TimeTraveler: Temporal Strategy Planning with Time Dictionary for Streaming Video Understanding

Minkuk Kim[1] **Junho Kim**[2†] **Seong Tae Kim**[1†]

[1]Kyung Hee University [2]University of Illinois Urbana-Champaign

Abstract

Streaming video understanding (SVU) requires models to answer user queries over continuously evolving video streams. Unlike offline video QA, SVU must handle queries whose evidence may lie in the past, appear in the current scene, or emerge only after future observations. Existing streaming video systems often rely on compact visual memories or fixed evidence-access schemes, which can lose fine-grained details and fail to adapt the answering process to the temporal intent of each query. In this paper, we propose TimeTraveler, a streaming VQA framework that performs temporal strategy planning with a Time Dictionary. Inspired by the need to search across different temporal regions of a stream, TimeTraveler stores each observed moment as a timestamp-indexed structured caption and uses this dictionary as an explicit source of queryable evidence. Given a query, TimeTraveler determines whether to recall past evidence, read the present context, or wait for future observations, and then applies a strategy-specific evidence acquisition process. Comparative experiments on streaming video QA benchmarks show that TimeTraveler improves streaming question answering by preserving fine-grained temporal evidence and selecting the appropriate temporal strategy for each query. The project page is available at `https://geppa.github.io/TimeTraveler`.

1 Introduction

The rapid progress of Large Vision-Language Models (LVLMs) has expanded the scope of video understanding [1, 2, 3, 4, 5, 6, 7], shifting the field from task-specific recognition toward open-ended visual reasoning [8, 9, 10], temporal localization [11, 12], and interactive question answering [13, 14]. Despite this progress, most video-LLMs [15, 16] are still designed for an offline assumption: a holistic video is given in advance, and the model answers after processing the entire input. This assumption breaks down in streaming scenarios, where visual observations arrive causally, the stream may be unbounded, and user queries can be issued at arbitrary moments. In such settings, the required evidence is structurally heterogeneous. The system must not only support rich perception of incoming visual information, but also preserve past events for later access and remain responsive to events that can only be confirmed through future observation. These requirements make streaming video understanding (SVU) particularly challenging.

Recent studies on SVU fall into two main directions. One develops memory-centric mechanisms [17, 18, 19] that compress and retain stream observations (*e.g.,* prototypes, memory tokens, hierarchical summaries, or compressed KV caches) to preserve past information under limited budgets. The other branch [20, 21, 22, 23] improves online interaction by enabling real-time response scheduling, proactive activation, or streaming adaptation of offline video MLLMs, allowing the system to respond at more appropriate moments as the stream unfolds.

†Corresponding authors.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

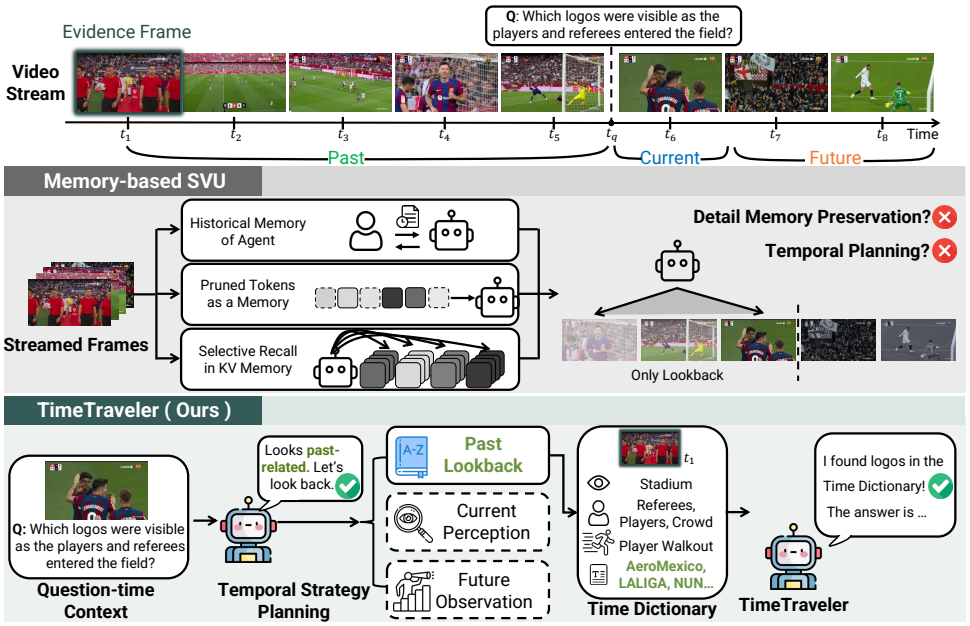


Figure 1: **Conceptual comparison between prior memory-based SVU and TimeTraveler.** Prior methods preserve streaming history as agentic memory [19], pruned tokens [22], or KV-memory recall [24], but rely on a single evidence-access path and do not ensure that retained evidence remains query-recoverable. In contrast, TimeTraveler plans whether each query should be grounded in the past, current scene, or future, and accesses a structured *Time Dictionary* accordingly.

Specifically, memory-based methods preserve evidence by converting the video stream into historical memory [19], caching visual information as KV memory [24, 18], or selectively storing query-relevant information via token pruning [22, 25]. To make unfolding streams tractable, these methods often encode streamed observations into latent tokens, visual prototypes, or compressed representations for stable and efficient retention. Although these approaches are effective for streaming memory, such representations can obscure fine-grained cues including small text, object attributes, spatial relations, or transient actions, making them difficult to recover at question time, as shown in Fig. 1. Since future queries are unknown during observation, cues that seem incidental when stored may later become answer-critical. Thus, beyond retaining a compressed history, SVU requires an explicit, fine-grained, and queryable memory representation that keeps such details recoverable at question time.

Moreover, preserving recoverable evidence alone does not determine how that evidence should be accessed for an arbitrary user query. Prior approaches [19, 18, 25] largely overlook when the evidence for answering a query should be sought, without explicitly modeling the temporal structure of user queries. We argue that SVU is fundamentally a temporal grounding problem, where each query implicitly specifies a temporal scope $\{\text{past}, \text{current}, \text{future}\}$. Accordingly, current queries rely on local evidence, past queries require retrieval from accumulated observations, and future queries must be monitored until the relevant event occurs.

To bridge this gap, in this paper, we propose TimeTraveler, a SVU framework that stores incoming observations as an explicit, fine-grained, and queryable memory representation, and accesses the stored evidence according to the temporal scope for the given user query. Specifically, TimeTraveler is built on two components: (i) Time Dictionary, which stores the stream as lightweight structured per-frame captions that decompose visual information within each frame into multiple semantic fields. Each timestamp is associated with explicit textual evidence covering scenes, objects, actions, text, spatial relations, and contextual details, making the memory both directly queryable and efficient to maintain in real-time streaming settings, and (ii) Temporal Strategy Planner for deciding whether a query should be answered from past memory, the current scene, or future observation. The model then follows the selected temporal strategy: for past queries, it performs memory lookback; for current queries, it refers to structured perception of the current scene; and for future queries, it carries out structured observation until the required evidence becomes available, all within a unified SVU system.

Through experiments on two streaming benchmarks [26, 27], we show that TimeTraveler consistently outperforms various baselines. More importantly, the results support our central claim that SVU depends not only on retaining fine-grained history (Table 3), but also on accessing the right evidence according to the temporal intent of each query (Table 4). By combining temporal strategy planning with a structured Time Dictionary, TimeTraveler improves evidence access across past, current, and future contexts, enabling more effective question answering in video streams.

Our contributions are summarized as follows:

- We propose TimeTraveler, a streaming video QA framework that stores fine-grained evidence in a lightweight Time Dictionary and retrieves it through query-aware Temporal Strategy Planning. Unlike compressed visual memory, our design preserves query-recoverable evidence over long streaming horizons while supporting efficient updates.
- We formulate SVU as a temporal grounding problem and identify temporal routing as a missing inductive bias. We introduce Temporal Strategy Planning, a query-aware mechanism that selects how evidence should be acquired from past, current, or future contexts.
- We validate TimeTraveler across multiple backbones, streaming video benchmarks, and efficiency metrics, and provide mechanistic analyses showing that temporal strategy planning and the Time Dictionary improve evidence selection and streaming question answering quality.

2 Related Work

2.1 Large Vision-Language Models

Early LVLMs [28, 29, 30] show that visual instruction tuning can connect pretrained language backbones [31, 4] with visual encoders for general-purpose multimodal conversation. Subsequent models improve this paradigm by scaling model and data, refining visual tokenization, and strengthening vision-language alignment [2, 32, 5, 1]. Video-oriented LVLMs [33, 34, 35, 6, 3] further extend these capabilities to temporal inputs by treating videos as sequences of images and introducing video-specific connectors, temporal aggregation modules, or video instruction-tuning pipelines. Caption-based approaches instead convert clips into text and reason over it with an LLM [36, 37], and retrieval over external memory further aids video and visual QA [38, 39, 40]. Despite strong progress on video QA and captioning, most of these methods are designed for offline or fixed-context inference, where the video is fully available before answering or has already been compressed into a finite context window. This assumption is restrictive for SVU, where frames arrive causally and a query may require evidence from past history, the current scene, or future observations.

2.2 Streaming Video Understanding

Streaming video understanding studies how multimodal models process continuously arriving visual evidence before the full video is available. A major line of work addresses the retention problem: how to maintain useful history from an unbounded stream under bounded computation and memory. StreamChat builds hierarchical memory for long-stream and multi-round interaction [41], Flash-VStream maintains compact memories for real-time long-video response [17], and StreamMem reduces streaming memory cost through visual cache compression [18]. Related memory-based or prototype-based systems summarize incoming observations into compact token memories, learned prototypes, or recurrent retained states [42, 19], and InfiniPot-V [43] enforces a length-independent cap on the visual KV cache through training-free compression during streaming. These methods make long-horizon processing practical by compressing accumulated context or reducing redundant visual tokens. In contrast, our framework keeps the retained history as explicit structured caption records: each sampled frame is decomposed into semantic fields such as actions, people, objects, scene context, text, and spatial relations, so the memory remains more structured and query-recoverable at question time rather than being collapsed into a compact visual state.

Another line of work focuses on online interaction and proactive response, asking when a streaming model should produce an output. VideoLLM-Online and related systems adapt Video-LLMs to interleaved video-text streams [21], while Dispider decomposes real-time interaction into perception, decision, and reaction modules [23]. StreamBridge converts offline Video-LLMs into proactive streaming assistants with a memory buffer and lightweight activation module [20], and QueryStream

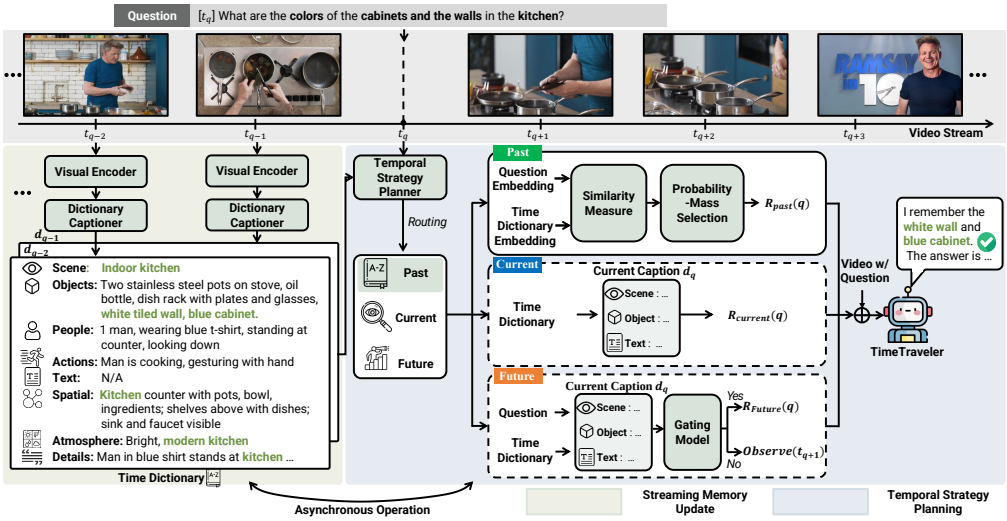


Figure 2: **Overview of TimeTraveler.** Incoming video frames are encoded into structured captions and stored in a timestamp-indexed Time Dictionary. Given a query, a Temporal Strategy Planner acquires evidence following temporal intent: past queries retrieve relevant captions through similarity-based lookback, current queries access local captions, and future queries continue observation until evidence appears. Then, the retrieved evidence is used to produce the final answer.

introduces query-aware pruning with relevance-triggered active response scheduling [22]. These approaches improve response timing and streaming efficiency, but they primarily decide whether or when the answer model should be activated, rather than explicitly modeling the temporal grounding decision implied by the user's query. In contrast, our work makes this intermediate decision explicit: before assembling evidence, the system first determines whether the query should be grounded in past memory, the current local context, or future observations, and then applies the corresponding evidence strategy over structured memory. Unlike tool selection in LLM agents [44, 45] or query-aware selection over already observed content [22, 25, 46], our planner predicts where in time the evidence lies, including evidence that has not yet appeared.

3 Proposed Method

Overview. We propose TimeTraveler, a streaming framework for query-conditioned evidence acquisition, as shown in Fig. 2. Our framework separates what to retain from the stream and how to access it for each query. Incoming frames are converted into structured caption records and stored in a causal Time Dictionary, together with sentence embeddings used only as lightweight retrieval handles. When a query arrives, a Temporal Strategy Planner selects one of three temporal strategies: past, current, or future. These respectively retrieve prior evidence with recency-aware lookback, read the query-time caption, or continue observing until a response condition or horizon is reached.

3.1 Time Dictionary for Structured Evidence Retention

The goal of the Time Dictionary is to retain queryable evidence from a causal video stream without collapsing the history only into a visual prototype or recurrent hidden state. Rather than compressing observations into latent visual states, we represent the stream as a temporally indexed structured representation that preserves fine-grained, query-recoverable evidence. Let f_i denote the i-th sampled frame and τ_i its timestamp. At each observed moment, sampled at 1 fps in our streaming pipeline, a frozen captioner Φ_{cap} processes the frame once following eight fields: scene, objects, people, actions, text, spatial relations, atmosphere, and details. This produces a bounded-length structured caption $c_i = \Phi_{\text{cap}}(f_i)$, written in a consistent schema of the form:

$$c_i = \{\texttt{Scene} : s_i^{\text{scene}},\ \texttt{Objects} : s_i^{\text{obj}},\ \texttt{People} : s_i^{\text{people}},\ \texttt{Actions} : s_i^{\text{act}},\ \texttt{Text} : s_i^{\text{text}},\ \texttt{Spatial} : s_i^{\text{spatial}},\ \texttt{Atmosphere} : s_i^{\text{atm}},\ \texttt{Details} : s_i^{\text{det}}\}. \tag{1}$$

Here, `Scene` summarizes locational and environmental context such as indoor or outdoor setting; `Objects` lists salient entities together with attributes such as color, size, and state; `People` describes count, appearance, clothing, and expression; `Actions` captures ongoing events and interactions among people and objects; `Text` records any visible signs, labels, subtitles, or logos; `Spatial` specifies layout and relative positions of key elements; `Atmosphere` summarizes lighting, color tone, time of day, weather, or mood; and `Details` provides a short coherent paragraph that captures contextual cues not fully covered by the other fields. The schema encourages the captioner to explicitly expose fine-grained cues that may later become relevant to an arbitrary query.

For efficient lookback, the system also computes an auxiliary sentence embedding for each caption. A frozen text encoder Φ_{emb} encodes the full caption as one textual sequence and L2-normalizes it to e_i. For each sampled frame i, we create a dictionary entry $d_i = (i, \tau_i, c_i, e_i)$, and the Time Dictionary is then updated causally by appending the new entry, $\mathcal{D}_i = \mathcal{D}_{i-1} \| d_i$, with $\mathcal{D}_0 = ()$.

Thus, when a query arrives at frame index i_q, the system can only access $\mathcal{D}_{i_q}$, ensuring that no future frame is used for past or current answering. The Time Dictionary preserves streamed observations as explicit timestamped text records, allowing fine-grained evidence to remain available over long time horizons without being diluted as the stream unfolds. The auxiliary embeddings serve only as lightweight handles for efficient lookback, not as the primary memory content. For memory-efficient retrieval, we maintain these embeddings in a finite bin queue covering roughly the most recent 600 seconds of the stream, while the text-based Time Dictionary remains the persistent evidence store. This horizon bounds index growth on potentially unbounded streams and can be extended at negligible cost (Appendix A.4). This timestamp-indexed structure makes the accumulated stream addressable by different temporal strategies rather than forcing all queries to share a single compact memory state.

3.2 Query-Conditioned Temporal Strategy Planning

We then formulate temporal strategy selection as a query-conditioned policy over a discrete temporal action space $\{\text{past}, \text{current}, \text{future}\}$. Rather than predicting an answer directly, the policy determines how evidence should be acquired from the streaming setup. For example, a question such as "*What did the person pick up earlier?*" requires retrieving evidence from past memory, whereas "*What is the person doing now?*" is answered from the current scene. A future-facing instruction such as "*Tell me when the car stops.*" should not be answered immediately, but should instead be monitored until the requested condition becomes true or the observation horizon is reached. Temporal Strategy Planning captures these distinctions by assigning each query to the strategy that matches its temporal scope.

Temporal strategy planner. Given a query q issued at timestamp τ_q and latest observed frame index i_q, the planner serializes q together with a short recent-caption context and prompts a frozen lightweight instruction model to output exactly one temporal strategy $\pi(q) \in \{\text{past}, \text{current}, \text{future}\}$. The prompt instructs the model to choose `future` when the query asks the system to wait for a condition that has not yet occurred, `past` when the answer requires information outside the recent context, and `current` otherwise. The role of the planner is not to answer the question, but to decide how evidence should be sought before answer generation.

Past strategy: online probabilistic lookback. For past-directed queries, the system performs online probabilistic lookback over the observed Time Dictionary $\mathcal{D}_{i_q}$. Rather than selecting a fixed top-K set, the past strategy first forms a similarity-based candidate pool and then adaptively selects evidence using a recency-aware softmax distribution.

We first restrict the search to causal entries whose timestamps do not exceed the query time:

$$\mathcal{I}_q = \{i \mid d_i \in \mathcal{D}_{i_q},\ \tau_i \le \tau_q\}. \tag{2}$$

From this causal set, we compute a semantic similarity score $u_i = \cos(\phi(q), e_i)$ for each $i \in \mathcal{I}_q$, where $\phi(q)$ denotes the normalized query embedding. We keep the top-N entries by u_i as a candidate index set $\mathcal{C}_N(q) \subseteq \mathcal{I}_q$. In our implementation, we use $N = 32$.

For each candidate entry, we form a recency-aware selection logit:

$$\ell_i = u_i - \frac{\tau_q - \tau_i}{\tau_r}, \qquad i \in \mathcal{C}_N(q), \tag{3}$$

where τ_r controls how strongly older evidence is penalized. The logits are converted into a probability distribution with temperature T_{past}:

$$p_i = \frac{\exp(\ell_i / T_{\text{past}})}{\sum_{j \in \mathcal{C}_N(q)} \exp(\ell_j / T_{\text{past}})}, \qquad i \in \mathcal{C}_N(q). \tag{4}$$

We then sort candidate indices in descending order of p_i, denoted by $(1), (2), \ldots, (|\mathcal{C}_N(q)|)$, and select the smallest prefix whose cumulative probability mass exceeds a threshold t_p:

$$K^\star = \min \left\{ k : \sum_{m=1}^{k} p_{(m)} \geq t_p \right\}. \tag{5}$$

We define the past evidence set as $\mathcal{R}_{\text{past}}(q) = \{d_{(m)} \mid 1 \leq m \leq K^\star\}$. This cumulative-mass cutoff makes the number of selected past entries adaptive: concentrated distributions yield a small evidence set, while diffuse distributions retain more candidates. Importantly, embeddings are used only to construct the candidate pool and selection distribution; the evidence passed to the answer model is the stored caption text. This strategy supports non-local lookback when the answer depends on earlier observations that may no longer be visible.

Current strategy: query-time caption. For current-directed queries, global memory lookback is unnecessary and can introduce distractors from semantically similar but temporally irrelevant past events. The current strategy therefore reads the caption record at the query time as $\mathcal{R}_{\text{current}}(q) = \{d_{i_q}\}$. This provides direct access to the dictionary caption record at the query time and avoids mixing current evidence with temporally irrelevant past entries.

Future strategy: observe-or-respond gating. For future-directed queries, the system enters a deferred observation mode. Rather than invoking the answer model immediately, it periodically checks whether later evidence satisfies the query condition. Let $W \in \mathbb{N}_{>0}$ denote the evidence window size, i.e., the number of most recent dictionary entries passed to the gate. A frozen lightweight gate model g, prompted with a fixed yes/no instruction, takes the query and the window $\mathcal{D}_{i-W+1:i}$ and predicts whether the condition has been satisfied:

$$b_i = g(q, \mathcal{D}_{i-W+1:i}) \in \{0, 1\}, \qquad i > i_q. \tag{6}$$

The system polls the gate at intervals of $\Delta_{\text{poll}} \in \mathbb{N}_{>0}$ frames after i_q, and continues polling up to a maximum observation horizon $T_{\max}$ (in seconds) measured from the query time τ_q. The first polling index whose gate output is positive is selected:

$$i^\star = \min\{i > i_q : i \equiv i_q \pmod{\Delta_{\text{poll}}},\ \tau_i - \tau_q \leq T_{\max},\ b_i = 1\}. \tag{7}$$

If no polling index satisfies the condition before the maximum horizon $T_{\max}$, the system falls back to the latest available evidence under the observation budget. If no future evidence is available, it falls back to the caption record at the query time. This strategy enables delayed response without repeatedly calling the full answer model: the gate is evaluated during observation, and the answer model is invoked only once after the gate fires or the horizon is reached.

Evidence assembly. Let $\mathcal{R}_{\text{past}}(q)$, $\mathcal{R}_{\text{current}}(q)$, and $\mathcal{R}_{\text{future}}(q)$ denote the evidence returned by the three strategies. The final evidence sequence is selected according to the planned temporal strategy:

$$\mathcal{R}(q) = \begin{cases} \mathcal{R}_{\text{past}}(q), & \pi(q) = \text{past}, \\ \mathcal{R}_{\text{current}}(q), & \pi(q) = \text{current}, \\ \mathcal{R}_{\text{future}}(q), & \pi(q) = \text{future}. \end{cases} \tag{8}$$

The selected entries are serialized with their timestamps and captions, and are then passed to a frozen long-context answer model LM_{QA}, which generates the final response.

3.3 Streaming Inference

At inference time, the system runs an observe-and-respond loop. The ingest process continuously applies Φ_{cap} and Φ_{emb} to incoming frames and appends the resulting entries to $\mathcal{D}$. Streaming updates

Table 1: **Evaluation results on OVO-Bench.** Overall accuracy is calculated as the macro average across taxonomies. Best results among open-source models are **boldfaced** and runner-up results are underlined. Proprietary model results are shown with gray background for reference.

Method	#Frames	Real-Time Visual Perception							Backward Tracing				Forward Act. Responding				Overall
		OCR	ACR	ATR	STU	FPD	OJR	Avg.	EPM	ASI	HLD	Avg.	REC	SSR	CRR	Avg.	Avg.
Human																	
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 Models (Offline)																	
Gemini 1.5 Pro [47]	1 fps	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 [48]	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 Models																	
LLaVA-Video-7B [33]	64	69.13	58.72	68.83	49.44	74.26	59.78	63.52	56.23	57.43	7.53	40.40	34.10	**69.95**	60.42	**54.82**	52.91
LLaVA-OV-7B [2]	64	66.44	57.80	73.28	53.37	71.29	61.96	64.02	54.21	55.41	21.51	43.71	25.64	67.09	58.75	50.50	52.74
LLaVA-N-Video-7B [49]	64	69.80	59.60	66.40	50.60	72.30	61.40	63.30	51.20	64.20	9.70	41.70	34.10	67.60	**60.80**	54.20	53.10
Qwen2-VL-7B [50]	64	60.40	50.46	56.03	47.19	66.34	55.43	55.98	47.81	35.48	**56.08**	46.46	31.66	65.82	48.75	48.74	50.39
InternVL-V2-8B [51]	64	67.11	60.55	63.79	46.07	68.32	56.52	60.39	48.15	57.43	24.73	43.44	26.50	59.14	54.14	46.60	50.15
LongVU-7B [52]	1 fps	55.70	49.50	59.50	48.30	68.30	63.00	57.40	43.10	66.20	9.10	39.50	16.60	69.00	60.00	48.50	48.50
Open-Source Online Models																	
Flash-VStream-7B [17]	1 fps	24.16	29.36	28.45	33.71	25.74	28.80	28.37	39.06	37.16	5.91	27.38	8.02	67.25	60.00	45.09	33.61
VideoLLM-Online-8B [21]	2 fps	8.05	23.85	12.07	14.04	45.54	21.20	20.79	22.22	18.80	12.18	17.73	-	-	-	-	-
VideoLLM-EyeWO [53]	1 fps	24.16	27.52	31.89	32.58	44.55	35.87	32.76	39.06	38.51	6.45	28.00	-	-	-	-	-
Dispider-7B [23]	1 fps	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
TimeChat-Online-7B [25]	1 fps	69.80	48.60	64.70	44.90	68.30	55.40	58.60	53.90	62.80	9.10	42.00	32.50	36.50	40.00	36.40	45.60
StreamAgent-7B [24]	1 fps	71.20	53.20	63.60	53.90	67.30	58.70	61.30	54.80	58.10	25.80	41.70	**35.90**	48.40	52.00	45.40	49.40
QueryStream-7B [22]	1 fps	74.50	47.70	70.70	46.60	71.30	57.60	61.40	54.20	63.50	8.60	42.10	33.20	43.10	40.80	39.03	47.51
InternVL3.5-8B [5]	1fps	79.87	65.14	69.83	**61.80**	70.30	65.22	68.69	33.00	59.46	4.30	32.25	32.66	60.10	35.00	42.59	47.84
+ TimeTraveler	1fps	**91.95**	70.64	75.00	**61.80**	**75.25**	**77.17**	**75.30**	33.67	60.81	9.68	34.72	22.21	50.24	38.75	37.06	49.03
Qwen3-VL-8B [16]	1fps	75.17	60.55	66.38	53.93	66.34	57.07	63.24	52.53	67.57	10.22	43.44	33.09	65.50	46.67	48.42	51.70
+ TimeTraveler	1fps	87.25	**73.39**	**76.72**	59.55	68.32	74.46	73.28	**59.26**	**69.59**	25.27	**51.37**	32.23	67.25	48.75	49.41	**58.02**

are efficient because each sampled frame produces one bounded caption record and one embedding append, while future queries require only lightweight periodic gating before a single final answer-generation step. When a query arrives, the dispatch process prompts the Temporal Strategy Planner once. Past and current queries invoke $\mathrm{LM}_{\mathrm{QA}}$ immediately using evidence available at i_q. Future queries create a polling state and evaluate the prompt-based gate only at newly reached polling indices while the ingest process continues updating the Time Dictionary.

Once the gate fires, or the maximum horizon is reached, the system invokes $\mathrm{LM}_{\mathrm{QA}}$ exactly once with the selected future evidence. At QA time, the answer model takes as input the query q, the video observations available at decision time, and the temporal evidence assembled by our method, $\mathcal{R}(q)$, and produces the final VQA answer, $a = \mathrm{LM}_{\mathrm{QA}}\big(v, q, \mathcal{R}(q)\big)$, where v denotes the video frames available when the system answers the query and a is the generated answer.

4 Experiments

4.1 Experimental Setting

Benchmarks and Metrics. We evaluate TimeTraveler on two complementary streaming video understanding benchmarks: OVO-Bench [27] and StreamingBench [26]. OVO-Bench measures three challenging capabilities in online video understanding: real-time visual perception, backward tracing, and forward active responding. Following the benchmark protocol, we report category-wise accuracies together with the macro average across the three taxonomies as the overall score. StreamingBench evaluates streaming comprehension over 18 tasks spanning real-time visual understanding, omni-source understanding, and contextual understanding.

Implementation Details. TimeTraveler builds on the Qwen3 family [16]: it uses a frozen 8-bit-quantized Qwen3-VL-2B as both the visual captioner and the planner/gate model, Qwen3-Embedding-0.6B as the text encoder, and Qwen3-VL-8B as the frozen answer model; we also evaluate with InternVL3.5-8B [5]. Incoming frames are sampled at 1 FPS and converted into structured caption records with eight semantic fields, while one auxiliary embedding per caption is maintained only as a retrieval handle in a finite cache covering roughly the most recent 600 seconds. For past-directed queries, we use a candidate pool size of $N = 32$ and set $\tau_r = 150$, $T_{\mathrm{past}} = 0.02$, and $t_p = 0.70$; for future-directed queries, we set $\Delta_{\mathrm{poll}} = 1$ frame, $T_{\max} = 30$ s, and $W = 1$. The answer model receives the 1-fps frames observed up to the decision time, capped at 512 frames for Qwen3-VL-8B and 128 frames for InternVL3.5-8B due to context limits; all other settings remain identical across backbones. All components are used in frozen form and only prompt-based planning is applied at inference time. Experiments are run on a single NVIDIA RTX PRO 6000 Blackwell GPU.

Table 2: **Performance comparison on StreamingBench.** Best results among open-source models are **boldfaced** and runner-up results are underlined.

Model	#Frames	Real-Time Visual Understanding											Omni-Source Understanding					Contextual Understanding					Overall
		OP	CR	CS	ATP	EU	TR	PR	SU	ACP	CT	All	ER	SCU	SD	MA	All	ACU	MCU	SQA	PO	All	
Human																							
Human‡	-	89.47	92.00	93.60	91.47	95.65	92.52	88.00	88.75	89.74	91.30	91.46	88.00	88.24	93.60	90.27	90.26	88.80	90.40	95.00	100.00	93.55	91.66
Proprietary MLLMs																							
Gemini 1.5 pro [47]	1 fps	79.02	80.47	83.54	79.67	80.00	84.74	77.78	64.23	71.95	48.70	75.69	46.80	39.60	74.90	80.00	60.22	51.41	40.73	54.80	45.10	48.73	67.07
GPT-4o [48]	64	77.11	80.47	83.91	76.47	70.19	83.80	66.67	62.19	69.12	49.22	73.28	41.20	37.20	43.60	56.00	44.50	41.20	38.40	32.80	56.86	38.70	60.15
Claude 3.5 Sonnet [54]	20	80.49	77.34	82.02	81.73	72.33	75.39	61.11	61.79	69.32	43.09	72.44	31.60	34.00	32.80	48.80	36.80	38.40	34.80	34.40	64.71	37.70	57.68
Open-Source Offline Video MLLMs																							
LLaVA-OV-7B [2]	32	80.38	74.22	76.03	80.72	72.67	71.65	67.59	65.45	65.72	45.08	71.12	40.80	**37.20**	33.60	44.80	38.40	35.60	36.00	27.27	29.55	32.74	56.36
Qwen2-VL-7B [50]	0.2–1 fps†	75.20	82.81	73.19	77.45	68.32	71.03	72.22	61.19	61.47	46.11	69.04	41.20	22.00	32.80	43.60	34.90	31.20	26.00	39.60	22.73	31.66	54.14
MiniCPM-V 2.6-8B [55]	32	71.93	71.09	77.92	75.82	64.60	65.73	70.37	56.10	62.32	53.37	67.44	40.80	24.00	34.00	41.20	35.00	34.00	31.60	41.92	22.22	34.97	53.85
InternVL-V2-8B [51]	16	68.12	60.94	69.40	77.12	67.70	62.93	59.26	53.25	54.96	56.48	63.72	37.60	26.40	37.20	42.00	35.80	32.00	31.20	32.32	40.91	32.42	51.40
Kangaroo-7B [56]	64	71.12	**84.38**	70.66	73.20	67.08	61.68	56.48	55.69	62.04	38.86	64.60	37.60	31.20	28.80	39.20	34.20	32.80	26.40	33.84	16.00	30.06	51.10
LongVA-7B [57]	128	70.03	63.28	61.20	70.92	62.73	59.50	61.11	53.66	54.67	34.72	59.96	39.60	32.40	28.00	41.60	35.40	32.80	29.60	30.30	15.91	29.95	48.66
VILA-1.5-8B [58]	14	53.68	49.22	70.98	56.86	53.42	53.89	54.63	48.78	50.14	17.62	52.32	41.60	26.40	28.40	36.00	33.10	26.80	34.00	23.23	17.65	27.35	43.20
Video-CCAM-14B [59]	96	56.40	57.81	65.30	62.75	64.60	51.40	42.59	47.97	49.58	31.61	53.96	33.60	22.00	28.40	34.80	29.70	27.60	24.40	16.67	22.73	22.88	42.53
Video-LLaMA2-7B [60]	32	55.86	55.47	57.41	58.17	52.80	43.61	39.81	42.68	45.61	35.23	49.52	30.40	32.40	30.40	36.00	32.40	24.80	26.80	18.67	0.00	21.93	40.40
Open-Source Online Video MLLMs																							
Flash-VStream-7B [17]	-	25.89	43.57	24.91	23.87	27.33	13.08	18.52	25.20	23.87	48.70	23.23	25.91	24.90	25.60	28.40	26.00	24.80	25.20	26.80	1.96	24.12	24.04
VideoLLM-online-8B [21]	2 fps	39.07	40.06	34.49	31.05	45.96	32.40	31.48	34.16	42.49	27.89	35.99	31.20	26.51	24.10	32.00	28.45	24.19	29.20	30.80	3.92	26.55	32.48
Dispider-7B [23]	1 fps	74.92	75.53	74.10	73.08	74.44	59.92	76.14	62.91	62.16	45.80	67.63	35.46	25.26	38.57	43.34	35.66	39.62	27.65	34.80	25.34	33.61	53.12
TimeChat-Online-7B [25]	1 fps	80.22	82.03	79.50	83.33	76.10	78.50	78.70	64.63	69.60	57.98	75.36	–	–	–	–	–	–	–	–	–	–	–
QueryStream-7B [22]	1 fps	**82.38**	**84.38**	79.18	82.37	77.99	81.31	78.70	65.04	69.32	47.34	75.32	–	–	–	–	–	–	–	–	–	–	–
InternVL3.5-8B [5]	1fps	78.86	82.81	85.17	81.41	77.99	79.75	83.33	70.73	76.99	55.32	77.60	40.40	30.80	42.80	53.60	41.90	42.40	36.40	**48.40**	40.00	41.80	61.71
+ TimeTraveler	1fps	82.11	82.81	**89.27**	84.62	**83.65**	87.54	87.04	**73.17**	**77.56**	53.72	**80.72**	**48.40**	34.40	**48.00**	51.20	**45.50**	**47.60**	**44.80**	46.80	53.23	48.10	**65.64**
Qwen3-VL-8B [16]	1fps	75.10	76.60	80.10	81.10	77.40	71.70	77.80	61.40	69.30	55.90	72.80	41.20	32.00	46.80	**55.20**	43.80	40.00	31.60	34.40	52.40	36.75	59.33
+ TimeTraveler	1fps	81.84	79.69	87.70	**85.58**	76.73	**90.97**	**87.96**	69.11	75.28	**59.04**	80.16	43.20	26.40	44.00	49.20	40.70	44.40	41.60	48.00	**62.90**	**49.20**	64.51

Table 3: **Comparison of memory representations under an equal token budget on OVO-Bench.** #Tokens denotes the average number of memory tokens injected into the answer model at inference time. Qwen3-VL-8B is used as the backbone.

Memory Type	**#Tokens**	**Real-Time Acc.**	**Backward Acc.**	**Forward Acc.**	**Overall Acc.**
No Memory	–	63.24	43.44	48.42	51.70
Visual Prototype Memory	203.11	66.19	44.16	**51.36**	53.90
Flat Text Memory	236.60	70.98	48.57	47.83	55.79
Structured Dictionary (Ours)	245.23	**73.28**	**51.37**	49.41	**58.02**

4.2 Comparison with State-of-the-art Methods

Tables 1 and 2 present results on OVO-Bench and StreamingBench, respectively. TimeTraveler achieves the best overall performance among open-source models on both benchmarks, consistently improving both Qwen3-VL (51.70→58.02 on OVO-Bench; 59.33→64.51 on StreamingBench) and InternVL3.5 (47.84→49.03; 61.71→65.64). The gains align with our two design goals: the Time Dictionary improves detail-sensitive tasks (e.g., OCR, TR) where fine-grained cues must be preserved, while temporal strategy planning benefits temporally grounded tasks (e.g., EPM, PO) that require selecting evidence from the right temporal scope. A per-category analysis is provided in Appendix A.6. With the same Qwen2.5-VL-7B backbone, TimeTraveler also exceeds QueryStream and StreamAgent by 5.01 and 3.12 points overall on OVO-Bench (Appendix A.5).

4.3 Effect of TimeTraveler

Effect of Structured Time Dictionary. Table 3 compares alternative memory representations under a similar memory-token budget on queries that require recovering fine-grained past visual details. We compare three memory designs: *visual prototype memory*, which clusters accumulated visual tokens into K representative prototypes via k-means over the streaming sequence (K=2, configured to match the token budget of our method for fair comparison); *flat text memory*, which stores a single unstructured free-form caption per frame; and our *structured dictionary*, which decomposes each caption into eight semantic fields. Relative to the no-memory baseline, both visual prototype memory and flat text memory improve performance, confirming that retaining historical information is essential for streaming video QA. However, the proposed structured dictionary achieves the strongest overall result, improving the direct Qwen3-VL baseline from 51.70 to 58.02 and outperforming flat text memory by 2.23 points overall, with consistent gains in both real-time and backward accuracy. This suggests that the benefit does not come merely from storing more past content, but from organizing memory into semantically typed fields that preserve fine-grained details while making relevant evidence easier to retrieve.

Effect of Temporal Strategy Planning. Table 4 provides direct evidence for temporal strategy planning. While all memory-based settings outperform the no-memory baseline, our routing achieves

Table 4: **Effect of temporal routing on OVO-Bench Backward Tracing.** EPM: Episodic Memory, ASI: Action Sequence Identification, HLD: Hallucination Detection. †Average over five runs.

Setting	Backward Tracing			
	EPM	HLD	ASI	Avg.
Baseline (no memory) [16]	52.53	10.22	67.57	43.44
All-to-Past Lookback	57.91	**25.81**	67.57	50.43
Random Routing†	52.12	19.78	63.24	45.05
Our Routing	**59.26**	25.27	**69.59**	**51.37**

Table 5: **Schema-group ablation for the Time Dictionary.** Each row removes one of four schema groups while keeping all else fixed. Overall is computed only over the listed subtasks, not over the full benchmark.

Setting	OVO-Bench			StreamingBench		
	OCR	OJR	Overall	TR	SU	Overall
Baseline [16]	75.17	57.07	66.12	71.70	61.40	67.23
Full Schema (Ours)	87.25	74.46	80.86	90.97	69.11	81.48
w/o Background	83.89	69.57	76.73	90.97	69.11	81.48
w/o Entity-Centric	83.89	72.83	78.36	91.59	67.07	80.95
w/o Explicit Text	83.22	73.37	78.30	85.98	68.29	78.31
w/o Contextual Sent.	85.23	71.74	78.49	89.41	67.07	79.72

Table 6: **Confusion matrix of the Temporal Strategy Planner on StreamingBench.** The 150 queries are sampled 50 each from Real-Time, Contextual, and Proactive Output tasks. Rows: human-assigned temporal label; columns: planner prediction.

Human	Current	Past	Future	Recall
Current (n=70)	**47**	23	0	67.1
Past (n=30)	6	**24**	0	80.0
Future (n=50)	0	2	**48**	96.0
Overall agreement: 79.3% (GPT-4o 86.7%, GPT-5 85.3%)				

Table 7: **Effect of the number of visual frames given to the answer model on OVO-Bench.** Qwen3-VL-8B is used as the answer model, and Time Dict. ✓ denotes the full TimeTraveler framework.

Frames	Time Dict.	RT	BW	FWD	Overall
512 (baseline)	✗	63.24	43.44	48.42	51.70
512 (Ours)	✓	73.28	51.37	**49.41**	**58.02**
1	✓	**77.78**	52.14	41.50	57.14
0	✓	68.94	**55.47**	34.84	53.08

the best backward-tracing average (51.37) with top scores on EPM (59.26) and ASI (69.59). Notably, indiscriminate lookback is not universally beneficial: some backward-tracing questions still depend on current grounding, and always retrieving past evidence can introduce temporally irrelevant distractors. Query-conditioned routing preserves the benefit of memory retrieval while avoiding unnecessary access to irrelevant history, confirming that streaming VQA requires selecting the appropriate temporal strategy per query, not just retaining past evidence. Here, Random Routing assigns each query to the past, current, or future strategy uniformly at random instead of using the planner; other routing variants are reported in Appendix A.8.

4.4 Diagnostic Analysis of TimeTraveler

Planner Accuracy. On 150 human-labeled StreamingBench queries (Appendix A.7), the planner agrees with the human labels on 79.3% of queries (GPT-4o: 86.7%, GPT-5: 85.3%), and its future predictions reach 100% precision and 96% recall (Table 6). Most disagreements (23 of 31) route a current query to past, which may add irrelevant history, while errors that omit the required evidence are rare (8 of 150). Replacing the planner's routes with the human labels improves accuracy on this subset from 58.7% to 61.3%, indicating a limited downstream impact of routing errors.

Evidence Coverage. We randomly sampled 100 past-routed OVO Backward queries and manually inspected whether the answer-critical evidence appears in the selected captions, in their source frames, or elsewhere in the Time Dictionary (details in Appendix A.7). The selected captions contain the evidence for 48% of the queries, and the dictionary contains it for 65% in total. For 10%, the evidence is visible in a selected source frame but missing from its caption, and for 17%, it is stored in the dictionary but not retrieved. For the remaining 25%, no supporting caption is found; since only the source frames of the selected captions were inspected, these cases may stem from the 1-fps sampling or from omissions in captions of non-selected frames.

Multi-moment Retrieval. Past lookback provides the selected captions jointly in temporal order and often covers multiple moments. For Forward instances routed to past, it retrieves 2.82 captions on average and up to 19; among the 619 instances with $K^\star \geq 2$, the retrieved captions span a median of 6 s and a 90th percentile of 19 s, with 28.3% spanning more than 10 s.

Effect of Visual Frames. Reducing the visual input given to the answer model from 512 frames to a single frame lowers the overall score from 58.02 to 57.14 (Table 7), mostly on Forward questions

that involve ongoing events (49.41→41.50). Notably, even without any frame, TimeTraveler still outperforms the 512-frame baseline (53.08 vs. 51.70) and attains the highest Backward accuracy (55.47), showing that the Time Dictionary by itself serves as an effective memory of the stream.

4.5 Ablation Study on Time Dictionary Schema

Table 5 reports a schema-group ablation that removes one semantic field group at a time while keeping all other components unchanged. The full schema achieves the best overall performance on both benchmarks. Removing *Explicit Text* causes the largest drop on Text-Rich Understanding (TR: 90.97 → 85.98) and Optical Character Recognition (OCR: 87.25 → 83.22), confirming that fine-grained on-screen cues such as signs, logos, and labels require a dedicated field to remain recoverable at query time. Removing *Entity-Centric Dynamics* (Objects + People + Actions + Spatial) degrades OCR (87.25 → 83.89), Object Recognition (OJR: 74.46 → 69.57), and Spatial Understanding (SU: 69.11 → 67.07), showing that object-level attributes and inter-entity relations likewise benefit from explicit decomposition. The remaining two groups (Background, Contextual Sentence) also contribute to overall performance, but their removal patterns are less task-specific; we attribute this to the fact that the eight schema fields are not fully independent, and the model leverages overlapping cues across fields to partially compensate for a missing group. Altogether, these results support our design: structured decomposition into typed fields preserves fine-grained evidence that would otherwise be lost in a flat or compressed representation.

4.6 Asynchronous Streaming and Query Latency

Streaming-state updates and query answering are executed asynchronously and therefore should not be collapsed into a single end-to-end latency number. Under a 1 fps input stream, maintaining the Time Dictionary requires 637.51 ms per second of video, which stays within the real-time budget and leaves headroom for continuous online operation. Once this state has been prepared, query-time reasoning introduces only modest overhead over direct visual QA: the total latency is 574.44 ms for past queries, 557.62 ms for current queries, and 565.11 ms for future queries with one polling step, compared with 547.22 ms for the visual-only baseline, corresponding to only 10–27 ms of additional delay depending on the temporal branch. This breakdown reflects the practical advantage of our design: the heavier video processing cost is amortized during streaming, while user-time inference operates on compact structured state, allowing responsive query answering without reprocessing the full visual history for each question. A detailed latency breakdown is provided in Table 8. Scalability beyond 1 fps and a comparison with InfiniPot-V [43] are provided in Appendices A.2 and A.3.

5 Discussion and Conclusion

Limitation. TimeTraveler relies on caption-based textual evidence as the persistent memory representation. While this improves query recoverability and efficiency, it may inherit errors from the captioner, especially for small objects, ambiguous actions, or OCR under low resolution; answer-critical details omitted from a caption cannot be recovered from the dictionary (10% of the sampled queries in Section 4.4). The current planner and future gate are prompt-based frozen models, so misclassification of temporal scope can lead to suboptimal evidence selection. Further limitations and future directions are discussed in Appendix D.

Conclusion. We present TimeTraveler, a streaming video understanding framework built on two complementary ideas: storing past observations in an explicit and queryable structure, and matching each query to the appropriate temporal scope. TimeTraveler maintains a lightweight Time Dictionary that records each observed moment as structured textual evidence, preserving fine-grained cues in a more recoverable form. On top of this memory, a Temporal Strategy Planner determines whether the answer should be grounded in past observations, the current scene, or future evidence, enabling query-specific evidence access. Extensive experiments and analyses on OVO-Bench and StreamingBench show that this combination improves both detail-sensitive and temporally grounded reasoning, while maintaining practical efficiency under asynchronous streaming deployment.

Acknowledgments

This work was supported in part by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2024-00334321), by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (No. IITP-2026-RS-2023-00258649) supervised by the IITP (Institute for Information and Communications Technology Planning and Evaluation), and by Global - Learning & Academic research institution for Master's·PhD students, and Postdocs(G-LAMP) Program of the National Research Foundation of Korea(NRF) grant funded by the Ministry of Education(No. RS-2025-25442355), by Institute of Information and Communications Technology Planning and Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2022-II220078, RS-2025-02653113).

References

[1] 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.

[2] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. LLaVA-OneVision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024.

[3] 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, 2025.

[4] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing GPT-4 with 90%* ChatGPT quality, March 2023.

[5] Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. InternVL3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025.

[6] 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.

[7] Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Khan. Video-ChatGPT: Towards detailed video understanding via large vision and language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12585–12602, 2024.

[8] Ziyang Wang, Shoubin Yu, Elias Stengel-Eskin, Jaehong Yoon, Feng Cheng, Gedas Bertasius, and Mohit Bansal. VideoTree: Adaptive tree-based video representation for LLM reasoning on long videos. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 3272–3283, 2025.

[9] Kaituo Feng, Kaixiong Gong, Bohao Li, Zonghao Guo, Yibing Wang, Tianshuo Peng, Junfei Wu, Xiaoying Zhang, Benyou Wang, and Xiangyu Yue. Video-R1: Reinforcing video reasoning in MLLMs. *arXiv preprint arXiv:2503.21776*, 2025.

[10] 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.

[11] Shuming Liu, Chen-Lin Zhang, Chen Zhao, and Bernard Ghanem. End-to-end temporal action detection with 1B parameters across 1000 frames. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18591–18601, 2024.

[12] Jinhui Ye, Zihan Wang, Haosen Sun, Keshigeyan Chandrasegaran, Zane Durante, Cristobal Eyzaguirre, Yonatan Bisk, Juan Carlos Niebles, Ehsan Adeli, Li Fei-Fei, et al. Re-thinking temporal search for long-form video understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 8579–8591, 2025.

[13] 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.

[14] Peng Jin, Ryuichi Takanobu, Wancai Zhang, Xiaochun Cao, and Li Yuan. Chat-UniVi: Unified visual representation empowers large language models with image and video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13700–13710, 2024.

[15] 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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4122–4134, 2025.

[16] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025.

[17] Haoji Zhang, Yiqin Wang, Yansong Tang, Yong Liu, Jiashi Feng, and Xiaojie Jin. Flash-VStream: Efficient real-time understanding for long video streams. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 21059–21069, 2025.

[18] Yanlai Yang, Zhuokai Zhao, Satya Narayan Shukla, Aashu Singh, Shlok Kumar Mishra, Lizhu Zhang, and Mengye Ren. StreamMem: Query-agnostic KV cache memory for streaming video understanding. *arXiv preprint arXiv:2508.15717*, 2025.

[19] 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.

[20] Haibo Wang, Bo Feng, Zhengfeng Lai, Mingze Xu, Shiyu Li, Weifeng Ge, Afshin Dehghan, Meng Cao, and Ping Huang. StreamBridge: Turning your offline video large language model into a proactive streaming assistant. *arXiv preprint arXiv:2505.05467*, 2025.

[21] 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.

[22] Kairui Zhang, Zhenyu Yang, Bing Wang, Shengsheng Qian, and Changsheng Xu. QueryStream: Advancing streaming video understanding with query-aware pruning and proactive response. In *The Fourteenth International Conference on Learning Representations*, 2026.

[23] 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. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 24045–24055, 2025.

[24] Haolin Yang, Feilong Tang, Lingxiao Zhao, Xiang An, Ming Hu, Huifa Li, Xinlin Zhuang, Yifan Lu, Xiaofeng Zhang, Abdalla Swikir, et al. StreamAgent: Towards anticipatory agents for streaming video understanding. *arXiv preprint arXiv:2508.01875*, 2025.

[25] 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. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pages 10807–10816, 2025.

[26] Junming Lin, Zheng Fang, Chi Chen, Haoxuan Cheng, Zihao Wan, Fuwen Luo, Ziyue Wang, Peng Li, Yang Liu, and Maosong Sun. StreamingBench: Assessing the gap for MLLMs to achieve streaming video understanding. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 12147–12151. IEEE, 2026.

[27] Junbo Niu, Yifei Li, 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? In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 18902–18913, 2025.

[28] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Advances in Neural Information Processing Systems*, 2023.

[29] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. InstructBLIP: Towards general-purpose vision-language models with instruction tuning. In *Advances in Neural Information Processing Systems*, 2023.

[30] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. *arXiv preprint arXiv:2310.03744*, 2023.

[31] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[32] Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Yuchen Duan, Hao Tian, Weijie Su, Jie Shao, et al. InternVL3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025.

[33] 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.

[34] 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.

[35] Junho Kim, Hyunjun Kim, Hosu Lee, and Yong Man Ro. SALOVA: Segment-augmented long video assistant for targeted retrieval and routing in long-form video analysis. *arXiv preprint arXiv:2411.16173*, 2024.

[36] Ce Zhang, Taixi Lu, Md Mohaiminul Islam, Ziyang Wang, Shoubin Yu, Mohit Bansal, and Gedas Bertasius. A simple LLM framework for long-range video question-answering. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 21715–21737, 2024.

[37] Ziqi Pang and Yu-Xiong Wang. MR. Video: "MapReduce" is the principle for long video understanding. *arXiv preprint arXiv:2504.16082*, 2025.

[38] Minkuk Kim, Hyeon Bae Kim, Jinyoung Moon, Jinwoo Choi, and Seong Tae Kim. Do you remember? Dense video captioning with cross-modal memory retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13894–13904, 2024.

[39] Minkuk Kim, Hyeon Bae Kim, Jinyoung Moon, Jinwoo Choi, and Seong Tae Kim. HiCM^2: Hierarchical compact memory modeling for dense video captioning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 4293–4301, 2025.

[40] Su Hyeon Lim, Minkuk Kim, Hyeon Bae Kim, and Seong Tae Kim. Retrieval-augmented natural language reasoning for explainable visual question answering. In *2024 IEEE International Conference on Image Processing Challenges and Workshops (ICIPCW)*, pages 4184–4189. IEEE, 2024.

[41] Jihao Liu, Zhiding Yu, Shiyi Lan, Shihao Wang, Rongyao Fang, Jan Kautz, Hongsheng Li, and Jose M Alvarez. StreamChat: Chatting with streaming video. *arXiv preprint arXiv:2412.08646*, 2024.

[42] 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.

[43] Minsoo Kim, Kyuhong Shim, Jungwook Choi, and Simyung Chang. InfiniPot-V: Memory-constrained KV cache compression for streaming video understanding. In *Advances in Neural Information Processing Systems*, 2025.

[44] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In *Advances in Neural Information Processing Systems*, 2023.

[45] Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. HuggingGPT: Solving AI tasks with ChatGPT and its friends in Hugging Face. In *Advances in Neural Information Processing Systems*, 2023.

[46] Minkuk Kim, Suyong Yun, Young Tae Kim, Jinyoung Moon, Jinwoo Choi, and Seong Tae Kim. ReQuest: Rethinking-based question-aware frame selection for long-form video QA. In *European Conference on Computer Vision*, pages 386–404. Springer, 2026.

[47] 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.

[48] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. GPT-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

[49] Feng Li, Renrui Zhang, Hao Zhang, Yuanhan Zhang, Bo Li, Wei Li, Zejun Ma, and Chunyuan Li. LLaVA-Interleave: Tackling multi-image, video, and 3D in large multimodal models. In *The Thirteenth International Conference on Learning Representations*, 2025.

[50] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024.

[51] 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.

[52] 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.

[53] Yulin Zhang, Cheng Shi, Yang Wang, and Sibei Yang. Eyes wide open: Ego proactive video-LLM for streaming video. *arXiv preprint arXiv:2510.14560*, 2025.

[54] Anthropic. Claude 3.5 Sonnet. `https://www.anthropic.com/news/claude-3-5-sonnet`, 2024. Accessed: 2026-05-05.

[55] 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.

[56] 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. *International Journal of Computer Vision*, 134(3):114, 2026.

[57] 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.

[58] 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.

[59] 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.

[60] 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.

Appendix Contents

A Additional Analysis of TimeTraveler **16**

A.1 Memory footprint 16
A.2 Scalability beyond 1 fps. 17
A.3 Comparison with memory-efficient streaming. 17
A.4 Retrieval horizon and index scaling. 18
A.5 Same-backbone comparison. 18
A.6 Per-category analysis. 18
A.7 Details of the planner and evidence-coverage evaluation. 18
A.8 Random routing variants. 19
A.9 Effect of gating on proactive output task. 19
A.10 Routing-based visual budget allocation. 19
A.11 Adaptive selection vs. fixed top-K. 20
A.12 Number of retrieved captions per question type. 20
A.13 Redundancy among retrieved captions. 20
A.14 Sensitivity to temperature T_{past}. 21
A.15 Sensitivity to nucleus threshold t_p. 21
A.16 Sensitivity to recency scale τ_r. 22
A.17 Textual Input Sensitivity 22
A.18 Qualitative Result of Lookback and Dictionary. 22

B Prompts in TimeTraveler **24**

B.1 Captioner Prompt 24
B.2 Planner Prompt 24
B.3 Gating Prompt 25

C Additional Benchmark Explanation **25**

C.1 OVO-Bench. 25
C.2 StreamingBench. 26

D Limitations and Broader Impacts **26**

Table 8: **Asynchronous latency breakdown of TimeTraveler.** The two blocks run independently: stream processing executes continuously, while query processing is triggered on demand. Stream latency is normalized to one second of input at 1 fps. For the past path, the QA cost is linearly interpolated between the measured 1- and 8-caption settings at the average number of retrieved captions ($\bar{K}^{\star}$=2.68 over past-routed OVO-Bench queries). Qwen3-VL-8B is used as the backbone.

Stream Processing (always-on)		
Stage	**Latency**	**TFLOPs**
Video decode	15.72 ms	–
Captioner	605.82 ms	1.53
Embedder	15.97 ms	0.20
Total (per 1 s)	**637.51 ms**	**1.73**

Query Processing (on-demand)			
Path	**Stage**	**Latency**	**TFLOPs**
Baseline	QA (Visual-only)	547.22 ms	1.42
Our Past	Scope planning	9.60 ms	1.39
	Retrieval	15.62 ms	0.03
	QA (2.68 cap)	549.22 ms	8.28
	Total	**574.44 ms**	**9.70**
Our Current	Scope planning	9.60 ms	1.39
	QA (1 cap)	548.02 ms	3.81
	Total	**557.62 ms**	**5.2**
Our Future	Scope planning	9.60 ms	1.39
	Gating ($\times$1; +7.49 ms/extra)	7.49 ms	5.73
	QA (1 cap)	548.02 ms	3.81
	Total	**565.11 ms**	**10.93**

Table 9: **Memory footprint comparison across memory designs on Qwen3-VL-8B [16].** Visual tokens per frame are measured at 240 for inputs resized to `max_pixels` $= 401408$. Visual prototype uses K=32 slots. Our structured dictionary stores 800-byte UTF-8 caption text and a 1024-dim FP16 embedding per frame. Retrieval embeddings are maintained in a finite queue covering the most recent 600 s, so their cost is capped beyond 10 minutes.

Memory Design	Per-frame	1 min	5 min	10 min	30 min	1 hour
Visual Full KV Cache	33.75 MB	1.98 GB	9.89 GB	19.78 GB	59.33 GB	118.65 GB
Visual Hidden State	1.88 MB	112.5 MB	562.5 MB	1.10 GB	3.30 GB	6.59 GB
Visual Prototype (K=32)	–	1.6 MB	1.6 MB	1.6 MB	1.6 MB	1.6 MB
Time Dictionary (Ours)						
Caption Text	0.78 KB	46 KB	234 KB	469 KB	1.4 MB	2.8 MB
Retrieval Embedding	2.00 KB	120 KB	600 KB	1.2 MB	1.2 MB	1.2 MB
Total	2.78 KB	167 KB	834 KB	1.6 MB	2.6 MB	4.0 MB

A Additional Analysis of TimeTraveler

A.1 Memory footprint

Table 9 quantifies the storage cost that underlies our design choice of textual over visual memory. A central motivation of TimeTraveler is that streaming memory should remain both *fine-grained* and *lightweight*; visual representations can preserve perceptual detail but scale linearly with stream length at a prohibitive rate, while compressed prototypes(K=32 slots, constructed via k-means clustering of visual tokens) stay compact but sacrifice queryable detail. Our structured dictionary occupies only 2.78 KB per frame—roughly $690\times$ less than a visual hidden state and over $12{,}000\times$ less than a full KV cache—yet retains semantically typed, directly queryable evidence. Over a one-hour stream at 1 FPS, the entire dictionary amounts to 4.0 MB, whereas visual KV memory would require over 118 GB. Even the most compact visual baseline, prototype memory with K=32 slots, achieves its small footprint only by collapsing the stream into a fixed-size state that discards temporal indexing and fine-grained cues. In contrast, our dictionary scales linearly but remains practically negligible, keeping every observed moment individually addressable while adding virtually no system-level memory burden.

Table 10: **Accuracy–throughput trade-off of the captioning stage.** Captioning throughput is measured in frames per second; all other components are identical to our default setting with Qwen3-VL-8B as the answer model. OVO-Bench reports the three category averages and the overall score; SB denotes StreamingBench overall.

Captioner	Frames/call	Throughput	RT	BW	FWD	OVO	SB
No memory (baseline)	–	–	63.24	43.44	48.42	51.70	59.33
Qwen3-VL-2B-FP8 (Ours)	1	1.65	**73.28**	**51.37**	49.41	**58.02**	**64.51**
Qwen3.5-0.8B	1	3.52	72.46	50.21	**50.94**	57.87	62.92
Qwen3-VL-2B-FP8	8	**8.97**	69.31	48.97	50.91	56.40	61.55

Table 11: **Accuracy–efficiency comparison with InfiniPot-V [43].** Both methods use the same backbone (Qwen2.5-VL-7B), environment, and evaluation protocol, and InfiniPot-V is run with its official implementation. Memory budget counts the answer-time tokens provided to the answer model. †InfiniPot-V reuses its compressed KV states built during streaming, so its answer latency mainly consists of question processing and answer generation, whereas TimeTraveler prefills its visual and textual evidence at query time.

Method	Visual memory	Text memory	Memory budget	Memory footprint	Answer latency	Answer-model memory	OVO-BW	SB-RT
InfiniPot-V	4K compressed KV	–	3.83K	220 MB	**53.75 ms**†	16.80 GB	45.60	76.76
TimeTraveler	Current frame	$K^\star \approx 2.5$	**0.70K**	**39 MB**	58.79 ms	**16.62 GB**	**57.14**	69.96
	Current frame	Top-20	∼3.8K	214 MB	72.02 ms	16.79 GB	56.37	70.88
	Uniformly selected frames	$K^\star \approx 2.5$	∼3.8K	214 MB	91.81 ms	16.79 GB	53.39	**77.72**

A.2 Scalability beyond 1 fps.

We follow the 1-fps setting of prior streaming methods [17, 22, 23, 25]. Captioning dominates stream processing (605.82 ms per frame), and the dictionary update leaves about 362 ms of headroom per second at 1 fps; since state updates run asynchronously, they do not block answer generation. For higher input rates, two complementary axes extend the operating range on the same eight-field schema (Table 10): clip-level dictionary captioning, which writes one structured caption per clip of consecutive frames and thus reduces the captioning frequency by the clip length, and a lighter captioner (Qwen3.5-0.8B) under the identical prompt and schema. The 0.8B captioner raises captioning throughput from 1.65 to 3.52 frames/s while staying within 0.15 points of our default overall OVO-Bench score, and 8-frame clips sustain 8.97 frames/s while retaining gains of 4.70 (OVO-Bench) and 2.22 (StreamingBench) points over the backbone. The two axes can be combined: increasing the clip size from 4 to 32 frames raises the sustainable input rate from about 5 to 33 fps, and the 0.8B captioner with 8-frame clips reaches about 22 fps.

A.3 Comparison with memory-efficient streaming.

We compare with InfiniPot-V [43], which bounds the visual KV cache, using the same backbone (Qwen2.5-VL-7B), environment, and protocol (Table 11). Under a comparable answer-time budget of about 4K tokens, TimeTraveler improves OVO Backward accuracy over InfiniPot-V by 7.79 points with uniformly selected frames and by 10.77 points when the budget is spent on retrieved captions, and the one-frame setting reaches 57.14 with only 0.70K tokens and a 39 MB footprint, suggesting that structured textual evidence is an effective representation for recovering historical evidence under a constrained budget. On StreamingBench Real-Time, the matched-budget setting with uniformly selected frames is competitive (77.72 vs. 76.76), whereas the text-heavy settings are lower (69.96–70.88), reflecting that real-time perception benefits more from visual frames than from historical text. InfiniPot-V attains lower answer latency because it reuses compressed KV states at query time, and our captioner/planner and embedder require about 4.66 GB of additional resident GPU memory during streaming. Since InfiniPot-V bounds the visual KV cache while TimeTraveler adds lightweight structured text entries and retrieves a small query-relevant subset, the two approaches manage different parts of streaming memory and are potentially complementary.

Table 12: **Same-backbone comparison on OVO-Bench with Qwen2.5-VL-7B as the answer model.** *Results reported in the original papers, which do not report backbone-only OVO-Bench scores; the backbone-only and TimeTraveler rows are evaluated in our environment with the captioner, planner, embedder, and all hyperparameters unchanged.

Method	Real-Time	Backward	Forward	Overall
QueryStream-7B* [22]	61.40	42.10	39.03	47.51
StreamAgent-7B* [24]	61.30	41.70	**45.40**	49.40
Qwen2.5-VL-7B [1]	63.46	44.21	39.40	49.02
+ TimeTraveler	**71.71**	**48.96**	36.90	**52.52**

A.4 Retrieval horizon and index scaling.

The 600-s horizon applies only to the retrieval keys (embeddings); caption values are lightweight (0.78 KB per caption; Table 9) and are kept permanently, so no evidence is discarded. Because keys accumulate at every frame and the end of a stream is unknown, we bound the key index by a fixed horizon. Extending the indexed horizon from 600 s to one hour changes the total per-query retrieval latency from 13.79 to 13.80 ms, while the total storage for a one-hour stream grows from 4.0 MB (Table 9) to about 10.1 MB; query encoding alone takes 13.76 ms, about 99.7% of the one-hour retrieval latency. Even with a four-hour index of 14,400 frame entries, cosine search takes only 0.083 ms and the complete store occupies about 40.49 MB. In a diagnostic run with a longer indexed horizon, evidence older than 600 s was selected in only 3 of the 1,473 past-routed queries (0.2%), so the cap was rarely binding in our setting. All results in this paper use the 600-s horizon; the 30-s lookahead $T_{\max}$ can likewise be extended when longer future observation is required.

A.5 Same-backbone comparison.

The compared streaming methods use different answer backbones: Dispider-7B builds on Qwen2-7B, Flash-VStream-7B on Vicuna-7B, and QueryStream-7B and StreamAgent-7B on Qwen2.5-VL-7B. Table 12 therefore evaluates TimeTraveler with Qwen2.5-VL-7B, the answer backbone of QueryStream and StreamAgent, while keeping the captioner, planner, embedder, and hyperparameters unchanged. TimeTraveler improves the backbone from 49.02 to 52.52 overall and exceeds QueryStream and StreamAgent by 5.01 and 3.12 points overall and by 10.31 and 10.41 points on Real-Time, indicating that the gains stem from the framework itself. Forward accuracy is lower with this backbone (39.40→36.90), in line with the backbone-dependent Forward behavior in Appendix A.6.

A.6 Per-category analysis.

With Qwen3-VL, the largest gains appear in categories aligned with the intended role of each module: detail-sensitive categories supported by the structured `Text` and `Objects` fields, such as OCR (+12.08) and OJR (+17.39) on OVO-Bench and TR (+19.27) on StreamingBench (cf. Table 5), and temporally grounded categories supported by routing and gating, such as EPM (+6.73) and HLD (+15.05) on OVO-Bench and PO (+10.50) on StreamingBench (cf. Tables 4 and 13). A few categories decrease. On OVO Forward, REC changes by -0.86 with Qwen3-VL and -10.45 with InternVL3.5, and SSR by -9.86 with InternVL3.5; since the deferred-observation gate is not used under the OVO multiple-triggering protocol (Appendix C.1), these changes likely reflect how each answer backbone uses retrieved evidence for counting and step recognition. On StreamingBench Omni-Source, the audio-dependent MA decreases (55.20→49.20 with Qwen3-VL) because neither the Time Dictionary nor the answer model receives audio in our visual-only setting, whereas visually inferable categories such as ER improve with both backbones (41.20→43.20 and 40.40→48.40).

A.7 Details of the planner and evidence-coverage evaluation.

Temporal routing. We sampled 150 StreamingBench queries, 50 each from Real-Time Visual Understanding, Contextual Understanding, and Proactive Output; Proactive Output was included to cover future-directed queries, which OVO Forward does not provide under its multiple-triggering protocol. A human annotator assigned each query to current, past, or future using the question,

options, ground-truth answer, query-time frame, and the corresponding caption, while remaining blind to the planner output. GPT-4o [48] and GPT-5 were used as additional reference judges under the same routing criteria but without access to the ground-truth answer. To measure the downstream effect of routing errors, we replaced the planner predictions with the human-assigned routes on this subset.

Ambiguous and mixed-temporal queries. The planner outputs a single strategy, but the strategies are not mutually exclusive evidence sources: since the answer model always receives the query-time visual input v, the past route augments rather than replaces the current evidence, so queries that need both past and current evidence can be served by routing to past. Most remaining disagreements arise from such ambiguous past–current cases. Future intent is comparatively explicit in Proactive Output, which expresses it through trigger instructions; implicit future intent and queries that require combining past, current, and future evidence are not explicitly modeled by the single-route planner (Appendix D).

Evidence coverage. We randomly sampled 100 past-routed OVO Backward queries, including both correctly and incorrectly answered ones. For each query, the annotator inspected the question, options, ground-truth answer, the selected $K^\star$ captions, and their source frames, and assigned one of four outcomes: the evidence is contained in a selected caption (48%); it is visible in a selected source frame but missing from its caption (10%); it is absent from the selected captions but found elsewhere in the Time Dictionary (17%); or it is not found in any dictionary caption (25%).

A.8 Random routing variants.

Random Routing replaces the Temporal Strategy Planner by assigning each query a temporal strategy uniformly at random and then applies the evidence acquisition of the selected strategy as usual.

Random Routing. Each query is assigned to the past, current, or future strategy, matching the action space of the planner; for the future strategy, the gate stops once sufficient evidence is available to answer the question. Over five runs, it achieves EPM 52.12$\pm$1.77, HLD 19.78$\pm$0.45, and ASI 63.24$\pm$1.13 on OVO-Bench Backward Tracing (mean values in Table 4), compared with 59.26, 25.27, and 69.59 for our routing; averaged over the three tasks, the scores are 45.05$\pm$0.63 and 51.37, respectively.

Random Routing over current and past. Each query is assigned to the current or past strategy only. When past is selected, the standard lookback is applied; when current is selected, the answer model receives the structured query-time caption. A single run achieves EPM 57.94, HLD 25.13, and ASI 66.52, and over five runs its task-averaged score is 50.01$\pm$0.70, compared with 51.37 for our routing under the same averaging.

Together, these results indicate that the gain of our routing stems from query-conditioned strategy selection rather than from mixing strategies.

A.9 Effect of gating on proactive output task.

Table 13 examines the effect of future gating on StreamingBench Proactive Output. In this task, a question is posed before the answer-relevant event, and the model is evaluated at one second after the ground-truth moment on whether it correctly decides to answer and, if so, whether the answer is correct. The baseline makes this decision using only the current visual frames, while our settings additionally provide the Time Dictionary memory. Without gating, the 8B answer model itself judges whether to respond, but this actually hurts Qwen3-VL (52.40$\rightarrow$45.20), suggesting that the model struggles to make reliable respond-or-wait decisions when given the full memory context. With gating, a lightweight 2B model sees only the current-frame caption and the question to make the respond-or-wait decision, then hands off to the 8B model for answering. This yields substantial gains on both backbones (Qwen3-VL: +10.50; InternVL3.5: +13.23), confirming that separating the gating decision from the answering model is more effective.

A.10 Routing-based visual budget allocation.

Table 7 shows that Real-Time questions benefit most from a single current frame, Backward questions perform best when relying on the dictionary without frames, and Forward questions degrade as frames

Table 13: **Effect of the future gating on StreamingBench Proactive Output.**

Backbone	Method	Acc.
Qwen3-VL-8B	Baseline	52.40
	+ Time Dictionary w/o gating	45.20
	+ Time Dictionary w/ gating (Ours)	**62.90**
InternVL3.5-8B	Baseline	40.00
	+ Time Dictionary w/o gating	41.20
	+ Time Dictionary w/ gating (Ours)	**53.23**

Table 14: **Routing-based visual budget allocation on OVO-Bench.** Qwen3-VL-8B is used as the answer model, and the number of visual frames it receives is assigned per temporal strategy using the planner output.

Policy	RT	BW	FWD	Overall
Ours (512 frames for all queries)	73.28	51.37	**49.41**	58.02
Past→512, Current→1	**78.38**	51.51	46.29	**58.73**
Past→0, Current→1	76.34	**54.99**	38.92	56.75

are removed. Motivated by this pattern, we assign different visual budgets to the past and current strategies using the planner output. Since the planner does not distinguish whether a past-routed query originates from a Backward or a Forward question, we evaluate two implementable policies (Table 14). Assigning 512 frames to past-routed queries and one frame to current-routed queries improves the overall score from 58.02 to 58.73, suggesting that the planner already provides a useful signal for adaptive visual budget allocation, while leaving more sophisticated policies to future work.

A.11 Adaptive selection vs. fixed top-K.

Table 15 compares our online probabilistic lookback with a fixed top-K retrieval variant under the same routing policy. We set $K = 4$, which gives the fixed-retrieval baseline a slightly larger budget than the average number of entries selected by our online probabilistic method ($\bar{K}^\star$=2.68 over past-routed queries; Table 16). Replacing the adaptive cumulative-mass selection rule with this fixed top-K strategy reduces backward accuracy from 51.37 to 49.64. This result shows that the number of relevant past entries should not be treated as constant across queries: some questions are resolved by a small amount of concentrated evidence, whereas others require a broader set of supporting captions. Our probabilistic selection mechanism adapts to this variability, preserving compactness when confidence is concentrated while expanding the evidence set when the retrieval distribution is more diffuse.

A.12 Number of retrieved captions per question type.

Table 16 reports the number of captions $K^\star$ selected by past lookback for the 1,473 OVO-Bench query instances routed to past. Across the 12 question types, the mean $K^\star$ ranges from 1.43 to 3.21 and the median from 1 to 3. Overall, past retrieval selects 2.68 captions on average, corresponding to approximately 480 caption tokens under the accounting of Table 3, or about 0.18% of the answer model's 256K-token context window. Only 10 of the 1,473 instances (0.68%) have $K^\star > 10$, all in REC and CRR; the largest case ($K^\star = 19$) corresponds to about 3.4K caption tokens, or 1.3% of the context window. The trend follows the temporal demands of each category: Real-Time instances retrieve the fewest captions on average (2.19), Backward instances more (2.61), and Forward instances the most (2.82), consistent with sequential-step and clue-based questions that need evidence accumulated across several moments. The retrieved context therefore remains small for all question types.

A.13 Redundancy among retrieved captions.

Using temporal proximity as a proxy for local redundancy, 28.2% of adjacent selected-caption pairs lie within 1 s of each other and 49.4% within 2 s. To test whether these nearby captions are redundant,

Table 15: **Effect of adaptive past-evidence selection on OVO Backward Tracing.** Since this comparison isolates the past-lookback module, we report overall backward accuracy only. Results are obtained by applying our method on top of Qwen3-VL-8B[16]. We compare the proposed online probabilistic lookback with a fixed top-K variant under the same routing setup. Replacing adaptive selection with fixed top-K retrieval degrades performance.

Selection Strategy	**Backward Avg.**
Top-K Selection ($K = 4$)	49.64
Online Probabilistic Selection (Ours)	**51.37**

Table 16: **Number of captions selected by past lookback ($K^\star$) across OVO-Bench question types.** Statistics are computed over the 1,473 query instances routed to past.

	Real-Time Visual Perception						Backward Tracing			Forward Active Responding		
	OCR	ACR	ATR	STU	FPD	OJR	EPM	ASI	HLD	REC	SSR	CRR
Mean $K^\star$	2.09	2.27	2.24	2.20	1.43	2.61	2.54	2.80	2.52	2.77	3.21	2.68
Median $K^\star$	2	2	2	2	1	2	2	2	2	2	3	2
Max $K^\star$	7	6	6	7	2	5	7	10	9	13	10	19

we insert a greedy temporal-deduplication step after causal filtering and before top-N truncation: candidates are processed in descending retrieval-score order, and a candidate is removed if it lies within 1 s or 2 s of a higher-scoring retained candidate, with all other components unchanged. On OVO Backward Tracing, a 1-s window reduces the mean $K^\star$ by 17% and the text budget by 15% without degrading accuracy (task-averaged $+0.12$, instance-averaged unchanged), whereas a 2-s window reduces them by 29% and 27% but lowers accuracy (task-averaged -0.44, instance-averaged -0.63). Local overlap is therefore present, but temporally nearby captions are not always redundant. The degree of redundancy likely depends on video content, such as the presence of shot boundaries and the balance between static dialog scenes and action.

A.14 Sensitivity to temperature T_{past}.

Table 17 examines the temperature T_{past} used in the recency-aware softmax for past-evidence selection. Since T_{past} is a hyperparameter of the past-lookback module, Table 17 reports overall backward accuracy only. Results are obtained by applying our method on top of Qwen3-VL. The best backward-tracing accuracy is obtained at $T_{\text{past}} = 0.02$, which reaches 51.37. When the temperature is reduced to 0.01, performance drops to 50.39, suggesting that overly sharp distributions can over-commit to a very small set of candidates and become brittle to retrieval noise. Increasing the temperature to 0.05 or 0.10 also hurts performance, indicating that excessively smooth selection weakens the model's ability to focus on the most relevant historical evidence. These results support our choice of a moderately sharp temperature that balances selectivity and robustness. We further repeat the sweep with InternVL3.5 (Table 20). The Backward average varies by at most 1.21 points with Qwen3-VL and 1.14 points with InternVL3.5, and every setting improves over the corresponding no-memory baseline (by at least 6.72 and 2.18 points, respectively). The best value for InternVL3.5 is 0.05, which exceeds the adopted value by 0.85 points, indicating that the adopted setting transfers across backbones.

A.15 Sensitivity to nucleus threshold t_p.

Table 18 studies the nucleus threshold t_p, which controls how much cumulative probability mass is retained in the adaptive evidence prefix. Since t_p is a hyperparameter of the past-lookback module, Table 18 reports overall backward accuracy only. Results are obtained by applying our method on top of Qwen3-VL. The strongest result again appears at $t_p = 0.70$, yielding 51.37 backward accuracy. Smaller thresholds such as 0.10 or 0.30 underperform, which suggests that overly aggressive truncation discards useful complementary context. On the other hand, very large thresholds such as 0.90 or 1.00 also degrade performance, implying that including too many lower-confidence entries introduces distracting historical evidence. This trend confirms that adaptive selection is most effective when it retains a compact but sufficiently informative subset of past captions.

Table 17. **Sensitivity to temperature T_{past} on OVO Backward Tracing.**

T_{past}	**Backward Avg.**
0.01	50.39
0.02	**51.37**
0.05	50.34
0.10	50.16

Table 18. **Sensitivity to threshold t_p on OVO Backward Tracing.**

t_p	**Backward Avg.**
0.10	50.72
0.30	50.50
0.50	50.84
0.70	**51.37**
0.90	50.74
1.00	50.07

Table 19. **Sensitivity to recency scale τ_r on OVO Backward Tracing.**

τ_r	**Backward Avg.**
No recency bias	49.51
150	51.37
300	51.35
600	**51.51**
1200	50.87

Table 20: **Sensitivity to T_{past} across answer backbones on OVO-Bench Backward Tracing (average accuracy).** All other settings are fixed. The no-memory baseline is 43.44 for Qwen3-VL-8B and 32.25 for InternVL3.5-8B.

Answer model	0.01	**0.02 (Ours)**	0.05	0.10	Range
Qwen3-VL-8B	50.39	**51.37**	50.34	50.16	1.21
InternVL3.5-8B	34.43	34.72	**35.57**	34.47	1.14

A.16 Sensitivity to recency scale τ_r.

Table 19 evaluates the recency scale τ_r, which determines how strongly older evidence is penalized during lookback. Since τ_r is a hyperparameter of the past-lookback module, Table 19 reports overall backward accuracy only. Results are obtained by applying our method on top of Qwen3-VL. Moderate recency scales perform best: τ_r between 150 and 600 yields 51.35–51.51 backward accuracy, while removing recency bias entirely (implemented by dropping the second term in Eq. (3)) leads to a clear drop to 49.51. This gap shows that semantic similarity alone is insufficient for streaming QA, because temporally distant but semantically related captions can act as distractors. Performance declines again at 1200, where the temporal penalty becomes too weak. In particular, the competitive result at $\tau_r = 600$ suggests that retrieval quality remains strong even over roughly 10 minutes of context. Overall, the results validate recency-aware retrieval as an important ingredient for temporally grounded evidence acquisition.

A.17 Textual Input Sensitivity

To verify that the Time Dictionary provides substantive evidence rather than merely additional context, we conduct a noise injection experiment (Fig. 3). We replace a fraction ρ of the stored captions with irrelevant captions sampled from unrelated videos and evaluate on OVO-Bench. Performance degrades steadily as ρ increases, dropping from 58.02 at ρ=0 to 50.27 at ρ=1.0, which falls below the no-memory baseline (51.70). This shows that the answer model actively relies on the retrieved caption content to produce correct answers, rather than benefiting from the mere presence of additional text input. It also validates the quality of our Time Dictionary: the captions generated by the streaming captioner carry accurate, answer-relevant evidence that meaningfully contributes to question answering.

A.18 Qualitative Result of Lookback and Dictionary.

Fig. 4 illustrates a representative lookback retrieval case. The user asks which muscle groups are highlighted during a front raise exercise. The similarity score peaks at a past frame showing a 3D anatomy diagram with red-highlighted deltoids, and the corresponding Time Dictionary caption retains the precise on-screen text, object descriptions, and spatial relations needed to answer. This example demonstrates that our structured captions preserve the fine-grained evidence that compressed visual representations would likely discard, and that the retrieval mechanism successfully surfaces the most relevant past observation.

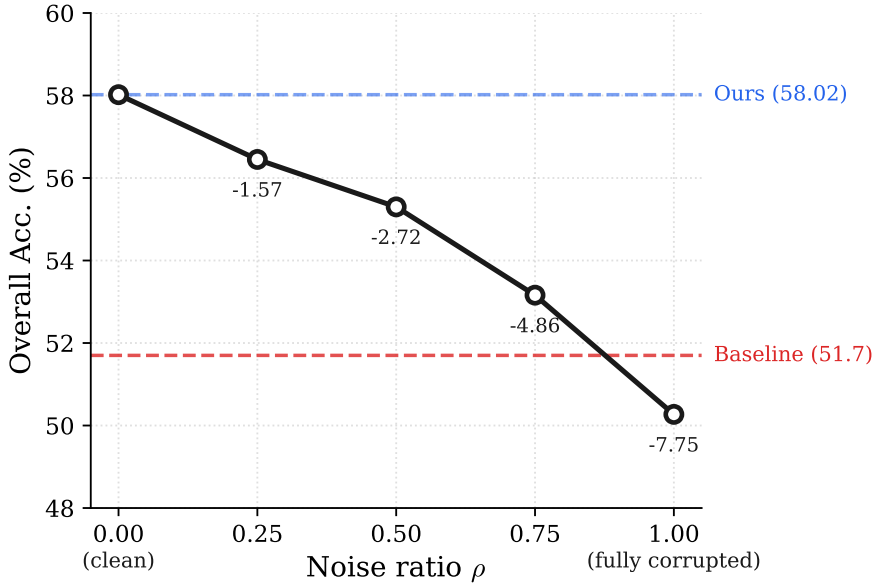


Figure 3: **Textual input sensitivity on OVO-Bench.** We replace a fraction ρ of the Time Dictionary captions with irrelevant captions sampled from unrelated videos and measure the resulting Overall accuracy. Dashed lines indicate the baseline (no memory) and our full method (clean captions).

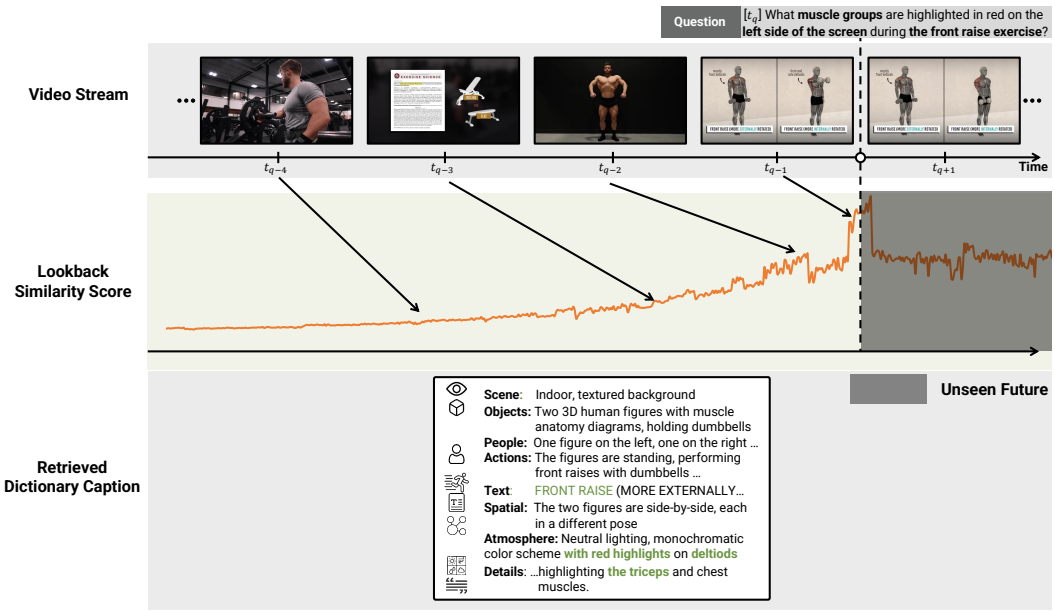


Figure 4: Qualitative example of lookback retrieval on OVO-Bench. Given a query at time t_q, the similarity score over past captions (orange curve) peaks at the frame most relevant to the question. The retrieved Time Dictionary caption preserves fine-grained details such as on-screen text (*"FRONT RAISE"*), spatial layout, and highlighted muscle groups, which are essential for answering correctly.

B Prompts in TimeTraveler

We document the three prompt templates used in our pipeline: the per-frame **captioner** that populates the Time Dictionary (§B.1), the **planner** that performs temporal routing (§B.2), and the **gating** model that decides cue-trigger emission for proactive-output questions (§B.3).

B.1 Captioner Prompt

Per-frame captions populating the Time Dictionary are produced by a lightweight VLM (8-bit-quantized Qwen3-VL-2B [16]) using the structured 8-field template below. The prompt enforces concision, non-redundancy across fields, and a fixed field order so that downstream schema-level operations (e.g., field-routed retrieval, schema-drop ablation) can address each field deterministically. Fields with no observable content are filled with the literal string "N/A".

Captioner Prompt

```
Analyze the image and provide a structured description using the format below.
Be concise. List key items only, do not repeat information across fields. Focus
on the most salient elements. If a field is not applicable, write 'N/A'.

Scene: [locational or environmental information, indoor/outdoor]
Objects: [key objects with attributes such as color, size, state]
People: [count, appearance, clothing, expression]
Actions: [what is happening, interactions between people/objects]
Text: [any visible text, signs, labels, subtitles, logos]
Spatial: [layout and relative positions of key elements]
Atmosphere: [lighting, color tone, time of day, weather, mood]
Details: [description sentences that are shown in the image in one coherent
paragraph. Write 2-3 sentences and keep it concise, around 80-120 words.]
```

Decoding configuration. Greedy decoding with temperature $= 0$, top-p $= 0.8$, top-k $= -1$, presence penalty $= 1.5$, repetition penalty $= 1.0$, and max new tokens $= 400$.

B.2 Planner Prompt

Each incoming question is routed into one of three temporal regions (`past`, `current`, or `future`) by a caption-augmented meta-classifier (8-bit-quantized Qwen3-VL-2B, text-only inference). In addition to the question, the planner receives the *current frame caption* – the most-recent caption from the Time Dictionary – so that routing decisions are grounded in present visual context. The decision selects the retrieval scope used downstream: specific past captions retrieved by query similarity (`past`), the most-recent frame caption from the temporal buffer (`current`), or a wait-and-emit gating loop (`future`).

Planner Prompt (curcap v11)

```
You route streaming video questions into one of three temporal regions based on
what the answer depends on.

• current: the answer is a property of the frame at the moment of asking. It
does not require looking back or aggregating earlier content.
• past: the answer requires information from earlier in the video. This
includes counting things that have occurred, summarizing prior activity,
comparing with an earlier moment, reasoning about causes or sequences, or
referring to a specific earlier event, object, or statement.
• future: the question sets up a trigger -- the system must wait and emit a
response only when a specified later condition occurs. Typical cues: ''when X
happens, output Y'', instructions to monitor and respond.

Present-tense phrasing alone does not imply current. If the correct answer needs
evidence from before this moment, it is past.

Current frame caption: ''{caption}''
Question: ''{question}''

Reply with exactly one word: current, past, or future.
```

Decoding configuration. Greedy decoding (temperature $= 0$, do_sample $=$ False), max new tokens $= 8$. The single-word output is parsed case-insensitively; malformed outputs default to `current`.

Caption substitution. The `{caption}` placeholder is filled with the 8-field structured caption (§B.1) of the frame nearest to the question's time stamp.

B.3 Gating Prompt

For proactive-output questions (the question sets up a future trigger of the form "when X happens, output Y"), a binary gating prompt is issued at each streaming step to decide whether the cue condition has been satisfied at the current moment. When the gate emits "yes", a follow-up cue-response call is issued on the same clip to extract the actual cue answer; the $(\text{time}, \text{answer})$ tuple is then scored by the evaluator against the ground-truth event time and the specified output string.

Source of `{specified_output}`. The placeholder `{specified_output}` holds the response text Y that the user instruction itself asks the system to emit ("…, output Y"). Following the official StreamingBench protocol [26], this text is parsed from the Proactive Output instruction and is part of the standard evaluation input, equally available to all evaluated methods. The ground-truth trigger timestamp is used only by the evaluator. At each step, the gate decides whether to respond or wait using only the question, the specified output, and captions obtained causally up to the current time.

Gating Prompt

```
You are a gating agent watching a streaming video at time t={t_now}s.
At t={t_q}s the user asked: ''{question}''
The expected output to deliver when the right moment arrives:
''{specified_output}''
Captions describing what is visible:
[t={t_now}s] {caption}
Based on the captions, has the moment to deliver the expected output arrived?
Answer with only ''yes'' or ''no'':
```

Decoding configuration. Greedy decoding (temperature $= 0$, do_sample $=$ False), max new tokens $= 32$. The gate output is parsed by case-insensitive substring containment of "yes". The cue-response output is scored by case-insensitive substring containment of `specified_output`.

Caption substitution. The `{caption}` placeholder is filled with the 8-field structured caption (§B.1) of the frame at the current polling time `t_now` (window size $W{=}1$).

C Additional Benchmark Explanation

C.1 OVO-Bench.

OVO-Bench [27] evaluates temporal awareness in online video understanding by posing questions at specific timestamps during a video stream, rather than after the entire video has been observed. The benchmark comprises 644 videos with 2,814 human-curated question-answer pairs across 12 tasks, organized into three temporal scenarios. *Real-Time Visual Perception* tests understanding of the current scene at query time, covering Optical Character Recognition (OCR), Attribute Recognition (ACR), Action Recognition (ATR), Spatial Understanding (STU), Future Prediction (FPD), and Object Recognition (OJR). *Backward Tracing* requires the model to recall past events, with tasks including Episodic Memory (EPM), Action Sequence Identification (ASI), and Hallucination Detection (HLD). *Forward Active Responding* presents questions whose answers depend on events that have not yet occurred; the model must decide to wait rather than respond prematurely. This scenario includes Repetition Event Count (REC), Sequential Steps Recognition (SSR), and Clues Reveal Responding (CRR). Backward Tracing and Real-Time Visual Perception adopt a multiple-choice format with accuracy as the metric. Forward Active Responding employs a multiple-triggering evaluation pipeline that densely queries models along the temporal axis, using both accuracy-based and score-based metrics. The overall score is the macro average across the three scenario-level averages.

Evaluation protocol for Forward Active Responding. OVO-Bench Forward Active Responding follows the official multiple-triggering protocol [27]: models are densely queried along the timeline, each round receives the clip from 0 s up to that polling time, and rounds are scored independently.

We follow this protocol identically for all methods, so the deferred-observation gate (Eq. (7)) does not intervene on OVO Forward, and the planner routes its 172 questions to past (104), current (67), and future (1). The future gate is instead evaluated on StreamingBench Proactive Output, where the system itself must decide when to respond.

C.2 StreamingBench.

StreamingBench [26] evaluates streaming comprehension by presenting questions at different temporal positions within a video, simulating real-time user interaction during playback. The benchmark contains 900 videos with 4,500 human-curated QA pairs across 18 tasks under three dimensions. *Real-Time Visual Understanding* (10 tasks) covers Object Perception (OP), Causal Reasoning (CR), Clips Summarization (CS), Attribute Perception (ATP), Event Understanding (EU), Text-Rich Understanding (TR), Prospective Reasoning (PR), Spatial Understanding (SU), Action Perception (ACP), and Counting (CT). *Omni-Source Understanding* (4 tasks) requires integrating audio and visual signals, including Emotion Recognition (ER), Scene Understanding (SCU), Source Discrimination (SD), and Multimodal Alignment (MA). *Contextual Understanding* (4 tasks) evaluates higher-level reasoning over accumulated context, including Anomaly Context Understanding (ACU), Misleading Context Understanding (MCU), Sequential Question Answering (SQA), and Proactive Output (PO). The PO task is notable in that the model must determine the appropriate moment to respond without an explicit user query; a response is correct only if the output timestamp falls within two seconds of the ground-truth timestamp. All tasks follow a multiple-choice format with accuracy as the primary metric.

D Limitations and Broader Impacts

Limitations and future work. The planner outputs a single temporal strategy per query and therefore does not explicitly model implicit future intent or queries that require combining past, current, and future evidence; confidence-aware or multi-route planning is a natural extension. The future gate responds at its first positive decision, so instructions that require accumulating repeated events over a fixed future interval (e.g., counting events during the next ten seconds) would require maintaining and updating an accumulated count until the horizon, a setting not covered by current benchmarks. Our system is visual-only, so audio-dependent tasks remain outside its scope. Broader validation under noisier real-world streams, multi-turn interactions, and longer future waiting horizons remains future work. Finally, since structured textual memory is complementary to visual memory methods such as KV-cache compression [43], combining the two is another promising direction.

Broader impacts. Our work improves streaming video question answering by preserving and retrieving temporal evidence from video streams. Potential negative impacts include privacy-sensitive deployment in surveillance, workplace monitoring, or bystander recording, where persistent textual memory may expose identities, locations, or sensitive activities. Incorrect captions or temporal misrouting may also cause false alarms or misleading summaries in safety-critical settings. We recommend deployment only with user notification, access control, retention limits for the Time Dictionary, and mechanisms for deleting stored temporal records.

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: The abstract and introduction state the scope of the proposed streaming video QA framework, including the Time Dictionary and Temporal Strategy Planner. The claims are supported by experiments on OVO-Bench and StreamingBench.

   Guidelines:

   - The answer [N/A] 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 [N/A] 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: The Limitation paragraph in Section 5 and Appendix D discuss limitations including dependence on caption quality, possible planner/gate errors, and the need for broader validation under noisy streams, multi-turn interactions, and longer future horizons.

   Guidelines:

   - The answer [N/A] 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: [N/A]

Justification: The paper does not present theoretical results or formal proofs.

Guidelines:

- The answer [N/A] 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: The paper specifies the benchmarks, backbone models, frame sampling rate, caption schema, retrieval hyperparameters, future-gating settings, prompts, and hardware. The appendix provides prompt templates and additional hyperparameter analyses.

Guidelines:

- The answer [N/A] 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: We included all the details in the appendix, and the code is available at `https://github.com/KHU-AMI/TimeTraveler`.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.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://neurips.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) necessary to understand the results?

   Answer: [Yes]

   Justification: Yes, we have baselines and benchmarks explanations in the manuscript.

   Guidelines:

   - The answer [N/A] 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: [N/A]

   Justification: Our experiments do not need statistical results.

   Guidelines:

   - The answer [N/A] 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 include the computational details.

   Guidelines:

   - The answer [N/A] 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: We reviewed CoE.

   Guidelines:

   - The answer [N/A] 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 add it in appendix: broader impact section.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] 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., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: This is not relevant with our work.

    Guidelines:

    - The answer [N/A] 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: [N/A]

    Justification: Our work do not include any license assets.

    Guidelines:

    - The answer [N/A] 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: [No]

    Justification: This is not relevant with our work.

    Guidelines:

    - The answer [N/A] 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: [N/A]

    Justification: This is not relevant with our work.

    Guidelines:

    - The answer [N/A] 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: [N/A]

    Justification: This is not relevant with our work.

    Guidelines:

    - The answer [N/A] 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 rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: We only use LLMs for grammar-level inspection.

Guidelines:

- The answer [N/A] 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 in the NeurIPS handbook for what should or should not be described.