OmniMMI: A Comprehensive Multi-modal Interaction Benchmark in Streaming Video Contexts

Supplementary Material

A. Audio Adaption Analysis

In this section, we further explore the adaptation of our methods to audio speech input. To adapt M4 to receive audio queries, we fine-tuned it on a randomly selected subset of the VoiceAssistant dataset [46], which comprises 30,000 audio instructions. To ensure a fair comparison, we maintained the same hyperparameters and other settings as those used in the tuning of M4. The results are presented in Table 6. Our findings indicate that tuning the model on purely audio instruction data, without incorporating visual data, does not enhance its proactive turn-taking ability. Consequently, we converted the queries in M4-IT to speech using CosyVoice and mixed them with the VoiceAssistant subset used during the tuning of M4-a. After integrating this audio data, we achieved a score of 68.5 on the **PT** task. Overall, the introduction of audio instruction data still limits the performance of tasks requiring both visual and audio inputs. We believe this limitation arises from the lack of mixed visual and audio data during the training phase. In future work, we aim to enhance the model's audio understanding capabilities by incorporating more high-quality multimodal data.

B. Highlight Spot Algorithm

In this section, we present the pseudo-code of our proposed training-free highlight spot algorithm, as illustrated in Algorithm 1. For any transformer-based model, incoming streaming video frames are stored in the KVCache to avoid redundant computations. Subsequently, we compute the attention weights from the model's final layer using the text query as the key and the video as the value. We then identify and save the frame indices whose attention weights exceed a threshold, determined by the mean and variance of the previous attention weights. These indices are labeled as consistently salient frames, signifying the frames that need to be highlighted. The consistency threshold is a hyperparameter, which is set to 4 in our experiments. Furthermore, we introduce an initial latency step to mitigate the challenges associated with calculating the mean and variance; in practice, this latency step is set to 2.

Algorithm 1 Highlight Spot

Require: Video stream V_∞, query q, threshold γ, Gaussian factor α

```
highlight_spot.init()
for all frame v in V_∞ do
    KVCache.update(W_K v, W_V v)
    attn ← SelfAttn(v ⊕ q, KVCache)
    (μ, σ) ← std_mean(attn)
    δ ← μ + α × σ
    cands ← {t; attn[t] > δ}
    for all c_t in cands do
        c_t ← highlight_spot.get(t) + 1
        highlight_spot.update(i, c_i)
    end for
    if highlight_spot.heap is not empty then
        (i, c) ← highlight_spot.peek()
        if freq > γ then
            send(i)
        end if
    end if
end for
```

C. Single Question Analysis of Multi-turn Dependency Reasoning

In this section, we detail the accuracy of each step in the multi-turn dependency reasoning task. The results are presented in Table 7. Unlike the results presented in Table 3, this experiment focuses solely on the accuracy of individual reasoning steps. Generally, we observe a decline in accuracy as the number of steps increases. However, in certain instances, accuracy at a later step exceeds that of a previous one. We attribute this anomaly to potential hallucinations generated by the language models. Overall, there is a significant drop in accuracy across successive steps, underscoring the importance of multi-step reasoning in evaluation. This approach helps to mitigate errors introduced by language models, demonstrating the necessity of a step-by-step evaluation process.

D. Single Question Analysis of Dynamic State Grounding

In this section, we extend our analysis of the Dynamic State Grounding task by examining the performance on each individual question. The results, as detailed in Table 8, indicate a notable decline in performance as the number of states increases. This decline can be attributed to the increased length of the video context and dialogue history, which complicates the process of dynamically grounding the current state to derive the correct answer. Furthermore, our analysis did not reveal significant performance differences across different models at the initial state. However,

Table 6. **Performance comparison of existing OmniLLM on OmniMMI**. The 1st, 2nd, 3rd of **SG** and **MD** tasks represent the cumulative accuracy up to and including these stages. The "avg." indicates average accuracy across all data points.

Models	LLM	Num Frames	SG				AP	MD				SI	PA	PT
			1st	2nd	3rd	avg.		1st	2nd	3rd	avg.			
Commercial Video LLMs														
Gemini-1.5-Pro [36]	-	128	52.33	19.67	9.35	16.33	43.00	35.00	16.26	7.14	12.00	38.50	✗	✗
GPT-4o [32]	-	50	48.67	16.95	5.61	15.00	39.50	34.33	15.57	7.65	12.33	17.00	✗	✗
OmniLLMs														
VideoLLaMA2 [6]	Qwen2-7B	8	41.00	12.88	0.00	10.33	35.00	23.33	4.15	0.51	3.00	5.00	✗	✗
VITA [11]	Mistrl-8×7B	16	8.67	0.00	0.00	0.00	39.00	11.33	3.11	1.52	2.00	1.50	✗	67.00
MiniOmini2 [47]	Qwen2-0.5B	1	17.00	5.08	0.93	4.67	14.00	6.00	1.00	0.00	1.00	1.00	✗	✗
M4 (ours)	Qwen2-7B	32 / 1 fps	35.67	6.44	1.87	5.67	33.5	35.67	6.44	1.87	1.67	9.00	25.50	62.00
M4-a(ours)	Qwen2-7B	32 / 1 fps	28.33	2.37	0.00	2.00	13.00	19.33	3.11	0.51	3.00	7.50	1.50	68.5

Table 7. Multi-turn Dependency Reasoning

Models	Step=1	Step=2	Step=3	Overall
Commercial Video LLMs				
Gemini-1.5-Pro	52.33	34.24	36.45	16.33
GPT-4o	48.67	31.53	20.56	15.00
Open-source Video LLMs				
VideoChatGPT	18.00	13.49	11.22	3.00
VideoChat2	16.33	13.15	12.24	2.67
Video-LLaVA	22.67	13.49	16.33	3.33
LLaMA-VID	21.33	15.22	13.78	2.67
MiniGPT4-Video	12.67	6.57	8.67	1.67
PLLaVA	21.00	13.49	17.35	1.33
LLaVA-NeXT-Video	17.00	10.03	10.71	2.00
ShareGPT4Video	20.33	15.57	14.80	2.00
LLaMA-VID-13B	22.67	14.88	14.29	3.33
PLLaVA-13B	25.67	17.80	16.84	4.33
PLLaVA-34B	18.67	17.30	10.20	3.00
LLaVA-NeXT-Video-DPO-34B	14.67	14.53	12.24	1.67
LongVA	20.67	16.27	13.78	2.33
LongVILA	22.33	14.19	14.29	3.00
LongLLaVA	26.33	18.69	20.41	3.67
VideoLLM-online	11.67	7.27	10.71	1.33
VideoLLaMB	18.67	13.15	17.86	3.00
OmniLLMs				
VideoLLaMA2	23.33	15.92	18.78	5.00
VITA	11.33	12.80	8.63	2.00
MiniOmini2	6.00	3.11	2.03	1.00
M4	19.33	10.73	12.18	1.67

Table 8. Dynamic State Grounding

Models	State=1	State=2	State=3	Overall
Commercial Video LLMs				
Gemini-1.5-Pro	35.00	37.02	38.78	12.00
GPT-4o	34.33	33.56	37.24	12.33
Open-source Video LLMs				
VideoChatGPT	35.33	17.97	10.28	3.33
VideoChat2	19.67	14.23	6.54	2.33
Video-LLaVA	32.00	16.27	11.21	1.67
LLaMA-VID	29.67	13.56	7.48	2.33
MiniGPT4-Video	25.00	15.25	14.02	4.00
PLLaVA	37.33	13.56	10.29	3.33
LLaVA-NeXT-Video	30.33	12.20	6.54	3.00
ShareGPT4Video	34.00	13.22	10.28	2.00
LLaMA-VID-13B	33.33	14.24	6.54	1.33
PLLaVA-13B	41.33	13.90	12.15	2.67
PLLaVA-34B	29.00	14.24	10.28	3.67
LLaVA-NeXT-Video-DPO-34B	30.33	11.19	5.61	2.67
LongVA	33.33	15.59	8.41	3.33
LongVILA	39.00	16.95	14.02	4.33
LongLLaVA	36.33	11.53	7.48	3.33
VideoLLM-online	18.00	13.56	5.61	4.67
VideoLLaMB	32.67	14.58	10.28	2.33
Open-source Video LLMs				
VideoLLaMA2	41.00	26.78	10.28	10.33
VITA	8.67	8.14	2.80	0.00
MiniOmini2	17.00	14.92	10.28	4.67
M4	35.67	13.22	6.54	5.67

the performance gap widens as the number of states increases, underscoring the importance of a model's ability to handle longer contexts while maintaining effective grounding capabilities.

E. Annotation Details

E.1. Raw Video Data Collection

To enhance our dataset, we specifically collect data from YouTube, concentrating primarily on videos that are particularly commonly useful in our real-life. We also focus on the videos which are in content involving personal introductions and interpersonal interactions.

E.2. Annotation Tool

The front-end interface for human annotation is depicted in Figure 7. In this interface, each question or statement is associated with the most relevant time span, which serves either as part of the label or as an aid for subsequent annotation tasks.

E.3. Annotation Guidelines

To ensure that annotators produce high-quality annotations that align with our specified standards, we provide detailed guidelines, including examples of various question types.

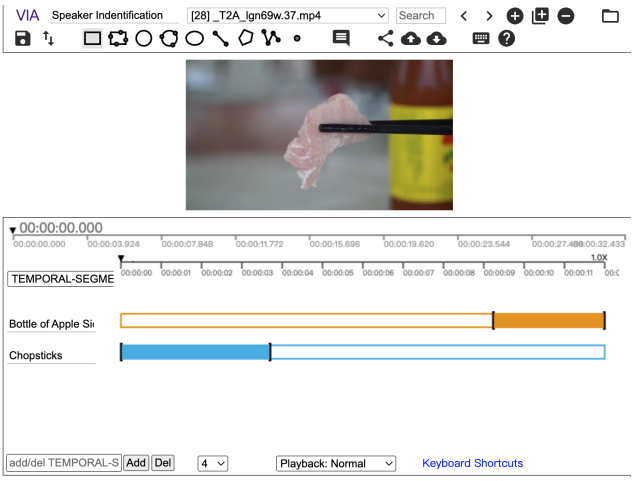

Figure 7. The Front-End Interface for Human Annotation

Category	Question
Object State	How many objects are in the scene?
	How many people are in the room?
	What is the color of the car?
	Is the door open or closed?
Spatial Relations	Where is the cat relative to the chair?
Dynamic Spatial Relations	Is the person walking towards or away from the camera?
	Where is the ball relative to the player?
Action State	What is the person doing?
	What activity is happening in the scene?
Scene State	Is the room well-lit or dim?
	What is the weather like?
	Is the street busy or quiet?
	What is the context of the scene?
Human Object Interaction	Is the person holding the book?
Human Human Interaction	Are the two people shaking hands?
	What is the interaction between the two characters?
Group Dynamics	How are the group members interacting?
Emotional State	What is the person's emotional state?
Audio/Speech State	What does the speaker mentioned?

Table 9. Annotation hints for annotators including category and example question.

The list of hints is demonstrated in Table 9.

F. M4-IT Construction Details

F.1. Noise data prompt

We employ GPT-4o to autonomously generate noise data for the purpose of instruction tuning. The prompt utilized for the generation of noise data is detailed below.

> *You are a sophisticated AI designed to simulate human-like conversation by generating 'noise.' This noise consists of naturally flowing statements that mimic the user's perspective. —— Review the user's questions and the assistant's responses carefully. Using this information, create coherent declarative statements that reflect the user's voice. These should resemble everyday human dialogue and do not require a response from the assistant. Ensure your output is in the form of declarative sentences and avoid questions. Keep the noise brief and in casual, conversational English. But do NOT need response*

F.2. Stop Words

We compile a set of frequently used stop words to incorporate into our instructional data, thereby serving as the designated stop words: "That's a good point, and", "Let me stop you there", "Just a second", "I don't mean to be rude, but", "If I could interject", "Pardon me, but", "Sorry to interrupt", "Before you continue", "Can we pause for a moment?", "May I add something here?", "I apologize for cutting in", "Could I stop you for a second?", "I'd like to add", "Could I clarify something?", "I have a quick question", "This reminds me of", "Let me add to that", "Can I share my thoughts?", "Hold on a moment", "One moment, please", "Allow me to explain", "Excuse me", "Can I jump in for a moment?", "I see what you mean, but", "I think it's important to mention"

G. M4 Implementation Details

Hyperparam	M4
α	2
β	0.2
γ	4
Model Max Length	32000
Learning Rate	1e-5
Warmup Ratio	0.03
Per Device Batch Size	1
Gradient Accumulation Steps	4
Epoch	1

Table 10. Hyperparameters for M4.

In practice, we conduct the training process using four Nvidia A800 GPUs, which requires approximately one hour to fine-tune the model. Table 10 presents a detailed account of the hyperparameters employed during both the training and inference procedures.