalizability of our approach.

Comparison with other decoding methods. We compare our FMD with other test-time decoding strategies that operate at the logit level. The evaluation is conducted using the VideoLLaMA2 model on the AVHBench dataset. Specifically, the comparison includes the following methods: DoLa (Chuang et al., 2024), VCD (Leng et al., 2024b), SID (Huo et al., 2025), and FMD variants with Gaussian noise injection and zero-out masking. **DoLa** contrasts intermediate-layer outputs with final predictions to factually correct outputs. **VCD** reduces language bias by injecting Gaussian noise into the visual input and subtracting the resulting logits from the original ones. **SID** adopts a similar approach with VCD but preserves the least informative visual tokens based on attention weights, contrasting their logits with those of the original outputs. For DoLa, we follow the original setup and extract outputs from the same intermediate layer reported in the paper. For VCD and SID, which were originally developed for vision-language models (VLMs), we adapt their procedures to handle both audio and video modalities.

As shown in Table 2, applying decoding strategies developed for LLMs or VLMs to AV-LLMs leads to degraded performance. This highlights the need for decoding methods that are specifically tailored to the unique characteristics of AV-LLMs, which differ from those of unimodal or bimodal models. Additionally, injecting Gaussian noise into the inputs within FMD (denoted as FMD with noise) results in lower performance on A→V and AV matching tasks compared to vanilla decoding, although it does lead to improved performance on the V→A task. We attribute this to the inability of Gaussian noise to equally isolate modality-specific information, as further discussed in Supp. A.2. In contrast, zero-out masking yields consistent performance improvements across all tasks, demonstrating that the proposed FMD design is suitable for audio-visual understanding.

4.3 Further analysis

Analysis of attention and model performance across different merge layers. To validate the suitability of the *fork* layer L_{fork} for subsequent merging, we further analyze the attention weight distribution across layers, as illustrated in Figure 3. We find that deeper *merge* positions lead to reduced attention to visual tokens and increased attention to audio tokens. However, forcing equal attention to video and audio can be detrimental, as it disrupts the inherent alignment of the pretrained model. To investigate this, we analyze task performance across various *merge* layers in Figure 4, using the AVHBench validation set. Extended results with larger sample sizes are provided in Supp. A.4, exhibiting consistent trends. We observe that as the *merge* layer becomes deeper, the performance on A→V and AV matching tasks decreases, while performance on the V→A task improves. This implies that overly low visual dependence can hinder the interpretation of visual information. Overall, performing the *fork* operation in the early layers achieves more balanced performance while largely preserving the original model characteristics. We propose a pipeline where forking occurs early and merging starts in the middle layers, promoting unimodal feature enhancement first, followed by cross-modal interaction and reasoning in later stages.

Ablation on audio-visual fusion strategy. To demonstrate the effectiveness of our attention-guided fusion, we compare it with three alternative fusion strategies on the AVHBench dataset using the VideoLLaMA2 model in Table 3.

In Eq. 3, the aggregation of masked and un-

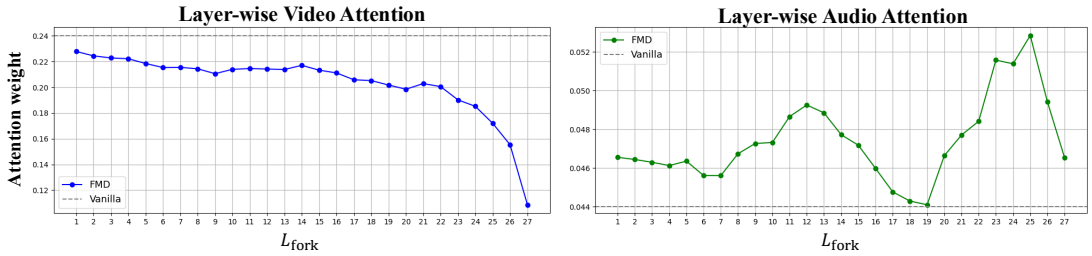


Figure 3: **Layer-wise attention weight comparison on VideoLLaMA2 using the AVHBench validation set.** We analyze the attention weights from the final token in the last decoder layer, focusing on the distribution across video and audio segments. Deeper merging reduces attention to visual tokens while increasing attention to audio tokens.

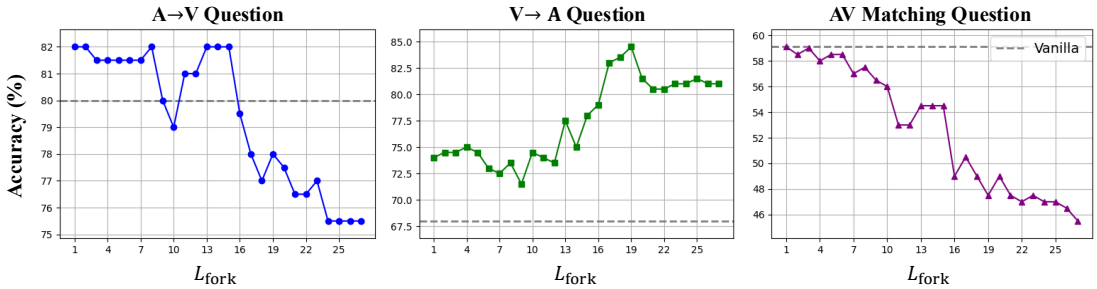


Figure 4: **Layer-wise ablation results on VideoLLaMA2, evaluated on the AVHBench validation subset across distinct tasks.** To verify the suitablity of the selected *fork* layer L_{fork}, we evaluate performance across three tasks, each focused on a certain modality: A→V for video-targeted understanding, V→A for audio-targeted understanding, and AV matching for joint audio-visual understanding.

Table 3: **Fusion strategy ablation in Eq. 3 with VideoLLaMA2.** Attention-guided fusion (Fixed) balances masked and unmasked inputs, improving performance.

Methods	AVHBench		
	A→V	V→A	AV Mat.
Vanilla	80.02	77.03	57.75
Exclusion	79.38	76.79	54.83
Average	75.11	55.31	57.16
Attention (Adaptive)	79.49	72.78	**59.19**
Attntion (Fixed)	**80.45**	**77.52**	59.01

masked features is controlled by the fusion weight α. **Exclusion** corresponds to setting $\alpha = 1$, where the masked modality is entirely excluded. This leads to a performance drop compared to vanilla decoding. We attribute this to the causal nature of autoregressive models: fully ignoring one modality can disrupt the flow of information from previous tokens to the next prediction. **Average** fusion, where $\alpha = 0.5$, also results in degraded performance, likely because it gives equal weight to informative signals and noisy features. Using an adaptive α for each input, referred to as **Attention (Adaptive)**, also causes performance drops on A→V and V→A tasks compared to vanilla decoding. Moreover, obtaining the adaptive α necessitates two additional full forward passes for the masked branches in the *fork* stage, leading to a 2.89× increase in inference latency compared to vanilla decoding. This indicates that the model is not robust to continuously varying α values and, and given the added inference cost, the approach is also inefficient. By contrast, a fixed α derived from the AVHBench validation subset not only enables efficient inference without requiring full forward pass to calculate last-layer attention, but also generalizes well to other datasets, as proven in Table 1. These results highlight the effectiveness of the attention-guided fusion strategy. Further analysis on the robustness of α estimation against varying sample sizes is provided in Supp. A.5, showing that even as few as 10 samples are sufficient to obtain stable values.

Decoding speed comparison. To validate the efficiency of FMD, we compare its inference speed against three representative test-time decoding methods, SID (Huo et al., 2025), VCD (Leng et al., 2024b) and DoLA (Chuang et al., 2024). We apply each decoding method to VideoLLaMA2 and measure the time required to generate a single token, reporting both the latency in seconds per token and the relative speed normalized to vanilla decoding as shown in Table 4. The results are obtained on 100 examples from the AVHBench dataset.

SID and VCD exhibit relatively slow decoding speed. Since they require two full forward passes, one on the original input and one on the modality-corrupted input to contrast the resulting logits, the

Table 4: **Decoding speed comparison.** FMD achieves faster inference since only the initial L_{fork} layers require forked processing.

Decoding	Latency (sec/token)↓	Relative↓
Vanilla	0.34	1
SID	0.71	2.09
VCD	0.69	2.03
DoLa	0.54	1.59
FMD (Ours)	**0.43**	**1.26**

computational cost nearly doubles. DoLA achieves faster inference than SID and VCD by leveraging intermediate layer outputs to refine the model's predictions, thereby mitigating the inefficiency of repeated full forward passes. Our proposed FMD achieves the fastest inference speed among the four methods. This efficiency stems from the fact that only the initial *fork* layers require dual forward passes, making FMD not only computationally efficient but also more effective than prior decoding methods originally developed for LLMs or VLMs, as evidenced by its superior performance reported in Table 2. Additional experiments analyzing how inference speed varies with deeper L_{fork} settings are provided in Supp. A.6.

5 Related Works

Audio-visual large language models. Building upon the success of LLMs, there has been a surge of interest in extending their capabilities to incorporate audio and visual modalities, with text serving as the central modality. ChatBridge (Zhao et al., 2023b) presents a text-centric modality bridging framework trained on limited paired data, whereas models such as PandaGPT (Su et al., 2023), ImageBind-LLM (Han et al., 2023), and OneLLM (Han et al., 2024) leverage unified encoders to accommodate various modalities. Other approaches (Chen et al., 2023b; Lu et al., 2024; Lyu et al., 2023; Panagopoulou et al., 2023; Zhan et al., 2024; Zhang et al., 2023) employ modality-specific encoders to better capture distinct feature spaces. To enhance spatial-temporal modeling across modalities, CAT (Ye et al., 2024) introduces a clue aggregator for cross-modal reasoning, VideoLLaMA2 (Cheng et al., 2024) utilizes a spatial-temporal convolutional connector for video synchronization, and video-SALMONN (Sun et al., 2024) proposes a multi-resolution causal Q-Former for audio-visual fusion. Meerkat (Chowdhury et al., 2024) further refines multimodal interactions by aligning audio and visual signals at multiple levels through interaction modules prior to decoding. Recently, Qwen2.5-Omni (Xu et al., 2025) is introduced as a model that perceives audio-visual inputs, generates text and natural speech, and achieves superior performance in audio-visual tasks.

Inference-time reasoning enhancement with LLMs. Recent work explores inference-time strategies to enhance LLM reasoning without additional training. Chain-of-Thought (CoT) elicits intermediate reasoning steps and has been extended to VLMs via structured textual representations (Himakunthala et al., 2023; Ni et al., 2024; Zhu et al., 2023a) or modular reasoning pipelines (Sun et al., 2025a; Xu et al., 2024; Yang et al., 2023), improving interpretability and robustness. In parallel, Contrastive Decoding (CD) refines decoding by comparing token-level logits between a weaker and a stronger model (Li et al., 2023b), while DoLA (Chuang et al., 2024) contrasts early and late layer outputs within a single model. VCD (Leng et al., 2024b) extends CD to VLMs by injecting Gaussian noise into image inputs and contrasting the biased predictions with the originals. Related approaches (Kim et al., 2024; Wang et al., 2024) leverage self-descriptions or distorted instructions, and SID (Huo et al., 2025) preserves the least informative visual tokens to contrast their influence on predictions. However, these inference-time methods are largely developed for VLMs, leaving AV-LLMs underexplored and motivating inference strategies tailored to the unique challenges of audio-visual inputs.

6 Conclusion

We identify and systematically analyze modality bias in current AV-LLMs, where jointly processing audio and visual inputs can hinder balanced reasoning. To address this, we propose Fork-Merge Decoding (FMD), a simple, training-free, efficient, and model-agnostic inference strategy that separates modality-specific processing in the early decoder layers (the *fork* phase) and merges their representations in later layers (the *merge* phase). FMD consistently improves performance on tasks requiring integrated multimodal understanding, as demonstrated across five audio-visual benchmarks using three representative AV-LLMs. Furthermore, our analysis of attention distributions confirms that FMD effectively relieves the observed modality bias. As a plug-and-play solution, our approach is broadly applicable to AV-LLMs, enabling deeper unimodal and multimodal understanding.

Limitations

While the fusion weight α can be calibrated with as few as $\sim$10 unlabeled samples per model, it remains a global choice that does not account for question-dependent shifts in modality importance. For example, a question such as *"What sound does the object in the video make?"* inherently demands stronger reliance on audio, whereas *"What is the person in the video wearing?"* shifts the balance toward video—yet our current formulation applies the same averaged α across all inputs. Extending FMD with instance-adaptive fusion, where α is conditioned on the input question or sample, is a natural next step toward more dynamic cross-modal reasoning.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv*.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, and 1 others. 2022. Flamingo: a visual language model for few-shot learning. In *Proc. NeurIPS*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. In *Proc. NeurIPS*.

Keqin Chen, Zhao Zhang, Weili Zeng, Richong Zhang, Feng Zhu, and Rui Zhao. 2023a. Shikra: Unleashing multimodal llm's referential dialogue magic. *arXiv*.

Sihan Chen, Handong Li, Qunbo Wang, Zijia Zhao, Mingzhen Sun, Xinxin Zhu, and Jing Liu. 2023b. Vast: A vision-audio-subtitle-text omni-modality foundation model and dataset. In *Proc. NeurIPS*.

Zesen Cheng, Sicong Leng, Hang Zhang, Yifei Xin, Xin Li, Guanzheng Chen, Yongxin Zhu, Wenqi Zhang, Ziyang Luo, Deli Zhao, and 1 others. 2024. Videollama 2: Advancing spatial-temporal modeling and audio understanding in video-llms. *arXiv*.

Sanjoy Chowdhury, Sayan Nag, Subhrajyoti Dasgupta, Jun Chen, Mohamed Elhoseiny, Ruohan Gao, and Dinesh Manocha. 2024. Meerkat: Audio-visual large language model for grounding in space and time. In *Proc. ECCV*.

Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James Glass, and Pengcheng He. 2024. Dola: Decoding by contrasting layers improves factuality in large language models. In *Proc. ICLR*.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. 2023. Instructblip: Towards general-purpose vision-language models with instruction tuning. In *Proc. NeurIPS*.

Tianrui Guan, Fuxiao Liu, Xiyang Wu, Ruiqi Xian, Zongxia Li, Xiaoyu Liu, Xijun Wang, Lichang Chen, Furong Huang, Yaser Yacoob, and 1 others. 2024. Hallusionbench: an advanced diagnostic suite for entangled language hallucination and visual illusion in large vision-language models. In *Proc. CVPR*.

Jiaming Han, Kaixiong Gong, Yiyuan Zhang, Jiaqi Wang, Kaipeng Zhang, Dahua Lin, Yu Qiao, Peng Gao, and Xiangyu Yue. 2024. Onellm: One framework to align all modalities with language. In *Proc. CVPR*.

Jiaming Han, Renrui Zhang, Wenqi Shao, Peng Gao, Peng Xu, Han Xiao, Kaipeng Zhang, Chris Liu, Song Wen, Ziyu Guo, and 1 others. 2023. Imagebind-llm: Multi-modality instruction tuning. *arXiv*.

Vaishnavi Himakunthala, Andy Ouyang, Daniel Rose, Ryan He, Alex Mei, Yujie Lu, Chinmay Sonar, Michael Saxon, and William Yang Wang. 2023. Let's think frame by frame with vip: A video infilling and prediction dataset for evaluating video chain-of-thought. *arXiv*.

Jack Hong, Shilin Yan, Jiayin Cai, Xiaolong Jiang, Yao Hu, and Weidi Xie. 2025. Worldsense: Evaluating real-world omnimodal understanding for multimodal llms. In *arXiv:2502.04326*.

Rongjie Huang, Mingze Li, Dongchao Yang, Jiatong Shi, Xuankai Chang, Zhenhui Ye, Yuning Wu, Zhiqing Hong, Jiawei Huang, Jinglin Liu, and 1 others. 2024. Audiogpt: Understanding and generating speech, music, sound, and talking head. In *Proc. AAAI*.

Shaohan Huang, Li Dong, Wenhui Wang, Yaru Hao, Saksham Singhal, Shuming Ma, Tengchao Lv, Lei Cui, Owais Khan Mohammed, Barun Patra, and 1 others. 2023. Language is not all you need: Aligning perception with language models. In *Proc. NeurIPS*.

Fushuo Huo, Wenchao Xu, Zhong Zhang, Haozhao Wang, Zhicheng Chen, and Peilin Zhao. 2025. Self-introspective decoding: Alleviating hallucinations for large vision-language models. In *Proc. ICLR*.

Junho Kim, Hyunjun Kim, Yeonju Kim, and Yong Man Ro. 2024. Code: Contrasting self-generated description to combat hallucination in large multi-modal models. In *Proc. NeurIPS*.

Sicong Leng, Yun Xing, Zesen Cheng, Yang Zhou, Hang Zhang, Xin Li, Deli Zhao, Shijian Lu, Chunyan Miao, and Lidong Bing. 2024a. The curse of multi-modalities: Evaluating hallucinations of large

multimodal models across language, visual, and audio. *arXiv*.

Sicong Leng, Hang Zhang, Guanzheng Chen, Xin Li, Shijian Lu, Chunyan Miao, and Lidong Bing. 2024b. Mitigating object hallucinations in large vision-language models through visual contrastive decoding. In *Proc. CVPR*.

Guangyao Li, Yake Wei, Yapeng Tian, Chenliang Xu, Ji-Rong Wen, and Di Hu. 2022. Learning to answer questions in dynamic audio-visual scenarios. In *Proc. CVPR*.

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 *Proc. ICML*.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. 2023b. Contrastive decoding: Open-ended text generation as optimization. In *Proc. ACL*.

Bin Lin, Yang Ye, Bin Zhu, Jiaxi Cui, Munan Ning, Peng Jin, and Li Yuan. 2024. Video-llava: Learning united visual representation by alignment before projection. In *Proc. EMNLP*.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023a. Visual instruction tuning. In *Proc. NeurIPS*.

Yiheng Liu, Tianle Han, Siyuan Ma, Jiayue Zhang, Yuanyuan Yang, Jiaming Tian, Hao He, Antong Li, Mengshen He, Zhengliang Liu, and 1 others. 2023b. Summary of chatgpt-related research and perspective towards the future of large language models. *Meta-Radiology*, page 100017.

Jiasen Lu, Christopher Clark, Sangho Lee, Zichen Zhang, Savya Khosla, Ryan Marten, Derek Hoiem, and Aniruddha Kembhavi. 2024. Unified-io 2: Scaling autoregressive multimodal models with vision language audio and action. In *Proc. CVPR*.

Chenyang Lyu, Minghao Wu, Longyue Wang, Xinting Huang, Bingshuai Liu, Zefeng Du, Shuming Shi, and Zhaopeng Tu. 2023. Macaw-llm: Multi-modal language modeling with image, audio, video, and text integration. *arXiv*.

Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. 2024. Video-chatgpt: Towards detailed video understanding via large vision and language models. In *Proc. ACL*.

Minheng Ni, Yutao Fan, Lei Zhang, and Wangmeng Zuo. 2024. Visual-o1: Understanding ambiguous instructions via multi-modal multi-turn chain-of-thoughts reasoning. *arXiv preprint arXiv:2410.03321*.

Taichi Nishimura, Shota Nakada, and Masayoshi Kondo. 2024. On the audio hallucinations in large audio-video language models. *arXiv preprint*.

Artemis Panagopoulou, Le Xue, Ning Yu, Junnan Li, Dongxu Li, Shafiq Joty, Ran Xu, Silvio Savarese, Caiming Xiong, and Juan Carlos Niebles. 2023. X-instructblip: A framework for aligning x-modal instruction-aware representations to llms and emergent cross-modal reasoning. *arXiv*.

Paul K Rubenstein, Chulayuth Asawaroengchai, Duc Dung Nguyen, Ankur Bapna, Zalán Borsos, Félix de Chaumont Quitry, Peter Chen, Dalia El Badawy, Wei Han, Eugene Kharitonov, and 1 others. 2023. Audiopalm: A large language model that can speak and listen. *arXiv preprint*.

Woomin Song, Seunghyuk Oh, Sangwoo Mo, Jaehyung Kim, Sukmin Yun, Jung-Woo Ha, and Jinwoo Shin. 2024. Hierarchical context merging: Better long context understanding for pre-trained llms. In *Proc. ICLR*.

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

Guangzhi Sun, Wenyi Yu, Changli Tang, Xianzhao Chen, Tian Tan, Wei Li, Lu Lu, Zejun Ma, Yuxuan Wang, and Chao Zhang. 2024. video-salmonn: Speech-enhanced audio-visual large language models. In *Proc. ICML*.

Linzhuang Sun, Hao Liang, Jingxuan Wei, Bihui Yu, Tianpeng Li, Fan Yang, Zenan Zhou, and Wentao Zhang. 2025a. Mm-verify: Enhancing multimodal reasoning with chain-of-thought verification. *arXiv*.

Qi Sun, Marc Pickett, Aakash Kumar Nain, and Llion Jones. 2025b. Transformer layers as painters. In *Proc. AAAI*.

Kim Sung-Bin, Oh Hyun-Bin, JungMok Lee, Arda Senocak, Joon Son Chung, and Tae-Hyun Oh. 2025. Avhbench: A cross-modal hallucination benchmark for audio-visual large language models. In *Proc. ICLR*.

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 *Proc. ICLR*.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, and 1 others. 2022. Lamda: Language models for dialog applications. *arXiv*.

Xintong Wang, Jingheng Pan, Liang Ding, and Chris Biemann. 2024. Mitigating hallucinations in large vision-language models with instruction contrastive decoding. In *Proc. ACL*.

Houjing Wei, Yuting Shi, and Naoya Inoue. 2024. Phase diagram of vision large language models inference: A perspective from interaction across image and instruction. *arXiv preprint*.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2022a. Finetuned language models are zero-shot learners. In *Proc. ICLR*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022b. Chain-of-thought prompting elicits reasoning in large language models. In *Proc. NeurIPS*.

Guowei Xu, Peng Jin, Li Hao, Yibing Song, Lichao Sun, and Li Yuan. 2024. Llava-o1: Let vision language models reason step-by-step. *arXiv*.

Jin Xu, Zhifang Guo, Jinzheng He, Hangrui Hu, Ting He, Shuai Bai, Keqin Chen, Jialin Wang, Yang Fan, Kai Dang, and 1 others. 2025. Qwen2. 5-omni technical report. *arXiv preprint*.

Pinci Yang, Xin Wang, Xuguang Duan, Hong Chen, Runze Hou, Cong Jin, and Wenwu Zhu. 2022. Avqa: A dataset for audio-visual question answering on videos. In *Proc. ACM MM*.

Yudong Yang, Jimin Zhuang, Guangzhi Sun, Changli Tang, Yixuan Li, Peihan Li, Yifan Jiang, Wei Li, Zejun Ma, and Chao Zhang. 2025. Audio-centric video understanding benchmark without text shortcut. In *Proc. EMNLP*.

Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Ehsan Azarnasab, Faisal Ahmed, Zicheng Liu, Ce Liu, Michael Zeng, and Lijuan Wang. 2023. Mm-react: Prompting chatgpt for multimodal reasoning and action. *arXiv*.

Qilang Ye, Zitong Yu, Rui Shao, Xinyu Xie, Philip Torr, and Xiaochun Cao. 2024. Cat: Enhancing multimodal large language model to answer questions in dynamic audio-visual scenarios. In *Proc. ECCV*.

Tianyu Yu, Yuan Yao, Haoye Zhang, Taiwen He, Yifeng Han, Ganqu Cui, Jinyi Hu, Zhiyuan Liu, Hai-Tao Zheng, Maosong Sun, and 1 others. 2024. Rlhf-v: Towards trustworthy mllms via behavior alignment from fine-grained correctional human feedback. In *Proc. CVPR*.

Zhuoran Yu and Yong Jae Lee. 2025. How multimodal llms solve image tasks: A lens on visual grounding, task reasoning, and answer decoding. In *Proc. COLM*.

Jun Zhan, Junqi Dai, Jiasheng Ye, Yunhua Zhou, Dong Zhang, Zhigeng Liu, Xin Zhang, Ruibin Yuan, Ge Zhang, Linyang Li, and 1 others. 2024. Anygpt: Unified multimodal llm with discrete sequence modeling. *arXiv*.

Hang Zhang, Xin Li, and Lidong Bing. 2023. Video-llama: An instruction-tuned audio-visual language model for video understanding. In *Proc. EMNLP*.

Renrui Zhang, Jiaming Han, Chris Liu, Peng Gao, Aojun Zhou, Xiangfei Hu, Shilin Yan, Pan Lu, Hongsheng Li, and Yu Qiao. 2024. Llama-adapter: Efficient fine-tuning of language models with zero-init attention. In *Proc. ICLR*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, and 1 others. 2023a. A survey of large language models. *arXiv*.

Zijia Zhao, Longteng Guo, Tongtian Yue, Sihan Chen, Shuai Shao, Xinxin Zhu, Zehuan Yuan, and Jing Liu. 2023b. Chatbridge: Bridging modalities with large language model as a language catalyst. *arXiv*.

Deyao Zhu, Jun Chen, Kilichbek Haydarov, Xiaoqian Shen, Wenxuan Zhang, and Mohamed Elhoseiny. 2023a. Chatgpt asks, blip-2 answers: Automatic questioning towards enriched visual descriptions. *arXiv*.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. 2023b. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv*.

A Further Analysis

A.1 Analysis of attention weights beyond the merge point

As shown in Figure 1 of the main paper, the final layer of VideoLLaMA2 (Cheng et al., 2024) assigns higher attention weights to video inputs compared to audio inputs. To examine attention trends across the entire model, we visualize the average attention weights after the merge point at L_{fork} in Figure A.1. Specifically, we focus on the last token in the sequence and aggregate attention across layers and heads over video and audio segments to estimate each modality's contribution to the prediction. The proposed FMD consistently produces a more balanced attention distribution between modalities.

A.2 Comparison between gaussian noise addition and zero-out masking

We analyze two input perturbation strategies that aim to suppress modality-specific information by replacing the original input: Gaussian noise injection, as proposed in VCD (Leng et al., 2024b), and zero-out masking, as employed in our method. Specifically, we apply each perturbation method to the video and audio inputs, respectively, and compute the cosine similarity between the final-layer hidden states of the perturbed inputs and those of the original inputs.

As shown in Figure A.2, injecting Gaussian noise into the video input yields hidden states that remain highly similar to those of the original input, indicating that this approach fails to effectively isolate the visual signal. In contrast, when applied to audio inputs, Gaussian noise successfully disrupts the signal, leading to substantial differences in the hidden representations. This observation supports the performance improvement observed in the V→A direction under the FMD with noise setting, as reported in Table 2.

In comparison, zero-out masking completely suppresses both video and audio signals, resulting in hidden states that are clearly separated from those of the original input. This demonstrates that zero-out masking more effectively blocks modality-specific information from being encoded and consistently outperforms vanilla decoding across all tasks, as shown in Table 2.

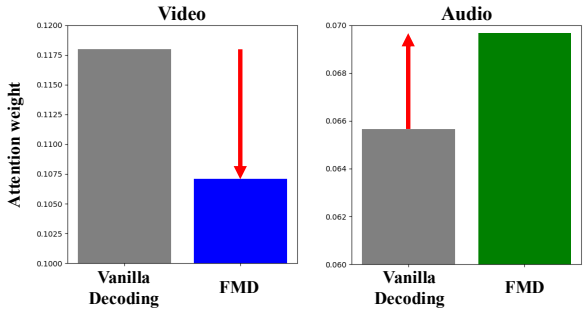


Figure A.1: **Attention weight analysis in VideoLLaMA2 on the AVHBench dataset.** We analyze 100 samples and examine the attention weights in decoder layers after L_{fork}, focusing on the final token. FMD narrows the gap between audio and video attention weights, encouraging more balanced contributions from both modalities.

A.3 Layer Selection for Merge Point

To determine the optimal layer for merging hidden states, we first measure the similarity across hidden states of VideoLLaMA2, following the approach of Sun et al. (2025b). Their study on LLMs shows that layers can typically be organized into 4-5 clusters: 1-2 clusters in the early stage, a large cluster in the middle, and 1-2 clusters in the later stage. In the early layers, modalities are aligned in the feature space, with intra-modal encoding strengthened and inter-modal interaction suppressed. In the middle layers, deeper modality fusion occurs, while the later layers prepare task-specific outputs. As shown in Figure A.3, we observe a similar clustering pattern in VideoLLaMA2. Based on this observation, we select the *fork* layer L_{fork} from the early stage, where feature alignment and intra-modal encoding occur (Wei et al., 2024; Yu and Lee, 2025). This choice aligns with our decoding strategy, which aims to strengthen independent intra-modal representations of vision and audio.

Furthermore, we apply the same cosine-similarity analysis to the channel-wise fusion model, video-SALMONN (Sun et al., 2024). As shown in Figure A.4, video-SALMONN exhibits the same progression observed in the token-wise fusion model: the early layers form one to two clusters, the middle layers collapse into a single cluster as multimodal features are integrated, and the later layers again split into one to two clusters. This consistent early-middle-late pattern supports our choice of L_{fork}, since placing it before middle-layer fusion allows modality-specific representations to be refined before being merged. Based on this anal-

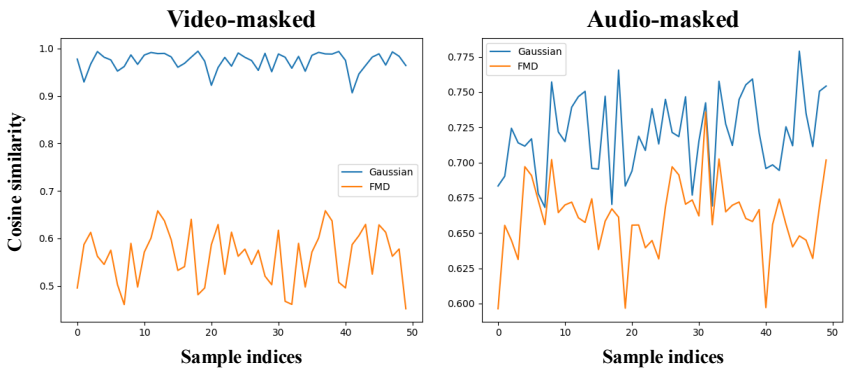


Figure A.2: **Cosine similarity comparison on VideoLLaMA2 using 100 samples from the AVHBench dataset.** We compute the cosine similarity between the final-layer hidden states of the intact input and those of audio- or video-perturbed inputs. The results indicate that video signals are not effectively isolated by additive Gaussian noise, whereas zero-masking reliably suppresses both modalities.

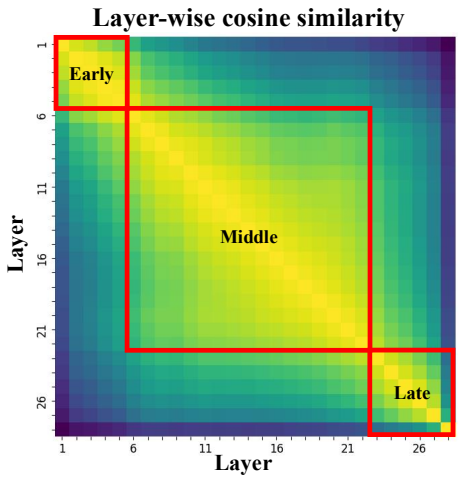


Figure A.3: **Layer-wise hidden state similarity in VideoLLaMA2.** L_{fork} is chosen from the early stage.

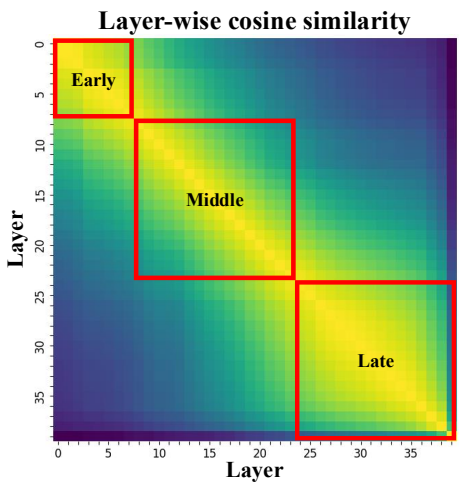


Figure A.4: **Layer-wise hidden state similarity in video-SALMONN.** L_{fork} is chosen from the early stage.

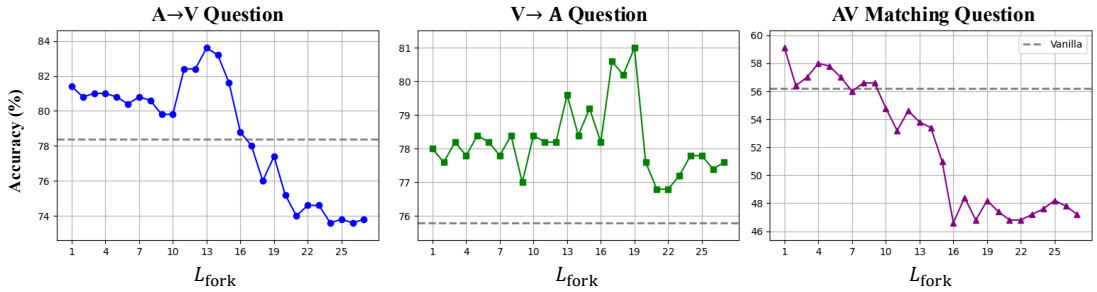
ysis, we select $L_{\text{fork}} = 6$ for video-SALMONN.

A.4 Layer-wise performance across dataset sizes

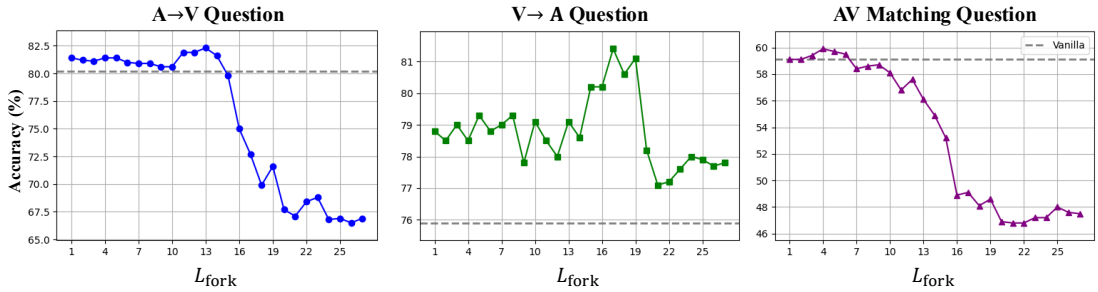
In the main paper (see Section 4.3 and Figure 4), we analyze task performance across L_{fork} using 200 samples from the AVHBench validation set. To examine whether this trend holds with more data, we repeat the analysis with 500 and 1,000 samples per task (see Figure A.5a and Figure A.5b). We observe consistent trends across all sample sizes, demonstrating that L_{fork} can be reliably validated even with a small number of samples.

A.5 Sensitivity of α to sample size

To examine how many samples are required to obtain a reliable estimate of the attention weight α used for attention-guided fusion (Section 3.1), we evaluate its robustness under realistic deployment scenarios with the VideoLLaMA2 model. While the main paper computes α using 100 samples, in practice the available data may be far smaller. To simulate this setting, we first combine examples from the three datasets (AVHBench, AVUT, and WorldSense) into a single pool. From this pool, we randomly sample 10, 50, or 100 examples, repeating each sampling 100 times to assess the stability of α. As reported in Table A.1, the estimated α remains highly stable even when only 10 samples are used, and the variance decreases further with larger subsets. This indicates that our procedure for estimating α is robust and data-efficient, enabling reliable parameter selection even with very limited data in real-world deployment.



(a) Performance on 500 samples from the AVHBench dataset.



(b) Performance on 1,000 samples from the AVHBench dataset.

Figure A.5: **Layer-wise accuracies of VideoLLaMA2 on each task in the AVHBench dataset.** The consistent trends observed across 500 and 1,000 samples further validate the 200-sample analysis presented in Figure 4.

Table A.1: **Stability of α values based on sample size.**

Number of Samples	10	50	100
Mean (μ)	0.829	0.829	0.829
Std (σ)	0.024	0.010	0.007

A.6 Analysis of inference speed across L_{fork}

To better understand the relationship between inference speed and the fork layer L_{fork}, we evaluate FMD on VideoLLaMA2 using 100 randomly selected AVHBench samples across different fork layer settings.

As shown in Table A.2, deeper fork layers require more decoder layers to be forwarded, leading to longer inference time. At the same time, accuracy drops significantly because later forking disrupts the pretrained model attribute as illustrated in Figure 3 and Figure 4. Based on this, we set $L_{fork} = 4$ in VideoLLaMA2 (one out of every seven layers), which yields consistent performance gains while increasing inference time by only 26%.

B Qualitative Results

We provide representative qualitative examples and additional positive and negative cases to better understand the behavior of FMD.

Table A.2: **Analysis between inference speed and fork layer L_{fork} in VideoLLaMA2 on AVHBench.** Deeper fork layers require more decoder computation, increasing latency while significantly reducing accuracy as illustrated in Figure A.5. Setting $L_{fork} = 4$ (one-seventh of the 28 layers) achieves consistent performance gains with only 26% increase in inference time.

L_{fork}	0 (Vanilla)	1	**4**	5	10	15	20	25
Latency (sec/token)	0.34	0.40	**0.43**	0.45	0.51	0.56	0.62	0.68
Relative	1	1.18	**1.26**	1.32	1.5	1.65	1.82	2

B.1 Representative Examples

Audio-visual matching from AVHBench (Figure A.6a). The video shows a man fishing in a stream, and the audio contains the sound of water flowing in a valley with occasional splashes from footsteps. The vanilla decoding predominantly captures the visual content, describing only the presence of a man fishing in a stream. However, it overlooks the acoustic context, failing to reflect the sound of flowing water and splashes. By contrast, FMD generates a more comprehensive description that integrates both modalities, enriching the visual detail (e.g., specifying the fishing rod) and the audio detail (e.g., capturing the sounds of water flow and splashes). This highlights the strength of FMD in balancing and fusing multimodal cues, leading to richer and more faithful outputs.

Audio-visual question & answering from AVQA (Figure A.6b). The video shows a train, but in

(a) An example of audio-visual matching from AVHBench.

(b) An example of audio-visual question & answering from AVQA.

Figure A.6: **Qualitative results with VideoLLaMA2 on AVHBench and AVQA.** Vanilla decoding often relies on a single modality, resulting in incomplete or inconsistent outputs, whereas FMD effectively integrates both audio and visual information to produce more accurate and coherent results.

the early frames its appearance could be confused with a cable car. The accompanying audio contains the sound of a train gradually accelerating as it departs from the station. The vanilla model, relying on the ambiguous visual information, incorrectly answers (A) Cable car. In contrast, FMD correctly identifies the scene as (C) Train by integrating both visual and auditory cues. This illustrates that FMD effectively leverages audio information to resolve visual ambiguity, resulting in more accurate and context-aware answers.

B.2 Additional Positive Cases

Beyond the representative examples, we further analyze audio-driven video hallucinations, video-driven audio hallucinations, and audio-visual matching cases in Figure A.7, Figure A.8, and Figure A.9. We also examine more complex audio-visual description scenarios in Figure A.10. Across these cases, FMD effectively leverages both audio and visual modalities to interpret the context. Notably, even when audio and video are artificially paired, FMD detects inconsistencies and generates correct responses. As shown in Figure A.10, FMD also produces more precise and detailed descriptions of both modalities than vanilla decoding.

B.3 Negative Cases

Although FMD enhances multimodal understanding without additional training, it can occasionally produce inaccurate phrases alongside otherwise detailed and accurate descriptions. For example, in the top case of Figure A.11, FMD captures both visual and audio content, whereas vanilla decoding describes only the visual scene. However, the phrase "while she does it" is temporally incorrect, since the speaker talks before blow-drying. In the bottom example, the mention of "cup" is inaccurate, as no cup is present. These cases suggest that FMD may still produce fine-grained temporal or object-level hallucinations. Nevertheless, FMD generally produces richer and more informative responses by modeling cross-modal relationships, such as capturing "talking," and reduces hallucinations at a broader scale, as reflected in the quantitative results in Table 1. Further investigation into minimizing fine-grained hallucinations remains an important direction for future work.

C Computational Resource

All experiments are conducted on a system equipped with an AMD EPYC 7513 32-Core CPU and a single NVIDIA RTX A6000 GPU. To ensure fair measurement of inference speed across all experiments, we terminate all non-experimental processes during inference time.

D The Use of LLMs

We use LLMs to refine words and sentences for a formal academic writing style and to identify relevant related work. For evaluation, we employ LLMs to assess long-form responses, such as AV captioning from AVHBench, following the protocol of VideoLLaMA2.

E Social Impact

The evolution towards AV-LLMs extends AI capabilities to real-world multimedia, significantly benefiting applications like assistive technology and video analysis. However, their inference mechanisms remain underexplored. Our proposed Fork-Merge Decoding (FMD) addresses this by offering a training-free framework to isolate and ana-

lyze modality-specific reasoning. By enhancing the interpretability of how models process complex audio-visual cues, this work paves the way for more reliable and transparent multimodal systems.

Figure A.7: **Qualitative results for audio-driven video hallucination tasks using VideoLLaMA2.** Compared to vanilla decoding, FMD generates more accurate responses by effectively leveraging both audio and visual modalities.

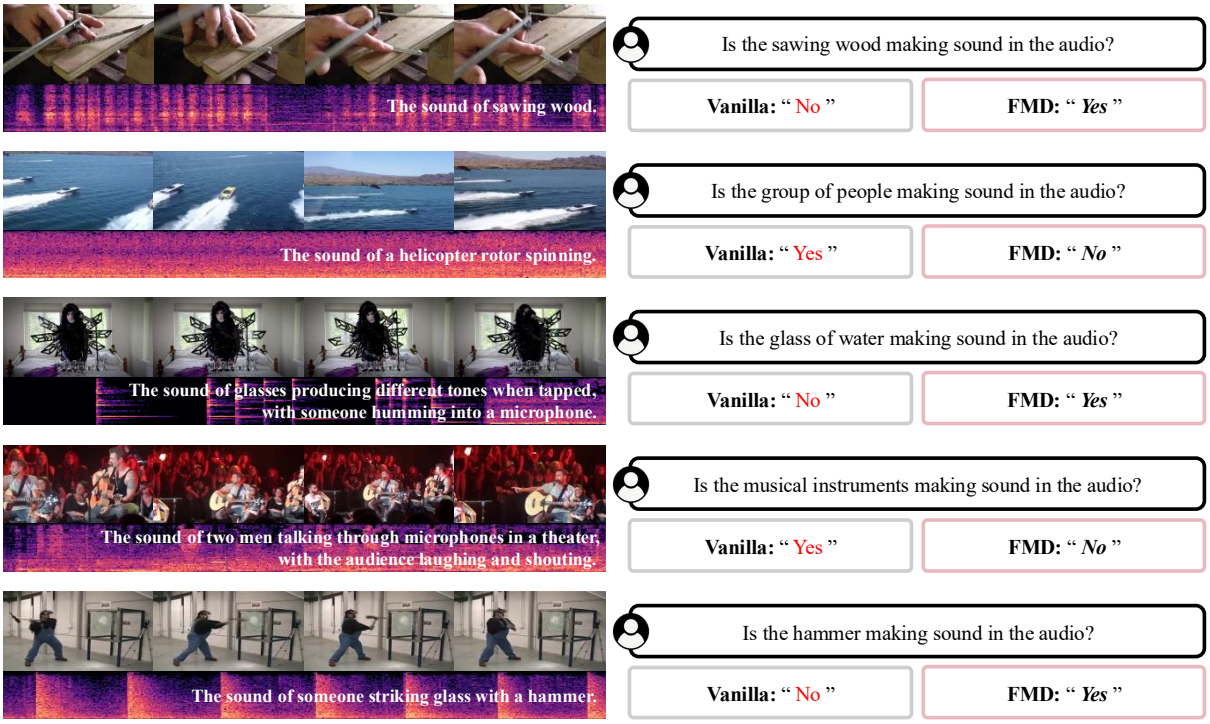
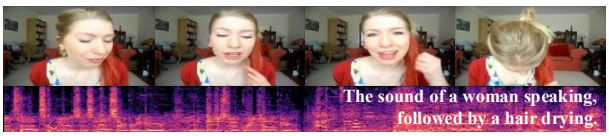


Figure A.8: **Qualitative results for video-driven audio hallucination tasks using VideoLLaMA2.** FMD produces more accurate responses in most cases, including instances where visual signals lead to confusion, as shown in the third example.

Figure A.9: **Qualitative results for audio-visual matching tasks using VideoLLaMA2.** FMD generates correct responses even when the synthesized audio and video inputs are semantically unrelated.

Figure A.10: **Qualitative results for audio-visual description tasks using VideoLLaMA2.** FMD effectively describes both audio and visual content, capturing fine-grained details (bold)—such as "A man is talking about the bus" in the top example.



Describe what you see and hear in detail.

Vanilla: "A woman is blow drying her hair."

FMD: "*A woman with bright orange hair is blow drying her hair. She is smiling and* ***talking*** *while she does it*."

Describe what you see and hear in detail.

Vanilla: "A person is washing a glass in the sink."

FMD: "*A person is washing a cup in a sink while* ***talking.***"

Figure A.11: **Failure case analysis on audio-visual description tasks with VideoLLaMA2 (Cheng et al., 2024).** FMD may produce occasional errors (red), but captures both audio and visual content more effectively and with finer-grained details (bold) than vanilla decoding.