StreamForest: Efficient Online Video Understanding with Persistent Event Memory

Xiangyu Zeng[*1,2], **Kefan Qiu**[*1], **Qingyu Zhang**[*1], **Xinhao Li**[1], **Jing Wang**[1], **Jiaxin Li**[1]
Ziang Yan[3,2], **Kun Tian**[4], **Meng Tian**[5], **Xinhai Zhao**[4], **Yi Wang**[2], **Limin Wang**[†1,2]
[1]Nanjing University [2]Shanghai AI Laboratory [3]Zhejiang University
[4]Noah's Ark Lab, Huawei [5]Yinwang Intelligent Tech.

https://github.com/MCG-NJU/StreamForest

Abstract

Multimodal Large Language Models (MLLMs) have recently achieved remarkable progress in video understanding. However, their effectiveness in real-time streaming scenarios remains limited due to storage constraints of historical visual features and insufficient real-time spatiotemporal reasoning. To address these challenges, we propose **StreamForest**, a novel architecture specifically designed for streaming video understanding. Central to StreamForest is the Persistent Event Memory Forest, a memory mechanism that adaptively organizes video frames into multiple event-level tree structures. This process is guided by penalty functions based on temporal distance, content similarity, and merge frequency, enabling efficient long-term memory retention under limited computational resources. To enhance real-time perception, we introduce a Fine-grained Spatiotemporal Window, which captures detailed short-term visual cues to improve current scene perception. Additionally, we present **OnlineIT**, an instruction-tuning dataset tailored for streaming video tasks. OnlineIT significantly boosts MLLM performance in both real-time perception and future prediction. To evaluate generalization in practical applications, we introduce **ODV-Bench**, a new benchmark focused on real-time streaming video understanding in autonomous driving scenarios. Experimental results demonstrate that StreamForest achieves the state-of-the-art performance, with accuracies of 77.3% on StreamingBench, 60.5% on OVBench, and 55.6% on OVO-Bench. In particular, even under extreme visual token compression (limited to 1024 tokens), the model retains 96.8% of its average accuracy in eight benchmarks relative to the default setting. These results underscore the robustness, efficiency, and generalizability of StreamForest for streaming video understanding.

1 Introduction

In recent years, multimodal large language models have made significant progress in video understanding tasks, demonstrating strong semantic comprehension and reasoning capabilities across videos of varying durations and scenarios [30, 52, 31, 35]. Benefiting from large-scale pretraining and enhanced cross-modal modeling capabilities, these models have been widely adopted in various domains [28, 42, 55, 59]. However, with the growing demand for real-time intelligent processing in online applications such as autonomous driving [49], live video streaming [6], and robotics [79], researchers have increasingly shifted their focus from conventional offline video understanding to the more challenging task of streaming video processing [5, 23, 46].

[*] Equal contribution.
[†] Corresponding author.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

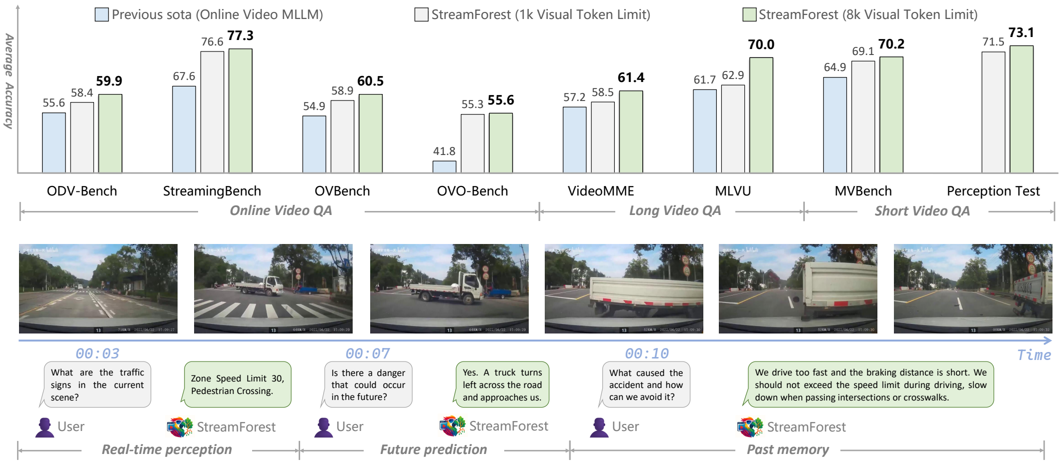


Figure 1: StreamForest achieves strong performance across various evaluation benchmarks while using significantly fewer visual tokens. It effectively handles key tasks in streaming video scenarios, including past memory, real-time perception, and future prediction.

In the field of streaming video understanding, efficiently caching continuously arriving video frame features remains a long-standing and challenging problem. To mitigate the storage and computational overhead associated with past frames, prior work has primarily employed two strategies for visual feature reduction: compression during sampling [5, 46, 64] and compression during storage [52, 72, 23]. Compression during sampling reduces a large portion of incoming visual features, which severely limits the model's capacity for fine-grained spatiotemporal reasoning. As a result, it can only perform coarse semantic summarization of the current scene. Conversely, compression during storage typically involves merging or discarding adjacent frames based on inter-frame similarity. While more memory-efficient, this strategy is susceptible to missing critical foreground actions due to background noise. It may also result in excessive local merging, introducing spatiotemporal irregularities that degrade the model's ability to retain and reason about key events over time.

To address the challenges of streaming video understanding, we propose a novel architecture called **StreamForest**. At its core is the **Persistent Event Memory Forest**, a mechanism designed to efficiently store and manage long-term visual information. This memory system enables a MLLM to process ultra-long streaming video at a constant rate of 1 fps by dynamically organizing video segments into a tree structure based on event boundaries. The merging of segments is guided by three penalty functions that consider temporal distance, content similarity, and merge frequency, ensuring an adaptive and meaningful memory hierarchy. To enhance real-time perception, we introduce a **Fine-grained Spatiotemporal Window**, which extracts rich local spatiotemporal features from nearby frames. This module enables the MLLM to better understand the current scene by focusing on temporally relevant visual context. We also present **OnlineIT**, a fine-tuning dataset specifically designed for streaming video understanding. OnlineIT improves the MLLM's ability to perceive the present moment and anticipate future events by leveraging both recent observations and long-term historical cues. It addresses the problem of hallucinations caused by spatiotemporal distribution shifts between past and current frames. In addition, we introduce **ODV-Bench**, a new benchmark for evaluating streaming video understanding in autonomous driving scenarios. ODV-Bench emphasizes real-time perception and future prediction, providing a systematic framework for assessing the generalization and real-world effectiveness of streaming video MLLMs in downstream tasks.

We conducted extensive experiments on both online and offline video understanding benchmarks to validate the effectiveness of StreamForest. Under the default setting with a visual token limit of 8192, StreamForest significantly outperforms previous state-of-the-art streaming video understanding MLLMs. It achieves an average accuracy of 77.3% on StreamingBench, 60.5% on OVBench, and 55.6% on OVO-Bench. StreamForest also matches or surpasses the performance of leading offline video understanding MLLMs on both long and short video benchmarks, despite operating in a streaming video input setting. Moreover, StreamForest demonstrates strong resilience under extreme compression. With a reduced visual token limit of just 1024, it retains 96.8% of its average performance across eight benchmarks compared to the default setting. These results highlight the robustness and efficiency of our approach in continuously processing streaming video input.

2 Related work

Multimodal Large Language Model. Extending multimodal capabilities from static images to dynamic video sequences introduces additional complexity, requiring models to possess stronger abilities in modeling long-range dependencies and understanding events [30, 40, 81, 41, 71, 58]. Recent advances in MLLMs for video have introduced a variety of innovative strategies to tackle the challenges of efficiently processing and reasoning over long video inputs [37, 42, 29, 51, 33]. LongVILA [7] proposes a Multimodal Sequence Parallelism system for long-context modeling, enabling efficient parallel training and inference on extended video content. However, most current research on video understanding remains focused on offline settings [57, 69, 62, 34], where the model has full access to the complete video sequence before inference. Although this setting facilitates global semantic modeling, it falls short in streaming scenarios, where real-time understanding of continuously evolving scenes is required. Therefore, the development of models specifically designed for online video understanding is of critical importance.

Streaming Video Understanding. In real-world applications, users increasingly expect MLLMs to support online processing and real-time interaction. This demand has prompted growing interest in the task of streaming video understanding. Recently, several works have explored this emerging area [5, 72, 63, 11, 61, 23]. However, most existing streaming video understanding approaches are primarily designed for streaming dense video captioning [5, 60, 32, 47, 12], focusing solely on summarizing semantic content from visual frames. As a result, they struggle to handle essential tasks such as memory recall and real-time perception, which are critical for comprehensive streaming video understanding. Moreover, in pursuit of computational efficiency, many methods apply aggressive compression to video frame sequences [46, 74, 64], making them unsuitable for complex and dynamic tasks that require fine-grained and real-time spatiotemporal understanding, such as autonomous driving. To address these limitations, our goal is to develop a more generalizable and practical approach for online video understanding. It emphasizes fine-grained spatiotemporal features at the moment of query and supports persistent memory storage based on events.

3 Methodology

3.1 Streaming Video Understanding Architecture: StreamForest

In this section, we detail our proposed StreamForest. Specifically, the core design of StreamForest lies in Fine-grained Spatiotemporal Window and Persistent Memory Forest, which work in tandem to enable the model to retain long-term memories of past events while supporting real-time perception.

3.1.1 Fine-grained Spatiotemporal Window

To meet the real-time spatiotemporal perception requirements of streaming video understanding, we introduce the Fine-grained Spatiotemporal Window (FSTW). We observe that in practical applications, most of the clues requiring fine-grained spatiotemporal reasoning are concentrated near the time of the question. Therefore, we retain only second-level short-term fine-grained spatiotemporal features.

Specifically, the FSTW consists of two components: real-time perception and short-term spatiotemporal memory. Real-time perception directly samples high-resolution visual features from the current frame, which are encoded with spatiotemporal positional information. As new frames arrive, older frames are compressed along the spatial dimension and transferred into short-term spatiotemporal memory. At the same time, the model computes inter-frame similarity between new and old frames to enable subsequent event-level segmentation. The short-term spatiotemporal memory maintains a frame sequence with a duration of t_s seconds. When its capacity is exceeded, overflowing visual features are offloaded into the Persistent Event Memory Forest. We segment continuous visual features into meta-events by identifying the position with the local minimum inter-frame similarity in the frame sequence. This ensures that each meta-event captures a coherent spatiotemporal transition. A meta-event is treated as an independent node, which consists of a collection of visual tokens from similar consecutive frames. These nodes form the foundation of the MLLM's long-term memory.

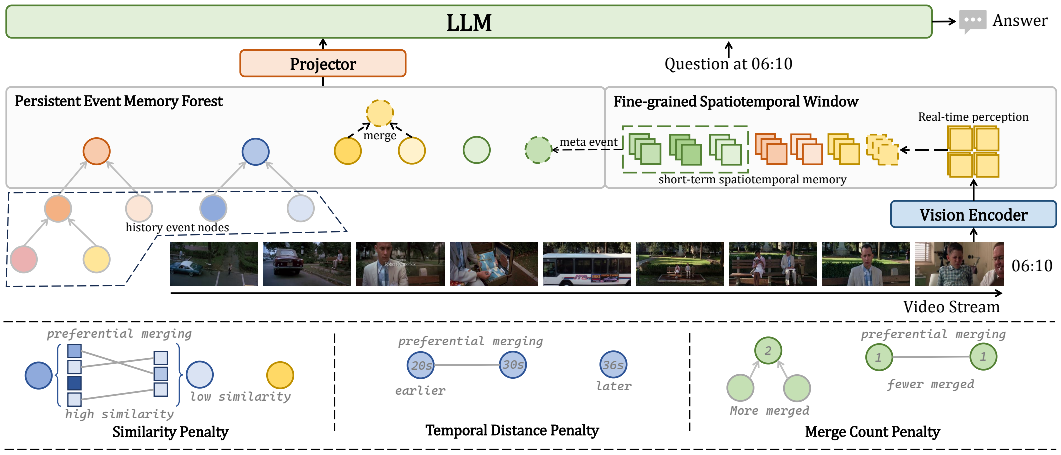


Figure 2: Overview of our proposed StreamForest. The Fine-grained Spatiotemporal Window captures instance-level spatiotemporal features, while the Persistent Event Memory Forest adaptively organizes event-level representations into a set of tree structures. Dashed arrows and feature tokens illustrate potential operations performed during each memory update iteration.

3.1.2 Persistent Event Memory Forest

To efficiently process continuously arriving video frame features in streaming scenarios, we propose the Persistent Event Memory Forest (PEMF), a memory architecture specifically designed to support long-term memory in the context of streaming video. Unlike prior methods that rely on direct inter-frame similarity compression [52] or static memory hierarchies [23], PEMF adaptively compresses and organizes video information at the event level. It constructs a hierarchical, tree-structured memory guided by three penalty functions, enabling the model to retain semantically rich and non-redundant content while managing memory efficiently as it evolves over time. To control memory growth, we impose an upper limit L_q on the number of long-term memory tokens stored in PEMF. When this limit is exceeded, PEMF performs hierarchical memory consolidation by adaptively merging adjacent event nodes into single nodes within the tree structure. The selection of nodes for merging is guided by three penalty functions that account for temporal distance, content similarity, and merge frequency, ensuring that the memory remains both informative and compact.

Similarity Penalty. In long videos, adjacent video segments often exhibit high visual similarity, resulting in substantial feature redundancy. Therefore, we introduce a similarity penalty that encourages the merging of event nodes with highly similar visual content. Due to differences in event durations, two candidate event nodes (denoted as x_i, x_{i+1}) may contain different numbers of visual tokens. To handle this discrepancy, inspired by ToMe [3], we adopt a bipartite graph matching approach. Specifically, we treat the visual features of the two event nodes as sets in a bipartite graph and compute pairwise similarities between tokens across these sets. Let $X_i \in \mathbb{R}^{n_i \times d}$ denote the visual token features of the event node x_i, where n_i is the number of tokens in x_i. We compute the pairwise cosine similarity matrix $S_i = sim(X_i, X_{i+1}) \in \mathbb{R}^{n_i \times n_{i+1}}$, and select the top k_i highest similarity scores, corresponding to the most similar token pairs between two event nodes. The similarity penalty P_s is defined as one minus the average of these k_i highest similarity scores:

$$P_s(x_i, x_{i+1}) = 1 - \frac{1}{k_i} \sum_{(p,q)\in\mathcal{T}_i} S_i^{(p,q)}, \quad \mathcal{T}_i = \text{argTopK}_{(p,q)}(S_i^{(p,q)},\ k_i). \tag{1}$$

Merge Count Penalty. When event nodes repeatedly participate in tree-structured hierarchical memory merging, their visual details may gradually degrade due to accumulated information loss. This degradation can lead to local spatiotemporal inconsistencies, ultimately impairing the accuracy of long-term video understanding. To mitigate this issue, we introduce a merge count penalty as a regularization term. It penalizes overly merged nodes and encourages a more balanced memory integration process, thereby preserving the fidelity of each event representation. Let c_i denote the historical merge count of the event node x_i, with its maximum value at the query time denoted as

c_{max}. We define the merge count penalty P_m as follows:

$$P_m(x_i, x_{i+1}) = \frac{c_i + c_{i+1}}{2c_{max}}. \tag{2}$$

Temporal Distance Penalty. In real-world streaming video understanding scenarios, frames that are temporally closer to the current query time often carry more relevant information. This observation suggests that recent visual features should be preserved with higher fidelity, while historical features can be compressed more aggressively. To implement this intuition, we introduce a temporal distance penalty, which encourages the model to retain more detailed representations of temporally proximate events while promoting the forgetting of details from distant past events. Let t_q denote the current query or interaction time, t_i denote the time of event i. The calculation of t_i is detailed in the Appendix A. We define the time penalty P_t as follows:

$$P_t(x_i, x_{i+1}) = 1 - \frac{d_i + d_{i+1}}{2}, \quad d_i = \frac{t_q - t_i}{t_q}. \tag{3}$$

Overall penalty. We incorporate the above three penalties to guide the adaptive merging process of event nodes in the PEMF, where the combination of these three factors determines the merge priority of event node pairs.

$$P(x_i, x_{i+1}) = w_s P_s(x_i, x_{i+1}) + w_m P_m(x_i, x_{i+1}) + w_t P_t(x_i, x_{i+1}). \tag{4}$$

The penalty weights w_s, w_m, and w_t collectively determine the behavior of PEMF. When only the similarity penalty is applied, the strategy degenerates into similarity-based compression. Using the merge count penalty alone leads to behavior similar to uniform downsampling. When the temporal distance penalty is used in isolation, the method approximates FIFO. By adjusting these penalty weights, our method enables a flexible trade-off among these strategies, allowing it to adapt effectively to various streaming tasks, enabling a balance between efficient storage saving and the retention of task-relevant information across diverse real-world scenarios.

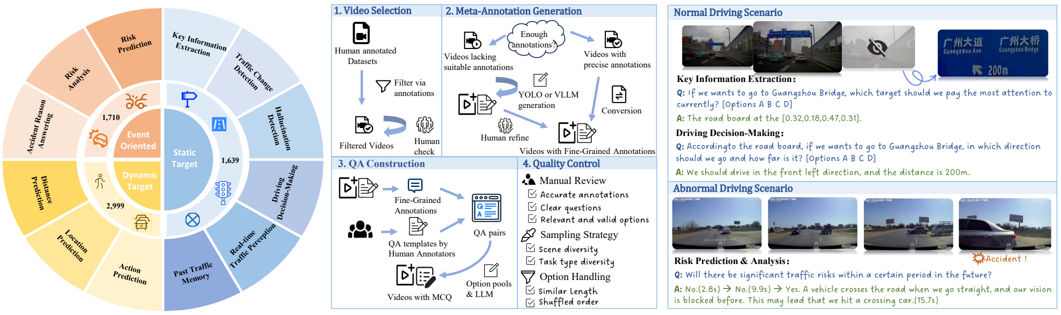
The nodes selected for merging are determined by identifying the pair with the lowest overall penalty score. We employ ToMe [3] for the merging process, compressing the number of visual tokens to half the total tokens of the selected node pair. Upon receiving a user query, the visual features of all root nodes in PEMF, along with all visual features stored in FSTW, are fed into the LLM to support real-time, streaming interaction.

3.2 Instruction-tuning Dataset: OnlineIT

Existing offline long video datasets often exhibit distributional bias, where the key evidence for answering questions is typically concentrated in the middle of the video. As a result, MLLMs fine-tuned on such data tend to overemphasize historical content, potentially leading to hallucinations in accurately interpreting the current moment. Although some datasets for streaming video understanding have been released [5, 60, 63, 47], they remain limited in terms of data volume, quality, and task diversity. To address these limitations, we construct OnlineIT, a training dataset specifically designed for streaming video understanding. OnlineIT focuses on fine-grained event comprehension and real-time spatiotemporal understanding in streaming settings, and it significantly enhances the performance of MLLMs on streaming video understanding tasks.

OnlineIT-general. Based on criteria of diversity, length, and difficulty, we curated and refined several existing high-quality fine-tuning datasets of streaming video understanding [23, 60, 47]. Building upon these, we further developed two new datasets comprising 32K high-quality streaming training instances. This dataset features a larger scale, broader distribution, and greater task diversity, facilitating the learning of more generalizable streaming video representations.

OnlineIT-drive. It includes 89K streaming QA training instances from autonomous driving scenarios. This dataset is designed to enhance MLLMs' performance on complex, real-time downstream tasks. Specifically, by integrating scene semantics, traffic regulations, and common driving events, we extract key elements from driving scenes and video clips to generate a question-answer dataset grounded in autonomous driving contexts. OnlineIT-Drive primarily covers four areas: (1) real-time localization and semantic awareness, (2) understanding of static traffic entities, (3) understanding of dynamic traffic entities, and (4) risk event and accident assessment.



Key Information Extraction:

Q: If we wants to go to Guangzhou Bridge, which target should we pay the most attention to currently? [Options A B C D]

A: The road board at the [0.32,0.18,0.47,0.31].

Driving Decision-Making:

Q: Accordingto the road board, if we wants to go to Guangzhou Bridge, in which direction should we go and how far is it? [Options A B C D]

A: We should drive in the front left direction, and the distance is 200m.

Risk Prediction & Analysis:

Q: Will there be significant traffic risks within a certain period in the future?

A: No.(2.8s) → No.(9.9s) → Yes. A vehicle crosses the road when we go straight, and our vision is blocked before. This may lead that we hit a crossing car.(15.7s)

(a) Category Distribution of ODV-Bench

(b) Generation pipeline of ODV-Bench

(c) Task examples in ODV-Bench

Figure 3: (a) The distribution of task types and the number of QA pairs. (b) The detailed pipeline for constructing the ODV-Bench. (c) Typical task examples in ODV-Bench.

4 ODV-Bench

Many existing benchmarks for streaming scenarios are derived from offline video evaluation datasets [39, 23, 36, 61], and may not adequately reflect real-world applications of streaming video understanding. Although some of them already incorporate Ego4D videos of daily activities [36, 61], these evaluation samples primarily evaluate MLLMs' ability to perceive static scenes and narrate human-environment interactions in a stepwise manner. In contrast, autonomous driving presents dynamic, high-stakes environments with rapidly changing scenes, complex multi-agent interactions (vehicles, pedestrians, and traffic signals), and demanding prediction tasks (such as risk assessment and motion planning). These scenarios require models to balance long-term event memory with fine-grained short-term perception to avoid accidents and make timely decisions. To address this gap, we introduce ODV-Bench, a benchmark specifically designed for online video understanding in autonomous driving scenarios.

4.1 Task Formulation

As shown in Figure 3 (a), we first explore the key traffic elements in autonomous driving scenarios and summarize them into three categories of task scenarios: **(1) Static-target-oriented tasks**, which involve the recognition and retrieval of stationary traffic elements such as traffic signs, lights, and road indicators; **(2) Dynamic-target-oriented tasks**, which focus on behavior and trajectory prediction of dynamic road participants such as vehicles and pedestrians; and **(3) Event-oriented tasks for multi-agent interaction**, which capture complex interactions, risk scenarios, and accidents involving multiple agents. Next, guided by temporal cues and the practical needs of driving, we further define fine-grained task types based on these categories to comprehensively assess model understanding in realistic online driving video scenarios. For more details on task formulation, please refer to Appendix C.1.

4.2 Benchmark Construction

The construction process of ODV-Bench is illustrated in Figure 3 (b). We adopt a four-stage approach to ensure the quality of each generated question, and then present some typical task examples across different driving scenarios in Figure 3 (c).

Data Collection. (1) Video Selection. To align with real-world driving scenarios, we first curated 6 datasets [24, 14, 82, 66, 73, 4] from different task scenarios within the autonomous driving domain, from normal driving to unexpected events. Then, we designed a semi-automatic pipeline that primarily relies on annotation filtering and YOLO-based detection [25], supplemented by manual inspection, to select task-relevant videos from the collected dataset. **(2) Meta-Annotation Generation.** To obtain meta-annotations with detailed spatiotemporal and semantic information, we developed tailored methods based on existing dataset annotations. For well-annotated datasets, we effectively convert existing labels into task-specific meta-annotations. For others, we design a semi-automatic pipeline that begins with coarse annotations generated by VLLM and YOLO [25], followed by structured human verification to ensure quality.

Table 1: **Evaluation results on ODV-Bench.** Our model significantly outperforms state-of-the-art offline and online video MLLMs under zero-shot testing conditions, and achieves further improvements after fine-tuning on driving-domain data.

Method	**Size**	**#Frames**	**Static Target**							**Dynamic Target**				**Event Oriented**				**Overall**
			RTP	HD	KIE	TCD	DDM	PTM	Avg.	AP	LP	DP	Avg.	RP	RA	ARA	Avg.	
Human																		
Human Agents	-	-	96.8	97.6	98.2	95.7	95.9	94.4	95.9	83.7	87.9	90.4	88.2	91.9	94.9	93.0	92.5	91.4
Open-source Offline Video MLLMs																		
MiniCPM-V2.6 [65]	7B	64	20.0	**87.8**	15.1	49.1	26.4	20.6	27.3	71.2	73.4	47.2	60.0	**73.4**	33.3	16.7	53.6	49.8
LongVA [75]	7B	64	29.9	7.3	37.7	47.3	**38.0**	33.6	31.8	66.6	58.6	**50.9**	56.6	57.5	58.1	46.2	56.7	50.2
LLaVA-Onevision [28]	7B	64	36.0	4.9	22.6	60.0	31.4	39.0	34.2	53.6	70.3	47.4	55.1	57.9	72.2	47.4	62.2	51.6
InternVL2.5 [8]	8B	32	40.1	16.3	37.7	52.7	30.4	40.9	37.2	64.1	**84.6**	49.5	**62.5**	54.0	60.6	50.6	56.1	54.2
VideoChat-Flash [33]	7B	256	29.6	15.5	45.3	**76.4**	26.1	36.1	32.2	**73.5**	75.3	47.2	61.0	67.1	64.8	46.2	**64.3**	54.4
Qwen2.5-VL [2]	7B	1fps	**51.8**	8.1	**79.3**	49.1	36.0	**57.3**	**48.3**	50.4	82.6	46.9	57.5	47.6	**78.6**	**52.6**	59.4	**55.6**
Open-source Online Video MLLMs																		
Flash-VStream [72]	7B	1fps	25.4	1.6	11.3	50.9	36.0	22.1	24.8	25.5	39.8	47.2	40.2	32.4	48.6	30.1	38.1	35.7
Dispider [47]	7B	1fps	31.1	7.3	34.0	63.6	34.0	35.4	32.5	43.2	73.1	45.8	52.7	38.2	55.4	36.5	44.3	45.2
VideoChat-Online [23]	4B	1fps	36.9	0.8	62.3	49.1	21.5	47.0	36.1	70.2	86.7	46.4	62.9	51.2	69.4	45.5	57.4	54.5
StreamForest	**7B**	**1fps**	51.4	15.5	54.7	56.4	**38.6**	65.3	51.5	**72.6**	83.2	46.0	62.3	60.2	73.3	47.4	63.8	59.9
StreamForest (FT-drive)	**7B**	**1fps**	**70.1**	**17.1**	**100.0**	**60.0**	32.7	**83.6**	**64.6**	64.0	**96.6**	**59.6**	**70.7**	**71.8**	**93.4**	**58.3**	**78.5**	**71.2**

QA Construction. (1) MCQ Generation. To enable efficient automatic QA generation, we first design accurate and diverse templates tailored to each defined task. These templates are then populated with fine-grained and precise annotations to generate high-quality QA pairs. Next, we develop a multiple-choice generation pipeline based on an option pool, introducing plausible yet misleading distractors alongside the correct answer to ensure the realism and effectiveness of choices. **(2) Quality Control.** To ensure benchmark quality, we first conduct multiple rounds of manual review to verify the clarity and accuracy of QA pairs and the plausibility of distractor options. Besides, to enhance scene diversity and task-type balance, we apply a sampling strategy that allocates questions proportionally to video length, maximizing coverage across scenarios.

5 Experiments

Implementation Details. We adopt SigLiP-so400M[70] as the visual encoder, use an MLP as the projection head, and we employ Qwen2-7B as the LLM. By default, the number of visual tokens is capped at 8192. Among these, 729 tokens are allocated to real-time perception, while short-term spatiotemporal memory consists of 18 frames, each represented by 128 visual tokens. We set the penalty weights for similarity, merge count, and temporal distance to 0.4, 0.4, and 0.2, respectively. The model is trained on 32 A100 GPUs using our proposed OnlineIT dataset, supplemented with offline video data from VideoChat-Flash [33] and LLaVA-Video [76], as well as image data from LLaVA-OneVision [28]. We adopt a five-stage training strategy to train StreamForest from scratch. The first three stages follow the training paradigm of offline long video MLLMs [33]. The fourth stage performs streaming video fine-tuning to yield the base StreamForest. In addition, an optional fifth stage can be incorporated by training with the OnlineIT-Drive, which yields the StreamForest(FT-drive). During the evaluation phase, we constrain the model to process streaming frames at 1 FPS. For more detailed implementation specifics, please refer to the Appendix D.

5.1 Online Benchmark Results

We evaluate the performance of our model on four benchmarks for online video question answering: ODV-Bench, StreamingBench [39], OVBench [23], and OVO-Bench [36]. These benchmarks follow streaming video QA scenarios, where the VideoLLMs must process only the video content available before the current timestamp.

ODV-Bench. It closely integrates spatiotemporal information to comprehensively evaluate MLLMs' ability to understand fine-grained details in online videos and to make future predictions based on both historical and current context in autonomous driving scenarios. The benchmark includes tasks such as identifying subtle objects or actions, describing object positions and spatial relations, and

Table 2: **Comparison of our method with existing approaches on video question answering tasks across various scenarios.** Our approach significantly outperforms previous methods on streaming video understanding benchmarks, while maintaining strong and competitive performance on both long and short video understanding.

		Online Video			**Long Video**		**Short Video**	
Method	**Size**	**StreamingBench**	**OVBench**	**OVO-Bench**	**VideoMME**	**MLVU**	**MVBench**	**PerceptionTest**
		Real-Time All	Avg	Overall	w/o sub.	M-Avg	Avg	Val
Open-source Offline Video MLLMs								
InternVL2 [9]	8B	63.7	48.7	50.1	54.0	64.0	65.8	-
LongVA [75]	7B	60.0	43.6	-	52.6	56.3	-	-
LLaVA-OneVision [28]	7B	71.1	49.5	52.9	58.2	64.7	56.7	57.1
Qwen2-VL [55]	7B	69.0	49.7	52.7	63.3	-	67.0	66.9
LongVU [50]	7B	-	-	48.5	60.6	65.4	66.9	-
LLaVA-Video [76]	7B	-	-	53.1	63.3	70.8	58.6	67.9
Open-source Online Video MLLMs								
VideoLLM-online [5]	8B	36.0	9.6	12.8	-	-	-	-
MovieChat [52]	7B	-	30.9	-	38.2	-	55.1	-
Flash-VStream [72]	7B	23.2	31.2	33.2	-	-	-	-
VideoChat-Online [23]	4B	-	54.9	-	52.8	-	64.9	-
Dispider [47]	7B	67.6	-	41.8	57.2	61.7	-	-
StreamForest	**7B**	**77.3**	**60.5**	**55.6**	**61.4**	**70.0**	**70.2**	**73.1**
StreamForest (FT-drive)	**7B**	**76.8**	**61.6**	**55.6**	**61.9**	**69.6**	**68.6**	**71.6**

forecasting object trajectories. These tasks require strong real-time spatiotemporal perception and contextual understanding. As shown in Table 1, StreamForest achieves an average accuracy of 59.9% on ODV-Bench without being trained on OnlineIT-drive and further improves to 71.2% after training on it. This significantly outperforms all existing online and offline MLLMs, demonstrating the strong generalization capability of our method to downstream streaming video understanding tasks. These results highlight its potential for real-world applications.

StreamingBench & OVBench & OVO-Bench. As shown in Table 2, StreamForest demonstrates strong performance across existing open-source streaming video understanding benchmarks. It achieves an accuracy of 77.3% on StreamingBench, 60.5% on OVBench, and 55.6% on OVO-Bench. These impressive results highlight the robustness of StreamForest in a wide range of online video understanding scenarios. The superior performance of our model can be attributed to two key architectural innovations. First, the Fine-grained Spatiotemporal Window enables precise spatial perception and responsive short-term temporal modeling, which are critical for real-time perception and forward responding tasks. Second, the Persistent Event Memory Forest adaptively organizes long-term visual content into a structured and efficient memory forest, significantly enhancing the MLLM's ability to retain and reason over past events. Together, these two modules offer complementary capabilities that allow our model to handle dynamic, long-horizon streaming video inputs effectively, while maintaining high contextual coherence.

5.2 Offline Benchmark Results

We further evaluate our method on two long video understanding benchmarks (VideoMME[16] and MLVU[77]) and two short video datasets (MVBench[31] and PerceptionTest[45]). In the offline setting, the entire video is provided as input to the MLLM. We sample video frames at 1 FPS, with a maximum limit of 2048 frames. For videos exceeding this limit, frames are uniformly sampled across the entire duration. As shown in Table 2, our method demonstrates superior performance on both long and short video understanding tasks compared to recent state-of-the-art online Video MLLMs. In addition, it outperforms leading offline models in most benchmarks, achieving 61.4% on VideoMME, 70.0% on MLVU, 70.2% on MVBench, and 73.1% on PerceptionTest. This strong performance in offline scenarios highlights the robust generalization capability of our proposed method.

5.3 Ablations

Effectiveness of the Persistent Event Memory Forest: We replace the proposed PEMF with several methods used in previous work. To ensure a fair comparison, we keep the visual token budgets consistent across all methods and fine-tune each model accordingly. The ablation results

Table 3: Comparison between our proposed PEMF and other commonly used memory strategies.

Memory Policy	OVBench	OVO-Bench	MLVU
	Avg	Overall	M-Avg
Uniform Sampling	58.2	52.7	69.4
First In First Out	58.7	52.9	56.7
Similarity Merge [52]	60.3	53.4	68.0
Pyramid Memory Bank [23]	60.3	53.9	68.2
PEMF (Ours)	**60.5**	**55.6**	**70.0**

Table 4: Ablation study on the key components of StreamForest.

Model	OVBench	OVO-Bench	MLVU
	Avg	Overall	M-Avg
w/o FSTW & PEMF	58.0	52.5	51.8
w/o FSTW	59.1	53.7	69.4
w/o PEMF	58.9	53.5	56.6
w/o Event	59.4	52.6	69.1
Ours	**60.5**	**55.6**	**70.0**

are shown in Table 3. The FIFO strategy shows the worst performance. This is especially evident on the long-video benchmark MLVU (56.7% vs. 70.0%), where the method fails due to unfiltered discarding of historical visual features. OVBench primarily emphasizes short-term, fine-grained spatiotemporal perception. Uniform sampling reduces the resolution of recent visual information, which is crucial for real-time understanding (58.2% vs. 60.5% on OVBench). Similarity Merge achieves performance comparable to our PEMF on OVBench (60.3% vs. 60.5%). However, its limitations become clear in tasks that require persistent memory and long-horizon reasoning. On OVO-Bench, PEMF outperforms Similarity Merge by +2.2%, and on MLVU by +2.0%. This is because similarity-based merging may over-merge frames within local video segments, potentially leading to spatiotemporal irregularities and the loss of local event-level representations. The pyramidal memory bank maintains memory through frame replacement. However, fixed capacity limits its ability to capture long-range spatiotemporal features (53.9% vs. 55.6% on OVO-Bench and 68.2% vs. 70.0 on MLVU). In contrast, our method evaluates each visual event based on event-level similarity, merge count, and temporal distance. Then it performs memory consolidation at the event level. This strategy supports efficient and persistent maintenance of historical visual features.

Effectiveness of the Overall Architecture: We conduct ablation studies on three key architectural components. Specifically, we ablate the Fine-grained Spatiotemporal Window and the Persistent Event Memory Forest, while ensuring that the total number of visual tokens remains consistent with the original configuration. In addition, we replace event-based node construction with a frame-based approach. As shown in Table 4, removing both modules leads to the most significant performance drop. Using either FSTW or PEMF alone improves performance compared to the baseline, but the best results are achieved when both components are integrated (+2.5% on OVBench, +3.1% on OVO-Bench, and +18.2% on MLVU). This joint ablation confirms that FSTW and PEMF provide complementary benefits. FSTW enhances real-time spatiotemporal perception near the query timestamp, while PEMF supports efficient and persistent long-term memory, together yielding the strongest overall performance. Moreover, event-level node construction effectively prevents over-merging within events, enabling the compression of visual features at the level of complete visual events rather than individual frames.

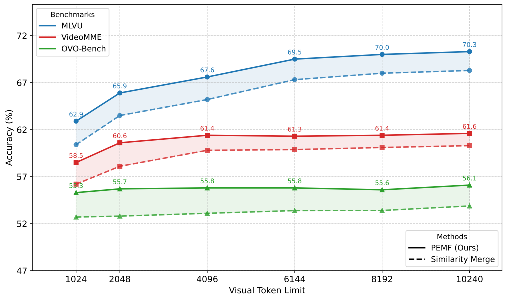


Figure 4: Performance under varying visual token budgets.

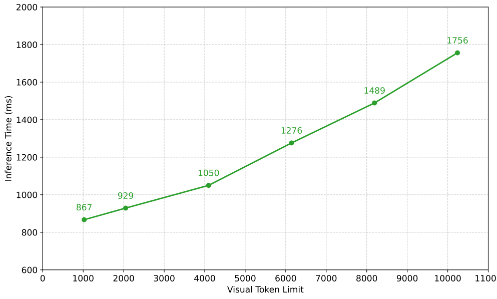


Figure 5: Average inference time under varying visual token budgets (on single A100 GPU).

Robustness to Different Visual Token Budgets: We evaluate the robustness of our method under varying budget constraints for visual tokens, ranging from 1K to 10K. Figure 4 illustrates the performance variation on the benchmarks MLVU, VideoMME, and OVO-Bench under these settings. Notably, under the strict constraint of only 1K visual tokens, StreamForest achieves an average

Table 5: Quantitative analysis of runtime and memory usage of StreamForest.

Input Frames	Memory (GB)	FLOPs (T)	Latency (s)
64	15.8	93.1	0.776
256	17.1	134.1	1.126
1024	17.2	137.3	1.497

Table 6: Runtime comparison of PEMF with other memory mechanisms for 500 frames.

Method	Vis. Encode (s)	Mem. Update (s)	LLM infer (s)	Total (s)
Similarity Merge [52]	5.198	0.183	1.388	6.769
PMB [23]	5.203	0.451	1.381	7.035
PEMF (Ours)	5.218	**0.172**	1.394	6.784

visual compression ratio of up to 99.8% on long video benchmarks. Despite this extreme level of compression, the model still maintains competitive performance, which strongly demonstrates the robustness of our approach in persistently preserving long-term, event-level visual memories. We also conduct a direct comparison between our PEMF and the Similarity Merge. The results clearly demonstrate two core benefits of our approach. First, PEMF exhibits superior absolute performance, consistently outperforming Similarity Merge across all token budgets with an average accuracy improvement of 2-3%. Second, our method shows stronger resilience to extreme compression. Under the most severe 1K token budget, PEMF retains a higher fraction of its full-budget performance, achieving a notable +1.8% relative retention advantage on VideoMME. These experimental results confirm that the performance gains stem from the intrinsic design of PEMF, which adaptively consolidates event-level memory to preserve semantically salient information, thereby ensuring both high accuracy and token efficiency under stringent resource constraints. Figure 5 presents the average inference time of StreamForest under different budgets of visual tokens. The fast and stable inference speed highlights the practical applicability of StreamForest.

Computational cost: We provide a quantitative analysis of the runtime and memory usage of StreamForest. To isolate the effect of the memory mechanism, we assume that frame-level visual features are already extracted by the vision encoder in real time. As shown in Table 5, PEMF enforces a strict upper bound of visual tokens (8K here), ensuring that memory usage remains stable ($\tilde{1}$7 GB) regardless of the number of processed frames. Consequently, the FLOPs and inference latency do not grow significantly even with longer streaming inputs. In addition, we compare PEMF with other memory mechanisms, including the Pyramid Memory Bank [23] and the Similarity Merge strategy [52]. As summarized in Table 6, the overall runtime is dominated by vision encoding (5.2s for 500 frames, $\sim$95 FPS). The memory update of PEMF is extremely lightweight (0.172s for 500 frames), which is negligible compared to the efficiency gains achieved by significantly reducing the total number of visual tokens.

Impact of Training Data: Table 7 presents the impact of our training strategy that integrates both online and offline datasets. The results clearly demonstrate that combining OnlineIT with existing offline VideoQA datasets significantly improves performance on streaming video understanding benchmarks. OnlineIT is specifically designed for real-time perception and future prediction in streaming scenarios, effectively mitigating hallucinations caused by inconsistencies between historical spatiotemporal context and the current moment.

Table 7: The contribution of our training data to performance on the streaming video understanding benchmark. O.general refers to OnlineIT-general, while O.drive refers to OnlineIT-drive.

Data	ODV-Bench	OVBench	OVO-Bench
	Avg	Avg	Overall
base	56.3	53.9	53.5
base + O.general	59.9	60.5	**55.6**
base + O.general + O.drive	**71.2**	**61.6**	**55.6**

6 Conclusions

In this work, we have proposed StreamForest, a novel architecture for streaming video understanding that addresses the limitations in long-term memory and fine-grained perception. By introducing the Persistent Event Memory Forest, our method effectively manages historical visual information through adaptive merging guided by temporal distance, content similarity, and merge count penalties. Coupled with the Fine-grained Spatiotemporal Window, the model maintains a precise understanding of the current scene. We also present OnlineIT, a streaming video understanding fine-tuning dataset that mitigates spatiotemporal shift issues and enhances real-time perception and reasoning. As well as ODV-Bench, a new benchmark tailored for real-time autonomous driving scenarios. Extensive experiments demonstrate that StreamForest not only outperforms state-of-the-art streaming video MLLMs but also rivals top offline video MLLMs under strict streaming input settings, showcasing its robustness and practical value in real-time streaming video understanding applications.

Acknowledgement

This work is supported by the National Key R&D Program of China (No. 2022ZD0160900), the Basic Research Program of Jiangsu (No. BK20250009), and the Collaborative Innovation Center of Novel Software Technology and Industrialization.

References

[1] Anthropic. Claude 3.5 sonnet, 2024. URL `https://www.anthropic.com/news/claude-3-5-sonnet`.

[2] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

[3] Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. *arXiv preprint arXiv:2210.09461*, 2022.

[4] Zhengping Che, Guangyu Li, Tracy Li, Bo Jiang, Xuefeng Shi, Xinsheng Zhang, Ying Lu, Guobin Wu, Yan Liu, and Jieping Ye. D^2-city: a large-scale dashcam video dataset of diverse traffic scenarios. *arXiv preprint arXiv:1904.01975*, 2019.

[5] Joya Chen, Zhaoyang Lv, Shiwei Wu, Kevin Qinghong Lin, Chenan Song, Difei Gao, Jia-Wei Liu, Ziteng Gao, Dongxing Mao, and Mike Zheng Shou. Videollm-online: Online video large language model for streaming video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18407–18418, 2024.

[6] Joya Chen, Ziyun Zeng, Yiqi Lin, Wei Li, Zejun Ma, and Mike Zheng Shou. Livecc: Learning video llm with streaming speech transcription at scale. *arXiv preprint arXiv:2504.16030*, 2025.

[7] Yukang Chen, Fuzhao Xue, Dacheng Li, Qinghao Hu, Ligeng Zhu, Xiuyu Li, Yunhao Fang, Haotian Tang, Shang Yang, Zhijian Liu, et al. Longvila: Scaling long-context visual language models for long videos. *arXiv preprint arXiv:2408.10188*, 2024.

[8] Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, et al. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*, 2024.

[9] Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, et al. How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites. *Science China Information Sciences*, 67(12):220101, 2024.

[10] Zesen Cheng, Sicong Leng, Hang Zhang, Yifei Xin, Xin Li, Guanzheng Chen, Yongxin Zhu, Wenqi Zhang, Ziyang Luo, Deli Zhao, et al. Videollama 2: Advancing spatial-temporal modeling and audio understanding in video-llms. *arXiv preprint arXiv:2406.07476*, 2024.

[11] Shangzhe Di, Zhelun Yu, Guanghao Zhang, Haoyuan Li, Tao Zhong, Hao Cheng, Bolin Li, Wanggui He, Fangxun Shu, and Hao Jiang. Streaming video question-answering with in-context video kv-cache retrieval. *arXiv preprint arXiv:2503.00540*, 2025.

[12] Xin Ding, Hao Wu, Yifan Yang, Shiqi Jiang, Donglin Bai, Zhibo Chen, and Ting Cao. Streammind: Unlocking full frame rate streaming video dialogue through event-gated cognition. *arXiv preprint arXiv:2503.06220*, 2025.

[13] Heng Fan, Liting Lin, Fan Yang, Peng Chu, Ge Deng, Sijia Yu, Hexin Bai, Yong Xu, Chunyuan Liao, and Haibin Ling. Lasot: A high-quality benchmark for large-scale single object tracking. In *CVPR*, pages 5374–5383, 2019.

[14] Jianwu Fang, Lei-lei Li, Junfei Zhou, Junbin Xiao, Hongkai Yu, Chen Lv, Jianru Xue, and Tat-Seng Chua. Abductive ego-view accident video understanding for safe driving perception. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22030–22040, 2024.

[15] Jiajun Fei, Dian Li, Zhidong Deng, Zekun Wang, Gang Liu, and Hui Wang. Video-ccam: Enhancing video-language understanding with causal cross-attention masks for short and long videos. *arXiv preprint arXiv:2408.14023*, 2024.

[16] Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. *arXiv preprint arXiv:2405.21075*, 2024.

[17] Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. Tall: Temporal activity localization via language query. In *ICCV*, pages 5267–5275, 2017.

[18] Chunhui Gu, Chen Sun, Sudheendra Vijayanarasimhan, Caroline Pantofaru, David A. Ross, George Toderici, Yeqing Li, Susanna Ricco, Rahul Sukthankar, Cordelia Schmid, and Jitendra Malik. Ava: A video dataset of spatio-temporally localized atomic visual actions. *CVPR*, 2017.

[19] Bin Huang, Xin Wang, Hong Chen, Zihan Song, and Wenwu Zhu. Vtimellm: Empower llm to grasp video moments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14271–14280, 2024.

[20] De-An Huang, Shijia Liao, Subhashree Radhakrishnan, Hongxu Yin, Pavlo Molchanov, Zhiding Yu, and Jan Kautz. Lita: Language instructed temporal-localization assistant. In *European Conference on Computer Vision*, pages 202–218. Springer, 2024.

[21] Gabriel Huang, Bo Pang, Zhenhai Zhu, Clara Rivera, and Radu Soricut. Multimodal pretraining for dense video captioning. In Kam-Fai Wong, Kevin Knight, and Hua Wu, editors, *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, December 2020.

[22] Lianghua Huang, Xin Zhao, and Kaiqi Huang. Got-10k: A large high-diversity benchmark for generic object tracking in the wild. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(5): 1562–1577, 2021. doi: 10.1109/TPAMI.2019.2957464.

[23] Zhenpeng Huang, Xinhao Li, Jiaqi Li, Jing Wang, Xiangyu Zeng, Cheng Liang, Tao Wu, Xi Chen, Liang Li, and Limin Wang. Online video understanding: Ovbench and videochat-online. In *CVPR*, pages 3328–3338, 2025.

[24] Salman Khan, Izzeddin Teeti, Reza Javanmard Alitappeh, Mihaela C Stoian, Eleonora Giunchiglia, Gurkirt Singh, Andrew Bradley, and Fabio Cuzzolin. Road-waymo: Action awareness at scale for autonomous driving. *arXiv preprint arXiv:2411.01683*, 2024.

[25] Rahima Khanam and Muhammad Hussain. Yolov11: An overview of the key architectural enhancements. *arXiv preprint arXiv:2410.17725*, 2024.

[26] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *ICCV*, 2017.

[27] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *IJCV*, 123:32–73, 2017.

[28] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024.

[29] Hongyu Li, Jinyu Chen, Ziyu Wei, Shaofei Huang, Tianrui Hui, Jialin Gao, Xiaoming Wei, and Si Liu. Llava-st: A multimodal large language model for fine-grained spatial-temporal understanding. *arXiv preprint arXiv:2501.08282*, 2025.

[30] KunChang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. Videochat: Chat-centric video understanding. *arXiv preprint arXiv:2305.06355*, 2023.

[31] Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, et al. Mvbench: A comprehensive multi-modal video understanding benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22195–22206, 2024.

[32] Wei Li, Bing Hu, Rui Shao, Leyang Shen, and Liqiang Nie. Lion-fs: Fast & slow video-language thinker as online video assistant. *arXiv preprint arXiv:2503.03663*, 2025.

[33] Xinhao Li, Yi Wang, Jiashuo Yu, Xiangyu Zeng, Yuhan Zhu, Haian Huang, Jianfei Gao, Kunchang Li, Yinan He, Chenting Wang, et al. Videochat-flash: Hierarchical compression for long-context video modeling. *arXiv preprint arXiv:2501.00574*, 2024.

[34] Xinhao Li, Ziang Yan, Desen Meng, Lu Dong, Xiangyu Zeng, Yinan He, Yali Wang, Yu Qiao, Yi Wang, and Limin Wang. Videochat-r1: Enhancing spatio-temporal perception via reinforcement fine-tuning. *arXiv preprint arXiv:2504.06958*, 2025.

[35] Yanwei Li, Chengyao Wang, and Jiaya Jia. Llama-vid: An image is worth 2 tokens in large language models. In *European Conference on Computer Vision*, pages 323–340. Springer, 2024.

[36] Yifei Li, Junbo Niu, Ziyang Miao, Chunjiang Ge, Yuanhang Zhou, Qihao He, Xiaoyi Dong, Haodong Duan, Shuangrui Ding, Rui Qian, et al. Ovo-bench: How far is your video-llms from real-world online video understanding? *arXiv preprint arXiv:2501.05510*, 2025.

[37] Bin Lin, Yang Ye, Bin Zhu, Jiaxi Cui, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection. *arXiv preprint arXiv:2311.10122*, 2023.

[38] Ji Lin, Hongxu Yin, Wei Ping, Pavlo Molchanov, Mohammad Shoeybi, and Song Han. Vila: On pre-training for visual language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 26689–26699, 2024.

[39] Junming Lin, Zheng Fang, Chi Chen, Zihao Wan, Fuwen Luo, Peng Li, Yang Liu, and Maosong Sun. Streamingbench: Assessing the gap for mllms to achieve streaming video understanding. *arXiv preprint arXiv:2411.03628*, 2024.

[40] Jiajun Liu, Yibing Wang, Hanghang Ma, Xiaoping Wu, Xiaoqi Ma, Xiaoming Wei, Jianbin Jiao, Enhua Wu, and Jie Hu. Kangaroo: A powerful video-language model supporting long-context video input. *arXiv preprint arXiv:2408.15542*, 2024.

[41] Zhijian Liu, Ligeng Zhu, Baifeng Shi, Zhuoyang Zhang, Yuming Lou, Shang Yang, Haocheng Xi, Shiyi Cao, Yuxian Gu, Dacheng Li, et al. Nvila: Efficient frontier visual language models. *arXiv preprint arXiv:2412.04468*, 2024.

[42] Zuyan Liu, Yuhao Dong, Ziwei Liu, Winston Hu, Jiwen Lu, and Yongming Rao. Oryx mllm: On-demand spatial-temporal understanding at arbitrary resolution. *arXiv preprint arXiv:2409.12961*, 2024.

[43] Andreea-Maria Oncescu, Joao F Henriques, Yang Liu, Andrew Zisserman, and Samuel Albanie. Queryd: A video dataset with high-quality text and audio narrations. In *ICASSP*, pages 2265–2269. IEEE, 2021.

[44] OpenAI. Hello gpt-4o. https://openai.com/index/hello-gpt-4o, 2024.

[45] Viorica Pătrăucean, Lucas Smaira, Ankush Gupta, Adrià Recasens Continente, Larisa Markeeva, Dylan Banarse, Skanda Koppula, Joseph Heyward, Mateusz Malinowski, Yi Yang, Carl Doersch, Tatiana Matejovicova, Yury Sulsky, Antoine Miech, Alex Frechette, Hanna Klimczak, Raphael Koster, Junlin Zhang, Stephanie Winkler, Yusuf Aytar, Simon Osindero, Dima Damen, Andrew Zisserman, and João Carreira. Perception test: A diagnostic benchmark for multimodal video models. In *Advances in Neural Information Processing Systems*, 2023. URL `https://openreview.net/forum?id=HYEGXFnPoq`.

[46] Rui Qian, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Shuangrui Ding, Dahua Lin, and Jiaqi Wang. Streaming long video understanding with large language models. *Advances in Neural Information Processing Systems*, 37:119336–119360, 2024.

[47] Rui Qian, Shuangrui Ding, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Yuhang Cao, Dahua Lin, and Jiaqi Wang. Dispider: Enabling video llms with active real-time interaction via disentangled perception, decision, and reaction. *arXiv preprint arXiv:2501.03218*, 2025.

[48] Shuhuai Ren, Linli Yao, Shicheng Li, Xu Sun, and Lu Hou. Timechat: A time-sensitive multimodal large language model for long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14313–14323, 2024.

[49] Hao Shao, Yuxuan Hu, Letian Wang, Guanglu Song, Steven L Waslander, Yu Liu, and Hongsheng Li. Lmdrive: Closed-loop end-to-end driving with large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15120–15130, 2024.

[50] Xiaoqian Shen, Yunyang Xiong, Changsheng Zhao, Lemeng Wu, Jun Chen, Chenchen Zhu, Zechun Liu, Fanyi Xiao, Balakrishnan Varadarajan, Florian Bordes, et al. Longvu: Spatiotemporal adaptive compression for long video-language understanding. *arXiv preprint arXiv:2410.17434*, 2024.

[51] Yan Shu, Zheng Liu, Peitian Zhang, Minghao Qin, Junjie Zhou, Zhengyang Liang, Tiejun Huang, and Bo Zhao. Video-xl: Extra-long vision language model for hour-scale video understanding. *arXiv preprint arXiv:2409.14485*, 2024.

[52] Enxin Song, Wenhao Chai, Guanhong Wang, Yucheng Zhang, Haoyang Zhou, Feiyang Wu, Haozhe Chi, Xun Guo, Tian Ye, Yanting Zhang, et al. Moviechat: From dense token to sparse memory for long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18221–18232, 2024.

[53] Yansong Tang, Dajun Ding, Yongming Rao, Yu Zheng, Danyang Zhang, Lili Zhao, Jiwen Lu, and Jie Zhou. COIN: A large-scale dataset for comprehensive instructional video analysis. In *CVPR*, pages 1207–1216, 2019.

[54] Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

[55] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

[56] Weiyun Wang, Yiming Ren, Haowen Luo, Tiantong Li, Chenxiang Yan, Zhe Chen, Wenhai Wang, Qingyun Li, Lewei Lu, Xizhou Zhu, et al. The all-seeing project v2: Towards general relation comprehension of the open world. *arXiv preprint arXiv:2402.19474*, 2024.

[57] Xidong Wang, Dingjie Song, Shunian Chen, Chen Zhang, and Benyou Wang. Longllava: Scaling multi-modal llms to 1000 images efficiently via a hybrid architecture. *arXiv preprint arXiv:2409.02889*, 2024.

[58] Yi Wang, Kunchang Li, Xinhao Li, Jiashuo Yu, Yinan He, Guo Chen, Baoqi Pei, Rongkun Zheng, Zun Wang, Yansong Shi, Tianxiang Jiang, Songze Li, Jilan Xu, Hongjie Zhang, Yifei Huang, Yu Qiao, Yali Wang, and Limin Wang. Internvideo2: Scaling foundation models for multimodal video understanding. In *ECCV*, pages 396–416, 2024.

[59] Yi Wang, Xinhao Li, Ziang Yan, Yinan He, Jiashuo Yu, Xiangyu Zeng, Chenting Wang, Changlian Ma, Haian Huang, Jianfei Gao, et al. Internvideo2. 5: Empowering video mllms with long and rich context modeling. *arXiv preprint arXiv:2501.12386*, 2025.

[60] Yueqian Wang, Xiaojun Meng, Yuxuan Wang, Jianxin Liang, Jiansheng Wei, Huishuai Zhang, and Dongyan Zhao. Videollm knows when to speak: Enhancing time-sensitive video comprehension with video-text duet interaction format, 2024. URL `https://arxiv.org/abs/2411.17991`.

[61] Haomiao Xiong, Zongxin Yang, Jiazuo Yu, Yunzhi Zhuge, Lu Zhang, Jiawen Zhu, and Huchuan Lu. Streaming video understanding and multi-round interaction with memory-enhanced knowledge. *arXiv preprint arXiv:2501.13468*, 2025.

[62] Ziang Yan, Zhilin Li, Yinan He, Chenting Wang, Kunchang Li, Xinhao Li, Xiangyu Zeng, Zilei Wang, Yali Wang, Yu Qiao, et al. Task preference optimization: Improving multimodal large language models with vision task alignment. *arXiv preprint arXiv:2412.19326*, 2024.

[63] Zhenyu Yang, Yuhang Hu, Zemin Du, Dizhan Xue, Shengsheng Qian, Jiahong Wu, Fan Yang, Weiming Dong, and Changsheng Xu. Svbench: A benchmark with temporal multi-turn dialogues for streaming video understanding. *arXiv preprint arXiv:2502.10810*, 2025.

[64] Linli Yao, Yicheng Li, Yuancheng Wei, Lei Li, Shuhuai Ren, Yuanxin Liu, Kun Ouyang, Lean Wang, Shicheng Li, Sida Li, et al. Timechat-online: 80% visual tokens are naturally redundant in streaming videos. *arXiv preprint arXiv:2504.17343*, 2025.

[65] Yuan Yao, Tianyu Yu, Ao Zhang, Chongyi Wang, Junbo Cui, Hongji Zhu, Tianchi Cai, Haoyu Li, Weilin Zhao, Zhihui He, et al. Minicpm-v: A gpt-4v level mllm on your phone. *arXiv preprint arXiv:2408.01800*, 2024.

[66] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2636–2645, 2020.

[67] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. Modeling context in referring expressions. In *ECCV*, pages 69–85. Springer, 2016.

[68] Abhay Zala, Jaemin Cho, Satwik Kottur, Xilun Chen, Barlas Oguz, Yashar Mehdad, and Mohit Bansal. Hierarchical video-moment retrieval and step-captioning. In *CVPR*, pages 23056–23065, 2023.

[69] Xiangyu Zeng, Kunchang Li, Chenting Wang, Xinhao Li, Tianxiang Jiang, Ziang Yan, Songze Li, Yansong Shi, Zhengrong Yue, Yi Wang, et al. Timesuite: Improving mllms for long video understanding via grounded tuning. *arXiv preprint arXiv:2410.19702*, 2024.

[70] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11975–11986, 2023.

[71] Boqiang Zhang, Kehan Li, Zesen Cheng, Zhiqiang Hu, Yuqian Yuan, Guanzheng Chen, Sicong Leng, Yuming Jiang, Hang Zhang, Xin Li, et al. Videollama 3: Frontier multimodal foundation models for image and video understanding. *arXiv preprint arXiv:2501.13106*, 2025.

[72] Haoji Zhang, Yiqin Wang, Yansong Tang, Yong Liu, Jiashi Feng, Jifeng Dai, and Xiaojie Jin. Flash-vstream: Memory-based real-time understanding for long video streams. *arXiv preprint arXiv:2406.08085*, 2024.

[73] Jun-Bo Zhang, Wei Feng, Meng-Biao Zhao, Fei Yin, Xu-Yao Zhang, and Cheng-Lin Liu. Video text detection with robust feature representation. *IEEE Transactions on Circuits and Systems for Video Technology*, 34(6):4407–4420, 2023.

[74] Pan Zhang, Xiaoyi Dong, Yuhang Cao, Yuhang Zang, Rui Qian, Xilin Wei, Lin Chen, Yifei Li, Junbo Niu, Shuangrui Ding, et al. Internlm-xcomposer2. 5-omnilive: A comprehensive multimodal system for long-term streaming video and audio interactions. *arXiv preprint arXiv:2412.09596*, 2024.

[75] Peiyuan Zhang, Kaichen Zhang, Bo Li, Guangtao Zeng, Jingkang Yang, Yuanhan Zhang, Ziyue Wang, Haoran Tan, Chunyuan Li, and Ziwei Liu. Long context transfer from language to vision. *arXiv preprint arXiv:2406.16852*, 2024.

[76] Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. Video instruction tuning with synthetic data. *arXiv preprint arXiv:2410.02713*, 2024.

[77] Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Shitao Xiao, Xi Yang, Yongping Xiong, Bo Zhang, Tiejun Huang, and Zheng Liu. Mlvu: A comprehensive benchmark for multi-task long video understanding. *arXiv preprint arXiv:2406.04264*, 2024.

[78] Luowei Zhou, Chenliang Xu, and Jason J. Corso. Towards automatic learning of procedures from web instructional videos. In *AAAI Conference on Artificial Intelligence*, 2017. URL `https://api.semanticscholar.org/CorpusID:19713015`.

[79] Haoyi Zhu, Honghui Yang, Yating Wang, Jiange Yang, Limin Wang, and Tong He. Spa: 3d spatial-awareness enables effective embodied representation. *arXiv preprint arXiv:2410.08208*, 2024.

[80] Zhe Zhu, Dun Liang, Songhai Zhang, Xiaolei Huang, Baoli Li, and Shimin Hu. Traffic-sign detection and classification in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2110–2118, 2016.

[81] Orr Zohar, Xiaohan Wang, Yann Dubois, Nikhil Mehta, Tong Xiao, Philippe Hansen-Estruch, Licheng Yu, Xiaofang Wang, Felix Juefei-Xu, Ning Zhang, et al. Apollo: An exploration of video understanding in large multimodal models. *arXiv preprint arXiv:2412.10360*, 2024.

[82] Jannik Zürn, Paul Gladkov, Sofía Dudas, Fergal Cotter, Sofi Toteva, Jamie Shotton, Vasiliki Simaiaki, and Nikhil Mohan. Wayvescenes101: A dataset and benchmark for novel view synthesis in autonomous driving. *arXiv preprint arXiv:2407.08280*, 2024.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Our main claims made in the abstract and introduction accurately reflect the paper's contributions and scope.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations of our approach in the Appendix.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: Our paper does not include theoretical proof.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We describe the implementation details for our experiments in our paper and all results can be reproduced.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Our models, data and code have been released.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We list the core experimental details in the main text, and more detailed experimental details are included in the Appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Our experiments did not conduet statistical significance.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide the computing resources needed for the experiments in the Experiments section.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our paper follows the NeurIPS Code of Ethics in every respect.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss the potential societal impacts in the Appendix.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Our paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We follow the license of each asset we have used and cite them properly.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: Our paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Our paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: Our paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: We deseribe the usage of LLMs in this research.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.

A Details of node's timestamp

Each event node's timestamp is initialized as the average time of the frames it represents (e.g., if an event node spans frames from 10s to 14s, its timestamp is initialized as 12s). When two event nodes are merged, the timestamp of the new node is computed as a token-count-weighted average of the original nodes:

$$t_{\text{new}} = \frac{t_i \cdot n_i + t_j \cdot n_j}{n_i + n_j}, \tag{5}$$

where t_i, t_j are the timestamps of the original event nodes, and n_i, n_j are the numbers of visual tokens contained in each node, respectively. This weighted scheme prevents timestamp drift during multiple rounds of merging, especially when the merged nodes contain significantly different amounts of visual tokens.

B Details of OnlineIT

In this section, we provide a comprehensive description of the task categorization and data distribution of the OnlineIT dataset. It is specifically designed to enhance the streaming video understanding capabilities of MLLMs in terms of real-time perception, future prediction, and event continuity. As shown in Table 8, the dataset is divided into two major components: OnlineIT-general, which targets general streaming video understanding, and OnlineIT-drive, which focuses on autonomous driving scenarios. Each subset is carefully designed to cover a diverse range of fine-grained perception and reasoning tasks with high-quality annotations.

Table 8: Task types and data volumes of OnlineIT.

Dataset	Categories	Task	Sourse	Instance Num
OnlineIT-general	Spatial Perception	Spatial Grounding	RefCOCO [67]	∼43k
			Allseeing-V2 [56]	∼45k
		Multi-round Spatial Understanding	Visual Genome [27]	∼43k
		Spatial Grounded VQA	Allseeing-V2 [56]	∼43k
		Relative Spatial Localization	LaSOT [13]	∼19k
	Temporal Perception	Temporal Grounding	Charades-STA [17]	∼11k
			HiREST [68]	∼0.4k
			QuerYD [43]	∼13k
		Reasoning Temporal Localization	ActivityNet-RTL [20]	∼10k
		Multi-format Temporal Grounding	InternVid-VTime [19]	∼20k
	Spatiotemporal Perception	Spatiotemporal Action Localization	AVA [18]	∼6k
		Object Backward Tracking	LaSOT [13]	∼51k
			GOT [22]	∼58k
		Spatiotemporal Detection	LaSOT [13]	∼14k
	Event Perception	Dense Video Captioning	ActivityNet-Captions [26]	∼10k
			ViTT [78]	∼5k
			Youcook2 [21]	∼1k
		Step Localization and Captioning	COIN [53]	∼9k
			HiREST [68]	∼0.5k
OnlineIT-drive	Static Target	Past Memory	D^2-City [4]	∼9k
		Real-time Perception	TT100k [80]	∼46k
			D^2-City [4]	∼13k
	Dynamic Target	Localization Prediction	Road-waymo [24]	∼1k
		Move Distance Prediction	Road-waymo [24]	∼7k
	Event Oriented	Accident Reasoning	MM-AU [14]	∼6k
		Risk Analysis	MM-AU [14]	∼7k

B.1 OnlineIT-general

OnlineIT-general encompasses a broad scope of tasks designed to foster a comprehensive understanding of spatiotemporal video content in streaming settings. As shown in Table 8, the dataset is categorized into four primary task types: spatial perception, temporal perception, spatiotemporal perception, and event perception. To ensure diversity, robustness, and fine-grained task coverage, we compiled and refined data from a wide array of sources. In total, OnlineIT-general comprises over 400k instances spanning various difficulty levels and video durations.

Spatial Perception. This task type includes four subtasks. ***Spatial Grounding*** requires the model to output the bounding box indicating the location of a queried object. ***Multi-round Spatial Understanding*** involves identifying the object's spatial location through multi-turn dialogue or generating a caption for the object within a specified spatial region. ***Spatial Grounded VQA*** combines visual question answering with spatial localization, requiring the model to provide the bounding box of the relevant area while answering the question. ***Relative Spatial Localization*** challenges the model to determine the position of a specified object relative to the overall scene. These tasks emphasize spatial grounding and reasoning, which are crucial for enhancing a model's fine-grained spatial perception in real-time streaming video scenarios.

Temporal Perception. This category consists of three subtasks. ***Temporal Grounding*** involves interpreting a natural language query and identifying the start and end timestamps of the corresponding video segment. In streaming scenarios, the model must also assess whether the described event is currently ongoing. ***Reasoning Temporal Localization*** requires identifying the relevant time span of an event while answering a reasoning-based question. ***Multi-format Temporal Localization*** incorporates both single-turn and multi-turn dialogues, covering a diverse range of question formats. These tasks focus on strengthening the MLLM's ability to track and reason about temporal dependencies, improving its understanding of both current and past moments in a video stream.

Spatiotemporal Perception. This task type integrates spatial and temporal reasoning and includes three subtasks. ***Spatiotemporal Action Localization*** requires the model to predict both the spatial location and the action being performed by a target at a specific query time. ***Object Backward Tracking*** tasks the model with identifying the current location of an object and tracing its position at previous time points, such as one or two seconds earlier. ***Spatiotemporal Detection*** operates over broader temporal windows, asking whether an object visible in the current frame existed several seconds ago or requiring the model to locate an object at a specified historical moment and determine its duration of existence. These tasks combine spatiotemporal cues to capture actions, motion, and transitions, allowing the model to track object trajectories and anticipate future states based on past and present context.

Event Perception. This category includes two subtasks. ***Dense Video Captioning*** involves detecting a sequence of events in a video and generating corresponding timestamps along with high-level descriptions. ***Step Localization and Captioning*** differs by focusing on segmenting and narrating key procedural steps within long-form videos. These tasks are aimed at improving the model's understanding of complex, multi-step events, enabling structured interpretation of dynamic sequences in streaming video understanding.

B.2 OnlineIT-drive

OnlineIT-drive is designed specifically for the domain of streaming video understanding in autonomous driving. The dataset includes 89k instances, which are organized into three major task categories. Collectively, these tasks aim to strengthen not only real-time perception capabilities but also the temporal reasoning and decision-making abilities of MLLMs in high-stakes and rapidly evolving environments.

Static Target Understanding. To improve the model's capacity for static scene understanding, two task types are introduced. ***Real-time Perception*** requires the model to accurately perceive and interpret the semantics and spatial attributes of traffic-related targets as they appear in real time. ***Past Memory*** assesses the model's ability to retain and retrieve the semantics and spatiotemporal characteristics of traffic targets that were observed at a prior point in time. These tasks collectively enhance the model's capability to perceive, understand, and remember static traffic elements and environmental context, such as road infrastructure and regulatory signage.

Dynamic Target Understanding. It includes two task types that aim to enhance predictive understanding of dynamic traffic participants. ***Location Prediction*** requires the model to estimate the future position of a moving target based on its historical motion trajectory. ***Move Distance Prediction*** focuses on predicting the distance traveled between the ego vehicle and other moving agents, given motion-related observations. These tasks are designed to improve the model's ability to track continuously moving objects and to anticipate future trajectories.

Event Oriented Reasoning. It is intended to foster the development of reasoning abilities necessary for risk assessment and accident interpretation. ***Risk Analysis*** requires the model to detect potential sources of danger in the current traffic scene and to assess the likelihood of accident occurrence. ***Accident Reasoning*** involves post hoc analysis, where the model must infer the causes of an observed accident and articulate plausible preventive strategies. These tasks are designed to enhance the model's ability to reason about causal relationships and to anticipate or reflect on traffic risks with contextual awareness.

C Details of ODV-Bench

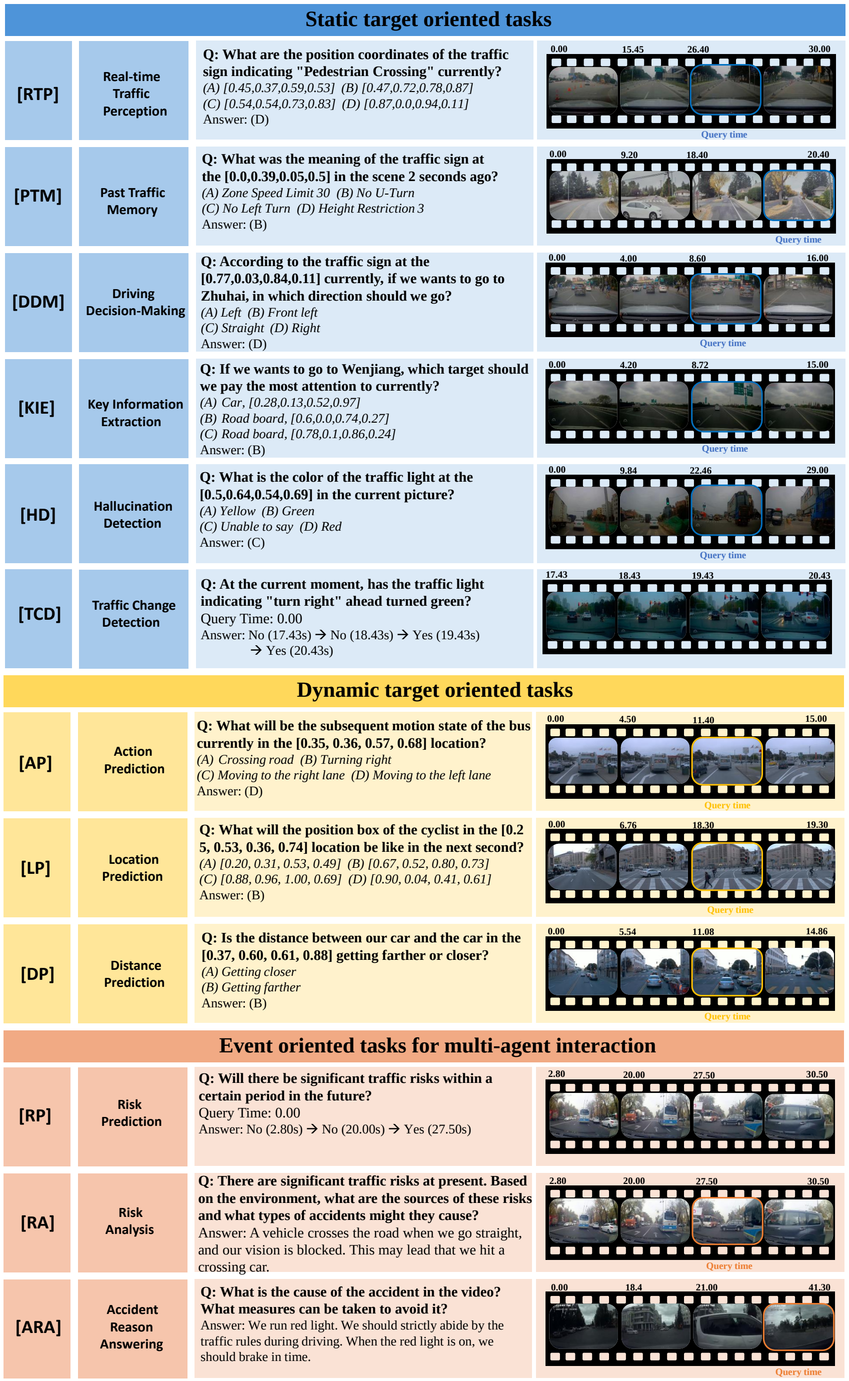
In this section, we detail the task taxonomy and formulation of the ODV-Bench, as well as the dataset statistics. We categorize task scenarios based on target entities and derive key perception and reasoning task types for each scenario in Table 9.

Table 9: Overview of task categories, their subcategories, and question templates.

Task Objective Scenario	Sub-task	Query Examples
Static Target	Real-time Traffic Perception	1) What is the meaning of the traffic sign at the [0.61,0.31,0.64,0.38] in the current picture? 2) What are the position coordinates of the traffic sign indicating ”Pedestrian Crossing” in the current picture? 3) What is the meaning of the road board at the [0.77,0.08,0.88,0.2] in the current picture? 4) According to the road board at the [0.09,0.46,0.19,0.52] currently, if going in the left direction, where will we go, how far is it? 5) What is the color of the traffic light at the [0.42,0.01,0.45,0.13] in the current picture? And what is its indication? 6) According to the road board at the [0.65,0.03,0.76,0.16] currently, how far is it from Fengle?
	Past Traffic Memory	1) What were the position coordinates of the traffic sign indicating ”No Left Turn” in the scene 3 seconds ago? 2) What was the meaning of the traffic sign at the [0.39,0.13,0.41,0.17] in the scene 1 seconds ago? 3) The traffic sign is currently located at [0.92,0.02,0.96,0.09]. What were its coordinates 2 seconds ago? 4) What was the color of the traffic light at the [0.35,0.1,0.38,0.22] in the scene 2 seconds ago?
	Driving Decision-Making	1) According to the road board at the [0.37,0.18,0.47,0.31] in the image taken 2 seconds ago, if we wants to go to Renhe, in which direction should we go and how far is it? 2) According to the road board at the [0.51,0.18,0.61,0.3] currently, if we wants to go to Libai Avenue, in which direction should we go and how far is it? 3) According to the road board at the [0.38,0.04,0.63,0.23] currently, if we wants to turn left, which lane should we be in?
	Key Information Extraction	1) If we wants to go to Suzhou, which target should we pay the most attention to currently? Provide the type and coordinates.
	Hallucination Detection	1) According to the road board at the [0.38,0.31,0.52,0.48], how far is it currently from Qingpu Town? 2) What is the meaning of the traffic sign at the [0.34,0.29,0.38,0.38] in the current picture? 3) What is the color of the traffic light at the [0.9,0.86,0.95,0.92] in the current picture?
	Traffic Change Detection	1) At the current moment, has the traffic signal light indicating ”turn right” ahead turned completely green? 2) At the current moment, has the traffic signal light ahead turned completely red?
Dynamic Target	Action Prediction	1) What will be the subsequent motion state of the car currently in the [0.993, 0.615, 1.0, 0.63] location?
	Location Prediction	1) What will the position box of the pedestrian in the [0.544, 0.561, 0.613, 0.895] location be like in the next second?
	Distance Prediction	1) Is the distance between our car and the car in the [0.488, 0.488, 0.501, 0.494] getting farther or closer?
Multi-agent Interaction Event	Risk Prediction	1) Is there a high probability of traffic accidents occurring within a certain period in the future? 2) Will there be significant traffic risks within a certain period in the future?
	Risk Analysis	1) There is a high risk of traffic accidents at present. Based on the environment, what types of accidents are likely to occur, and what is the basis for this prediction? 2) There are significant traffic risks at present. Based on the environment, what are the sources of these risks and what types of accidents might they cause?
	Accident Reason Answering	1) What is the cause of the accident in the video? What measures can be taken to avoid it?

C.1 Task Taxonomy and Formulation

We first identify the primary categories of traffic entities relevant to autonomous driving and organize task scenarios into three groups: **(1) Tasks for Static Targets**, which involve the recognition and retrieval of stationary traffic elements such as traffic signs, lights, and road indicators; **(2) Tasks for dynamic targets**, which focus on behavior prediction and localization of moving entities such as vehicles and pedestrians; and **(3) Tasks for multitarget interaction events**, which capture complex interactions, risk scenarios, and accidents involving multiple agents. Based on these categories and guided by temporal cues and the practical needs of driving, we further define fine-grained task types to comprehensively assess model understanding in realistic online driving video scenarios.



Static target oriented tasks

[RTP] Real-time Traffic Perception

Q: What are the position coordinates of the traffic sign indicating "Pedestrian Crossing" currently?
(A) [0.45,0.37,0.59,0.53] (B) [0.47,0.72,0.78,0.87]
(C) [0.54,0.54,0.73,0.83] (D) [0.87,0.0,0.94,0.11]
Answer: (D)

[PTM] Past Traffic Memory

Q: What was the meaning of the traffic sign at the [0.0,0.39,0.05,0.5] in the scene 2 seconds ago?
(A) Zone Speed Limit 30 (B) No U-Turn
(C) No Left Turn (D) Height Restriction 3
Answer: (B)

[DDM] Driving Decision-Making

Q: According to the traffic sign at the [0.77,0.03,0.84,0.11] currently, if we wants to go to Zhuhai, in which direction should we go?
(A) Left (B) Front left
(C) Straight (D) Right
Answer: (D)

[KIE] Key Information Extraction

Q: If we wants to go to Wenjiang, which target should we pay the most attention to currently?
(A) Car, [0.28,0.13,0.52,0.97]
(B) Road board, [0.6,0.0,0.74,0.27]
(C) Road board, [0.78,0.1,0.86,0.24]
Answer: (B)

[HD] Hallucination Detection

Q: What is the color of the traffic light at the [0.5,0.64,0.54,0.69] in the current picture?
(A) Yellow (B) Green
(C) Unable to say (D) Red
Answer: (C)

[TCD] Traffic Change Detection

Q: At the current moment, has the traffic light indicating "turn right" ahead turned green?
Query Time: 0.00
Answer: No (17.43s) → No (18.43s) → Yes (19.43s) → Yes (20.43s)

Dynamic target oriented tasks

[AP] Action Prediction

Q: What will be the subsequent motion state of the bus currently in the [0.35, 0.36, 0.57, 0.68] location?
(A) Crossing road (B) Turning right
(C) Moving to the right lane (D) Moving to the left lane
Answer: (D)

[LP] Location Prediction

Q: What will the position box of the cyclist in the [0.2 5, 0.53, 0.36, 0.74] location be like in the next second?
(A) [0.20, 0.31, 0.53, 0.49] (B) [0.67, 0.52, 0.80, 0.73]
(C) [0.88, 0.96, 1.00, 0.69] (D) [0.90, 0.04, 0.41, 0.61]
Answer: (B)

[DP] Distance Prediction

Q: Is the distance between our car and the car in the [0.37, 0.60, 0.61, 0.88] getting farther or closer?
(A) Getting closer
(B) Getting farther
Answer: (B)

Event oriented tasks for multi-agent interaction

[RP] Risk Prediction

Q: Will there be significant traffic risks within a certain period in the future?
Query Time: 0.00
Answer: No (2.80s) → No (20.00s) → Yes (27.50s)

[RA] Risk Analysis

Q: There are significant traffic risks at present. Based on the environment, what are the sources of these risks and what types of accidents might they cause?
Answer: A vehicle crosses the road when we go straight, and our vision is blocked. This may lead that we hit a crossing car.

[ARA] Accident Reason Answering

Q: What is the cause of the accident in the video? What measures can be taken to avoid it?
Answer: We run red light. We should strictly abide by the traffic rules during driving. When the red light is on, we should brake in time.

Figure 6: Examples of each task in ODV-Bench. The 12 tasks are divided into three different perception modes for online video understanding for autonomous driving.

C.1.1 Tasks for Static Targets

Static traffic elements, such as traffic signs and road indicators, play a crucial role in driving decisions and hazard avoidance under normal driving conditions. To evaluate the model's ability to retrieve and recognize these elements in online video streams, we design a dedicated set of tasks. Specifically, we distinguish between basic perception tasks and more advanced reasoning tasks, and further refine them based on temporal cues and practical driving needs: **(1) Real-time Traffic Perception:** Perceive and interpret the semantics and spatial locations of static traffic elements in real time; **(2) Past Traffic Memory:** Recall and track the semantics and spatiotemporal states of previously observed static elements; **(3) Driving Decision-Making:** make driving decisions based on the perceived information; **(4) Key Information Extraction:** Identify and locate key traffic elements critical to driving decisions; **(5) Hallucination Detection:** identify questions irrelevant to the existing video input; and **(6) Traffic Change Detection:** detect timestamps for changes in traffic elements, such as traffic lights.

C.1.2 Tasks for Dynamic Targets

The position and behavior of other road participants, such as vehicles and pedestrians, are crucial reference factors influencing autonomous driving decisions and safety. The ability to predict the position and behavior of dynamic traffic objects is essential to ensure the safety of autonomous driving. Therefore, we focus on the following three tasks to effectively evaluate this capability: **(1) Action Prediction:** predicting the next action of vehicles and pedestrians based on continuous spatiotemporal cues; **(2) Distance Prediction:** predicting the relative distance change between the ego-vehicle and other vehicles based on motion information; and **(3) Location Prediction:** predicting the future spatial position of dynamic traffic targets based on their movement trajectories.

C.1.3 Tasks for Multi-Target Interaction Events

To achieve safe and reliable autonomous driving, the system must be able to identify risks and analyze accidents in complex road interaction scenarios. In the context of online video streams, this ability involves the dynamic recognition and analysis of multi-agent interactions, as well as the reasonable prediction of traffic risks. To evaluate this capability, we design the following three task categories: **(1) Risk Prediction:** predicting the occurrence of significant traffic risks and responding proactively; **(2) Risk Analysis:** detecting the sources of current traffic risks and analyzing the potential causes of accidents; and **(3) Accident Reason Answering:** post-accident analysis, providing potential causes for the incident and summarizing actionable lessons learned.

C.2 Dataset Statistics

ODV-Bench comprises 1,190 unique first-person driving video clips, encompassing a diverse range of driving scenarios across different countries from routine driving conditions to potential hazards and accidents. The length of videos ranges from 5 seconds to 90 seconds, effectively capturing the diversity of real-world streaming driving experiences. The benchmark includes 6,348 question-answer pairs, with an average query timestamp of 18.9 seconds. Specifically, the static-object-oriented category comprises 247 videos with a total of 1,639 questions; the dynamic-object-oriented category includes 162 videos and 2,999 questions; and the event-oriented category consists of 781 videos with 1,710 questions. All questions are in multiple-choice format, with the number of options varying between 2 and 4 depending on the question type.

D More Implementation Details

Table 10: Parameter settings for three-stage offline pre-training.

		Stage-1	**Stage-2**	**Stage-3**
Vision	**Resolution×Num. frames**	384	384×8	Max 384×512
	#Tokens	64×4	64×8	Max 16×512
Data	**Dataset**	Image & Short Video	Image & Short Video	Image & Short / Long Video
	#Samples	0.6M & 0.5M	3.8M & 3.4M	0.5M & 2.8M
Model	**Trainable**	Projector	Full Model	Full Model
	#parameters	16.98MB	8030.35MB	8030.35MB
Training	**Batch Size**	512	256	256
	LR of *vision encoder*	$1{\times}10^{-3}$	2×10^{-6}	2×10^{-6}
	LR *of connector & LLM*	$1{\times}10^{-3}$	1×10^{-5}	1×10^{-5}
	Epoch	1	1	1

Table 11: Parameter settings for the fourth stage online fine-tuning and fifth dirve fine-tuning.

		Stage 4	**Stage5**
Data	**Dataset**	Image & (Short/Long/Online)-Video	Image & (Short/Long/Online)-Video
	#Samples	0.4M & 1.3M	0.2M & 0.5M
Model	**Trainable**	Projector & LLM	Projector & LLM
	#parameters	7632.60MB	7632.60MB
Vision	**Resolution**	384×384	384×384
	Frames	2∼512	2∼512
	FPS	1	1
Memory	**Real-time Perception Qouta**	729	729
	Spatiotemporal Memory Quota	128 × 18	128 × 18
	Total Visual Token Limits	8192	8192
	Similarity Penalty Weight	0.4	0.4
	Merge Count Penalty Weight	0.4	0.4
	Temporal Distance Penalty Weight	0.2	0.2
Training	**Batch Size**	256	256
	LR	1×10^{-5}	1×10^{-5}
	Epoch	1	1
	Optimizer	AdamW	AdamW
	Weight Decay	0	0
	Warmup Ratio	0.03	0.03
	LR Schedule	cosine	cosine
	Vision Select Layer	-2	-2
	GPU Nums	32	32

We adopt a five-stage training strategy to systematically train the proposed StreamForest model, aiming to fully exploit its potential for streaming video understanding tasks. In the first three stages, we follow and extend the training paradigm of VideoChat-Flash [33], employing offline training to endow the model with strong capabilities in long-form video comprehension and cross-modal alignment. These stages are designed progressively, covering diverse data scales and task objectives, enabling the model to gradually acquire core competencies such as basic vision-language alignment, long-term temporal modeling, and complex scene reasoning. Detailed training procedures and hyperparameter configurations for these stages are provided in Table 10.

In the fourth and fifth stages, we perform online fine-tuning to enhance the model's ability to process streaming inputs in realistic scenarios. By continuously feeding frame sequences during training, the model learns to retain a fine-grained perception of the current moment while maintaining long-term memory of past events, even under high compression constraints. The full configuration and parameter settings for the online fine-tuning phase are listed in Table 11. These stages are critical for transitioning the model from offline understanding to real-time reasoning, significantly improving its robustness and practical effectiveness in real-world applications.

E Full Performances

In the following parts, we present the full results and compare StreamForest with leading proprietary and open-source models. To comprehensively evaluate the effectiveness of StreamForest, we conduct experiments on three online video understanding benchmarks: StreamingBench, OVBench, and OVO-Bench.

E.1 StreamingBench

Table 12 presents the full evaluation results on StreamingBench, covering 12 real-time video understanding tasks. StreamForest achieves the highest average score (77.26%) among all evaluated models, both open-source and proprietary, while operating efficiently at 1 fps. Notably, StreamForest outperforms leading proprietary MLLMs such as GPT-4o (73.28%) and Gemini 1.5 Pro (75.69%). It also significantly surpasses top open-source offline models such as LLaVA-OneVision (71.12%) and Qwen2.5-VL (73.68%), underscoring its robust multimodal representation and reasoning capabilities. In the online video MLLM category, StreamForest sets a new state-of-the-art, outperforming

Table 12: Full evaluation results of real-time understanding tasks on StreamingBench.

Method	Size	#Frames	OP	CR	CS	ATP	EU	TR	PR	SU	ACP	CT	ALL
Human	-	-	89.47	92.00	93.60	91.47	95.65	92.52	88.00	88.75	89.74	91.30	91.46
Proprietary MLLMs													
Gemini 1.5 pro [54]	-	1fps	79.02	80.47	83.54	79.67	80.00	84.74	77.78	64.23	71.95	48.70	75.69
GPT-4o [44]	-	64	77.11	80.47	83.91	76.47	70.19	83.80	66.67	62.19	69.12	49.22	73.28
Claude 3.5 Sonnet [1]	-	20	73.33	80.47	84.09	82.02	75.39	79.53	61.11	61.79	69.32	43.09	72.44
Open-source Offline Video MLLMs													
Video-LLaMA2 [10]	7B	32	55.86	55.47	57.41	58.17	52.80	43.61	39.81	42.68	45.61	35.23	49.52
VILA-1.5 [38]	8B	14	53.68	49.22	70.98	56.86	53.42	53.89	54.63	48.78	50.14	17.62	52.32
Video-CCAM [15]	14B	96	56.40	57.81	65.30	62.75	64.60	51.40	42.59	47.97	49.58	31.61	53.96
LongVA [75]	7B	128	70.03	63.28	61.20	70.92	62.73	59.50	61.11	53.66	54.67	34.72	59.96
InternVL2 [9]	8B	16	68.12	60.94	69.40	77.12	67.70	62.93	59.26	53.25	54.96	56.48	63.72
Kangaroo [40]	7B	64	71.12	84.38	70.66	73.20	67.08	61.68	56.48	55.69	62.04	38.86	64.60
LLaVA-NeXT-Video [76]	32B	64	78.20	70.31	73.82	76.80	63.35	69.78	57.41	56.10	64.31	38.86	66.96
MiniCPM-V2.6 [65]	8B	32	71.93	71.09	77.92	75.82	64.60	65.73	70.37	56.10	62.32	53.37	67.44
LLaVA-OneVision [28]	7B	32	80.38	74.22	76.03	80.72	72.67	71.65	67.59	65.45	65.72	45.08	71.12
Qwen2.5-VL [2]	7B	1fps	78.32	80.47	78.86	80.45	76.73	78.50	79.63	63.41	66.19	53.19	73.68
Open-source Online Video MLLMs													
Flash-VStream [72]	7B	-	25.89	43.57	24.91	23.87	27.33	13.08	18.52	25.20	23.87	48.70	23.23
VideoLLM-online [5]	8B	2fps	39.07	40.06	34.49	31.05	45.96	32.40	31.48	34.16	42.49	27.89	35.99
Dispider [47]	7B	1fps	74.92	75.53	74.10	73.08	74.44	59.92	76.14	62.91	62.16	45.80	67.63
StreamForest(Ours)	7B	1fps	83.11	82.81	82.65	84.26	77.50	78.19	76.85	69.11	75.64	54.40	77.26

open-source counterparts Dispider (67.63%) by a wide margin. Its consistent accuracy and real-time efficiency demonstrate a strong potential for practical deployment in streaming applications.

E.2 OVBench

Table 13: Full evaluation results on OVBench.

Task Name	Size	FP			THV			PM			SP		STP		TP			AVG
Subset Name		AA	GSP	MP	AP	SV	OP	AR	PR	TR	AL	OP	AT	OT	AS	SL	OES	
Proprietary MLLMs																		
Gemini-1.5-Flash [54]	-	71.4	53.6	21.9	56.5	60.8	40.6	36.7	47.9	62.5	32.3	37.5	87.0	50.0	83.3	22.3	46.9	50.7
Open-source Offline Video MLLMs																		
InternVL2 [9]	7B	52.6	60.2	27.6	57.5	52.0	58.5	38.8	67.1	58.3	38.1	31.3	87.4	37.0	75.4	31.4	5.9	48.7
InternVL2 [9]	4B	57.7	57.0	14.4	59.2	49.4	60.0	30.3	61.8	46.3	30.9	20.1	83.0	32.3	70.7	29.4	3.4	44.1
LLaMA-VID [35]	7B	43.6	50.9	19.6	64.0	47.5	46.8	29.4	48.9	51.2	31.9	11.2	75.7	24.8	59.1	26.0	40.0	41.9
LLaVA-Onevision [28]	7B	68.0	62.7	35.9	58.4	50.3	46.5	29.4	60.7	58.0	43.1	14.2	86.5	49.7	70.7	28.1	30.2	49.5
LongVA [75]	7B	64.1	56.5	29.5	54.9	51.9	34.8	35.3	55.6	57.7	31.6	3.4	67.4	44.7	80.0	26.7	4.0	43.6
MiniCPM-V2.6 [65]	7B	33.3	35.9	15.0	59.2	50.8	55.1	25.0	37.4	41.7	26.6	11.8	98.3	36.3	66.1	26.4	6.2	39.1
Qwen2-VL [55]	7B	60.3	66.1	22.1	54.9	51.5	51.1	37.8	64.4	69.3	35.3	28.5	97.0	49.4	65.1	30.8	11.7	49.7
LITA [20]	7B	19.2	24.5	19.9	40.8	48.9	24.9	3.1	27.3	6.4	6.9	14.6	35.2	23.9	27.4	0.5	3.4	20.4
TimeChat [48]	7B	7.7	15.3	18.7	20.6	15.7	11.7	9.1	14.7	9.8	7.5	19.5	13.9	10.3	9.3	10.1	10.8	12.8
VTimeLLM [19]	7B	37.2	23.4	15.0	64.8	43.8	53.2	25.9	38.8	32.5	25.9	20.4	40.9	6.8	48.4	43.5	8.6	33.1
Open-source Online Video MLLMs																		
VideoLLM-Online [5]	7B	0	1.8	20.9	5.2	5.9	32.6	0	2.3	26.7	0.6	26.6	0.9	19.9	0.9	1.7	8.3	9.6
MovieChat [52]	7B	23.1	27.5	23.6	58.4	43.9	40.3	25.6	31.1	23.9	26.9	39.6	24.4	28.9	29.3	25.5	21.9	30.9
Flash-Vstream [72]	7B	26.9	37.6	23.9	60.1	41.9	40.0	23.4	35.3	26.1	24.7	28.8	27.0	21.4	29.8	25.6	26.8	31.2
Videochat-Online [23]	4B	64.1	59.7	16.6	63.1	58.3	62.8	42.2	54.4	70.6	54.1	24.8	88.7	48.5	73.0	25.9	71.7	54.9
StreamForest (Ours)	7B	69.2	60.0	34.4	69.1	54.0	72.9	50.9	64.9	82.2	56.6	87.9	95.2	61.2	64.2	30.6	92.6	60.5

Table 13 shows the comprehensive results on OVBench, encompassing six diverse task categories (FP, THV, PM, SP, STP, TP). StreamForest achieves the top average score of 60.5%, surpassing all open-source online and offline Video MLLMs. It significantly outperforms other open-source online models, e.g., Videochat-Online (54.9%) and Flash-VStream (31.2%), as well as offline models such as Qwen2-VL (49.7%) and LLaVA-OneVision (49.5%). Compared to Gemini-1.5-Flash (50.7%), StreamForest delivers nearly 10 points higher accuracy on average, affirming its capability to balance real-time efficiency with high performance.

E.3 OVO-Bench

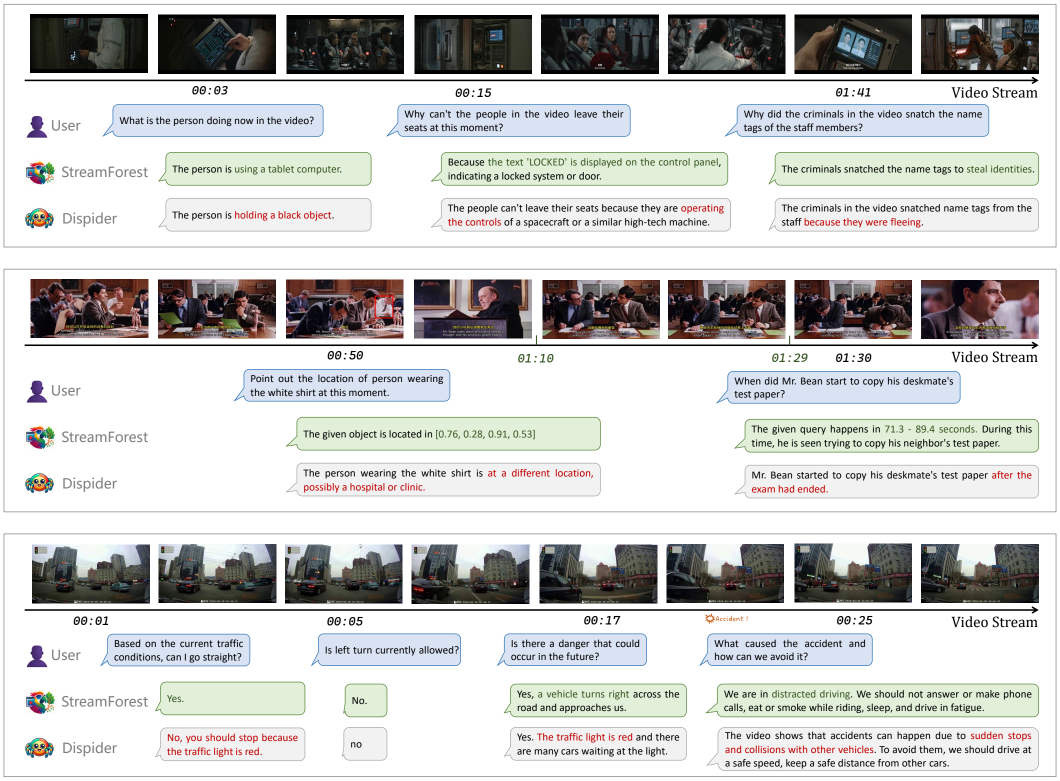
Table 14 details performance on OVO-Bench, where StreamForest again leads among open-source online video MLLMs with an overall average of 55.57%, outperforming Dispidier-7B (41.78%) and Flash-VStream-7B (33.15%). It excels in critical areas such as real-time visual perception (61.20%), backward tracing (52.02%), and forward active responding (53.49%), showcasing robust temporal

Table 14: Detailed evaluation results on OVO-Bench.

Model	# Frames	Real-Time Visual Perception							Backward Tracing				Forward Active Responding				Overall Avg.
		OCR	ACR	ATR	STU	FPD	OJR	Avg.	EPM	ASI	HLD	Avg.	REC	SSR	CRR	Avg.	
Human	-	93.96	92.57	94.83	92.70	91.09	94.02	93.20	92.59	93.02	91.37	92.33	95.48	89.67	93.56	92.90	92.81
Proprietary MLLMs																	
Gemini 1.5 Pro [54]	1fps	85.91	66.97	79.31	58.43	63.37	61.96	69.32	58.59	76.35	52.64	62.54	35.53	74.24	61.67	57.15	63.00
GPT-4o [44]	64	69.80	64.22	71.55	51.12	70.30	59.78	64.46	57.91	75.68	48.66	60.75	27.58	73.21	59.40	53.40	59.54
Open-source Offline Video MLLMs																	
LLaVA-Video-7B [76]	64	69.80	59.63	66.38	50.56	72.28	61.41	63.34	51.18	64.19	9.68	41.68	34.10	67.57	60.83	54.17	53.06
LLaVA-OneVision-7B [28]	64	67.11	58.72	69.83	49.44	71.29	60.33	62.79	52.53	58.78	23.66	44.99	24.79	66.93	60.83	50.85	52.88
Qwen2-VL-7B [55]	64	69.13	53.21	63.79	50.56	66.34	60.87	60.65	44.44	66.89	34.41	48.58	30.09	65.66	50.83	48.86	52.70
InternVL2-8B [9]	64	68.46	58.72	68.97	44.94	67.33	55.98	60.73	43.10	61.49	27.41	44.00	25.79	57.55	52.92	45.42	50.05
LongVU-7B [50]	1fps	55.70	49.54	59.48	48.31	68.32	63.04	57.40	43.10	66.22	9.14	39.49	16.62	69.00	60.00	48.54	48.48
Open-source Online Video MLLMs																	
VideoLLM-online-8B [5]	2fps	8.05	23.85	12.07	14.04	45.54	21.20	20.79	22.22	18.80	12.18	17.73	-	-	-	-	12.84
Flash-VStream-7B [72]	1fps	25.50	32.11	29.31	33.71	29.70	28.80	29.86	36.36	33.78	5.91	25.35	5.44	67.25	60.00	44.23	33.15
Dispider-7B [47]	1fps	57.72	49.54	62.07	44.94	61.39	51.63	54.55	48.48	55.41	4.30	36.06	18.05	37.36	48.75	34.72	41.78
StreamForest-7B (Ours)	1fps	68.46	53.21	71.55	47.75	65.35	60.87	61.20	58.92	64.86	32.26	52.02	32.81	70.59	57.08	53.49	55.57

reasoning across both past and future events. These results position StreamForest as a practical and powerful solution for real-time video-language understanding.

F Qualitative Comparison



What is the person doing now in the video?

The person is using a tablet computer.

The person is holding a black object.

Why can't the people in the video leave their seats at this moment?

Because the text 'LOCKED' is displayed on the control panel, indicating a locked system or door.

The people can't leave their seats because they are operating the controls of a spacecraft or a similar high-tech machine.

Why did the criminals in the video snatch the name tags of the staff members?

The criminals snatched the name tags to steal identities.

The criminals in the video snatched name tags from the staff because they were fleeing.

Point out the location of person wearing the white shirt at this moment.

The given object is located in [0.76, 0.28, 0.91, 0.53]

The person wearing the white shirt is at a different location, possibly a hospital or clinic.

When did Mr. Bean start to copy his deskmate's test paper?

The given query happens in 71.3 - 89.4 seconds. During this time, he is seen trying to copy his neighbor's test paper.

Mr. Bean started to copy his deskmate's test paper after the exam had ended.

Based on the current traffic conditions, can I go straight?

Yes.

No, you should stop because the traffic light is red.

Is left turn currently allowed?

No.

no

Is there a danger that could occur in the future?

Yes, a vehicle turns right across the road and approaches us.

Yes. The traffic light is red and there are many cars waiting at the light.

What caused the accident and how can we avoid it?

We are in distracted driving. We should not answer or make phone calls, eat or smoke while riding, sleep, and drive in fatigue.

The video shows that accidents can happen due to sudden stops and collisions with other vehicles. To avoid them, we should drive at a safe speed, keep a safe distance from other cars.

Figure 7: Qualitative comparison between StreamForest and other method.

Figure 7 presents a qualitative comparison between our model and other method. In the top example, StreamForest demonstrates a superior ability to capture fine-grained visual details and maintain persistent memory over time, enabling more coherent and informed inferences. The middle example highlights StreamForest's strong spatiotemporal grounding capabilities, accurately localizing objects and events across space and time. The bottom example illustrates the model's potential in intelligent driving scenarios, where it delivers precise real-time perception and supports future predictions based on both historical and current observations.

G More Ablations

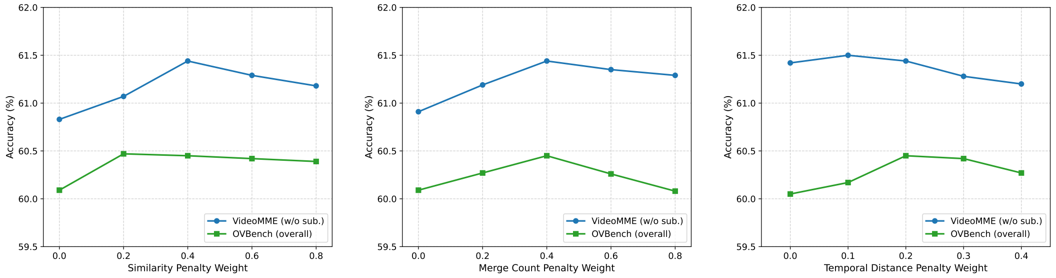


Figure 8: Ablation experiments of three penalty weights.

To evaluate the effect of each penalty on the effectiveness of long-term memory, we conducted ablation studies by varying the weights of the similarity penalty, merge count penalty, and temporal distance penalty. The results are presented in Figure 8, which reports the accuracy of both VideoMME and OVBench under different settings. It demonstrates that a balanced combination of penalty weights is more effective. Specifically, 0.4 for both similarity and merge count penalties, and 0.2 for the temporal distance penalty yield the most effective memory construction. This configuration achieves a favorable trade-off between preserving semantic coherence, maintaining diversity in memory representation, and ensuring reasonable temporal continuity.

H Efficiency of Multi-round Inference

We evaluated StreamForest's multi-round response efficiency by streaming a 600-second video to the model at a constant rate of 1 FPS. To isolate processing throughput from text generation latency, the model was constrained to produce a single-token response for each frame. Under this rigorous setting, StreamForest achieved an average processing speed of 9.9 FPS, which is competitive with VideoLLM-Online (12.3 FPS), a model renowned for its real-time capabilities. Crucially, StreamForest delivers this high efficiency without compromising its substantial superiority in reasoning accuracy over VideoLLM-Online. In stark contrast, Qwen2-VL, another model that also prioritizes reasoning accuracy, demonstrated severe performance bottlenecks. Its processing speed dropped below 1 FPS on a video merely two minutes long, and it encountered out-of-memory (OOM) errors on a single A100-80G GPU after processing only 79 frames.

Table 15: Comparison of multi-round inference speed.

Method	Resolution	FPS
Qwen2.5-VL	384	OOM
VideoLLM-Online	384	12.3
StreamForest (1k)	384	9.9

I Discussions

I.1 Limitations

Despite the effectiveness of our proposed method, several limitations remain that warrant further investigation. Our approach can only rely on computing inter-frame similarity to determine moments when the model should proactively produce outputs. Specifically, the method identifies local minima in similarity scores to detect transitions. However, this technique primarily captures coarse scene changes and often fails to accurately detect true semantic event boundaries. To address this limitation, one possible solution is to incorporate a lightweight MLLM as an auxiliary reminder module. This module could provide semantic-level guidance to support more precise and context-aware output decisions. These limitations suggest promising directions for future work.

I.2 Broader Impacts

Our proposed method shows strong potential for real-world streaming video understanding, especially in critical applications like autonomous driving. With domain-specific fine-tuning, it can be adapted to various downstream tasks that require continuous visual processing. As shown in the main text, the model performs well in autonomous driving scenarios, where accurate and timely perception is crucial for safety and decision-making. It can efficiently process live video streams while preserving fine-grained perception and long-term contextual memory. This capability is particularly valuable under limited computational resources, helping improve the reliability and responsiveness of intelligent systems in dynamic environments.

However, as with many vision-language models, potential negative social impacts must also be considered. If deployed without proper safeguards, models may inherit or amplify biases present in training data, leading to unreliable behavior. For instance, performance disparities across different environments or conditions (e.g., weather, lighting, or geographic location) could affect the robustness of StreamForest. To mitigate such risks, we should explore techniques for enhancing interpretability and controllability of streaming video models in safety contexts.