MEMENTO: TOWARD AN ALL-DAY PROACTIVE ASSISTANT FOR ULTRA-LONG STREAMING VIDEO

Anonymous authors
Paper under double-blind review

ABSTRACT

Multimodal large language models have demonstrated impressive capabilities in visual-language understanding, particularly in offline video tasks. More recently, the emergence of online video modeling has introduced early forms of active interaction. However, existing models, typically limited to tens of minutes, are not yet capable of all-day proactive understanding over ultra-long video streams. They struggle to maintain long-term context online, as they suffer from token accumulation and lack scalable memory mechanisms. These limitations hinder critical tasks such as reminding users that medication was taken hours earlier—an ability that exemplifies the shift from reactive to memory-oriented assistants with long-term reasoning. To bridge this gap, we present Memento, the first proactive vision-language framework for ultra-long streaming video. To avoid token growth and support scalable long-duration understanding, we introduce Dynamic Memory and Query-related Memory Selection, enabling sparse memory retention and efficient retrieval. To address the training challenges of memory-based modeling, we propose Step-Aware Memory Attention, which aligns memory access with temporal steps for stable supervision. To support both training and evaluation of active, long-term behavior, we construct Memento-54K and MementoBench, a dataset-benchmark suite covering diverse tasks on text, object, and action across video streams up to 7 hours. Experiments demonstrate that Memento achieves superior performance, paving the way toward reliable all-day proactive video assistants.

1 INTRODUCTION

Recent advancements in large language models (LLMs) (Ouyang et al., 2022; Touvron et al., 2023; Yang et al., 2024b;a; Xin et al., 2025; Guo et al., 2025) and vision-language models (VLMs) (Liu et al., 2023; 2024a; Achiam et al., 2023; Bai et al., 2025) have shown remarkable progress in video understanding, particularly with the emergence of long-form (Ren et al., 2024; Song et al., 2024a; Zeng et al., 2025) and online video LLMs (Chen et al., 2024a; Wu et al., 2024b; Li et al., 2025a; Qian et al., 2025). Such progress has further raised expectations for an all-day, proactive assistant. This assistant would continuously perceive the environment through ultra-long video streams and proactively interact with humans, rather than merely responding passively to explicit user queries. Achieving this capability would not only fundamentally transform the role of AI assistants in daily human activities, but also represent a critical step toward genuine autonomous agents (Fan et al., 2024; Wang et al., 2024b; Putta et al., 2024; Hong et al., 2024).

Despite this promising progress, existing models still fall short of realizing such a proactive assistant in practice. Their limitations become especially evident in scenarios requiring extremely long-term behavioral monitoring and temporal reasoning. For instance, an all-day assistant should be able to recall whether the user has already taken a specific medication hours ago, detect that the same object has been accessed multiple times throughout the day, or notice a warning text previously ignored. Fig. 1 illustrates a detailed case, inspired by a scene from the film *Memento*: the wife asks for an insulin shot three times within a few hours, but the husband, due to short-term memory loss, fails to recognize the repeated requests, potentially leading to serious consequences. In this scenario, existing long-form video models fail to assist during critical moments. They cannot issue timely warnings during the shots and fail to respond accurately. On the other hand, even the most advanced online streaming models struggle with ultra-long durations due to their token-based architectures, which cause visual tokens from each frame to accumulate in memory usage over time. As a result, after

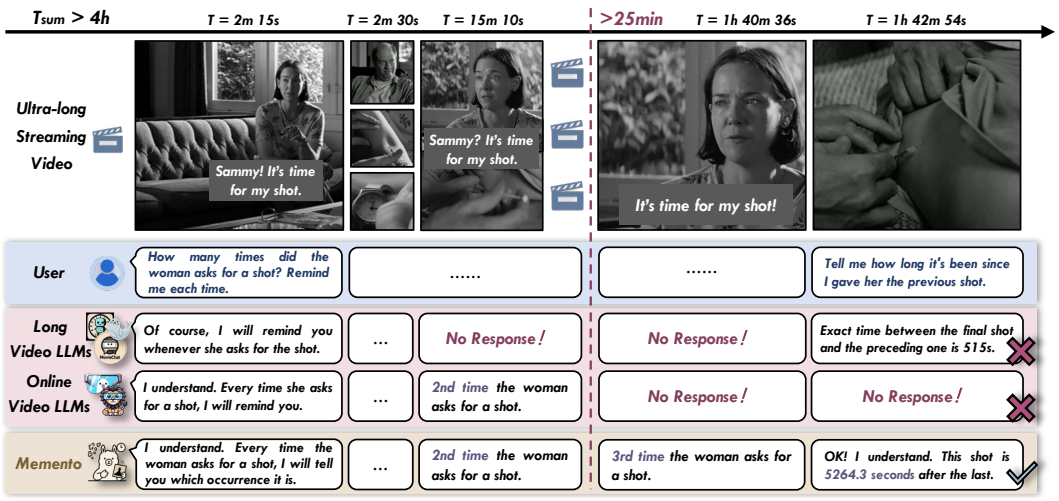


Figure 1: **Comparison of model behaviors for all-day proactive assistance.** Long-term and online video models both fail to assistant at injection points beyond 25 minutes. Conversely, Memento continuously tracks repeated shots, demonstrating its capability toward serving as an all-day proactive assistant. Results for online models and Memento are obtained via supervised fine-tuning (SFT), while long video model outputs are based on prompt engineering due to architectural limitation.

at most a few dozen minutes, the model exceeds GPU memory limits, and cannot recognize that previous requests occurred.

To address the above issues, we propose **Memento**, a proactive vision-language framework for ultra-long video streams. To handle the long-term memory challenge, we introduce a Dynamic Memory (DM) mechanism that learns to retain or fuse incoming visual information over time, allowing Memento to preserve relevant context while keeping memory usage bounded. In addition, we propose a Query-related Memory Selection (QMS) module that retrieves only the most relevant memory slots during generation, enabling efficient and targeted access across extended video durations. This framework departs from the token-based paradigm, in which frame-level features are concatenated and multiple positions are supervised jointly. In contrast, Memento operates over dynamically updated memory representations, which evolve over time and cannot be aligned to discrete frame steps. As a result, directly applying token-level supervision leads to misaligned inputs and invalid training. To resolve this structural mismatch, we introduce Step-Aware Memory Attention (SAMA), which restricts attention to memory available at each step, ensuring temporally consistent and semantically valid learning. While Memento addresses the architectural challenges of proactive interaction with long-range memory, existing datasets (Chen et al., 2024a; Yao et al., 2025; Grauman et al., 2022; Yang et al., 2025) offer limited support for training or evaluation. Online benchmarks (Chen et al., 2024a; Li et al., 2025b; Wu et al., 2024a) include only short-term proactive tasks such as behavior recap based on recent frames, lacking supervision for long-term monitoring. To bridge this gap, we construct **Memento-54k** and **MementoBench**, covering diverse task types on text, object, and action over video streams up to 7 hours, all requiring long-range, proactive understanding.

Our contributions are summarized as follows:

- **Framework.** For the first time, a framework for proactive interaction over ultra-long video streams, named Memento, is proposed.
- **Memory modeling.** To address the scalable long-term memory challenges, we introduce dynamic memory and a query-related selection for selective retention and efficient retrieval.
- **Training strategy.** To enable training compatibility with dynamic memory, we propose step-aware memory attention, ensuring stable and effective learning for proactive vision-language modeling.
- **Dataset and benchmark.** We construct Memento-54k and MementoBench, covering diverse long-range proactive tasks, validating the effectiveness of Memento and supporting the development of an all-day proactive assistant.

Related Work	*Visual Input*	*Long Form*	*Proactive*
LLaMA-VID (ECCV 2024)	*fixed token*	✓	✗
TimeSuite (ICLR 2025)	*fixed token*	✓	✗
MovieChat (CVPR 2024)	*fixed memory*	✓	✗
MA-LMM (CVPR 2024)	*fixed memory*	✓	✗
VideoLLM-online (CVPR 2024)	*fixed token*	✗	✓
VideoLLM-MoD (NeurIPS 2024)	*dynamic token*	✗	✓
LION-FS (CVPR 2025)	*dynamic token*	✗	✓
Memento	***dynamic memory***	✓	✓

Table 1: **Comparison between related methods and the proposed Memento.** "Proactive" indicates whether the model supports interaction without explicit queries.

2 Related Work

Long-Form Video Understanding. Recent multimodal large language models have demonstrated strong instruction-following capabilities in video understanding (Cheng et al., 2024; Zhang et al., 2023; Li et al., 2024b; Liu et al., 2024b; Wang et al., 2025; Zhang et al., 2024b), particularly for long-range content. As early approaches based on sparse frame sampling often fail to capture key clues in long videos (Lin et al., 2024; Li et al., 2024a; Maaz et al., 2024; Ma et al., 2024; Zhou et al., 2024), fixed token-based methods have been introduced via encoding each frame into a fixed number of visual tokens, with compression algorithm for acquiring more frames (Wang et al., 2024c; Ren et al., 2024; Weng et al., 2024). For example, LLaMA-VID (Li et al., 2024c) represents each frame only using two visual tokens, enabling efficient processing of hour-long videos. Beyond token-based compression, fixed memory-based models, including MovieChat (Song et al., 2024b), Koala (Tan et al., 2024), MA-LMM (He et al., 2024), and others (Fan et al., 2024; Wang et al., 2024b; Zhang et al., 2024a), maintain a fixed-length memory bank as the visual tokens. They achieve effective long-video compression by aggregating redundant frames with similar features. However, these approaches suffer from increasing inference overhead, limited long-term memory and the inability to proactively interact, making them unsuitable for all-day assistant scenarios.

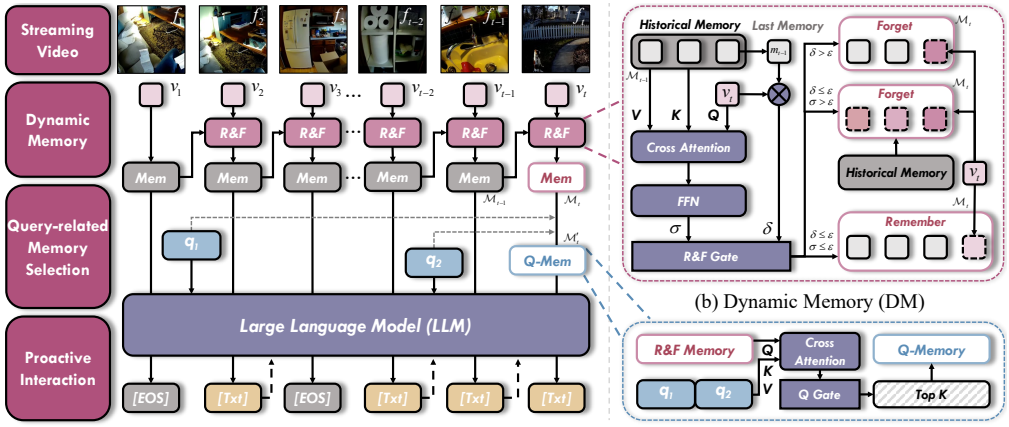
Online Video LLMs. Online Video LLMs aim to achieve real-time, proactive interaction over streaming inputs, with the ultimate ambition of supporting continuous operation across ultra-long video streams in open-ended scenarios. VideoLLM-online (Chen et al., 2024a) is the first to enable proactive interaction in video-language modeling by introducing a Streaming-EOS objective to decide when to respond or remain silent. However, like other fixed token-based approaches, it requires extracting visual tokens for each incoming video frame, leading to unacceptable growth in memory usage and computational cost. To reduce overhead, subsequent models introduce dynamic token strategies, such as MoE-style (Jacobs et al., 1991; Fedus et al., 2022; Shazeer et al., 2017; Lepikhin et al., 2021) token routing in VideoLLM-MoD (Wu et al., 2024b) and LION-FS (Li et al., 2025a), where only a subset of tokens are forwarded into deeper layers, and patch-level token dropping in TimeChat-online (Yao et al., 2025), where high redundancy regions are discarded. These methods increase the supported video duration to tens of minutes, but still retain frame token accumulation. Even the most advanced multimodal models (Wang et al., 2024a; Chen et al., 2024c; Gao et al., 2024; Chen et al., 2024b), such as GPT-4o (Achiam et al., 2023) and Gemini 1.5 Pro (Team et al., 2024; 2023), struggle to proactively reason over ultra-long streaming video. Unlike prior works, Memento introduces a dynamic memory design and query-related retrieval, as shown in Table. 1, avoiding token burden and preserving relevant information beyond fixed memory limits. Overall, it paves the way toward reliable, all-day proactive assistants..

3 Memento: A Proactive LLM over Ultra-Long Video Streams

3.1 Overview

In this section, we introduce our Memento in detail. As shown in Fig. 2 (a), given a streaming video $\mathcal{V} = \{f_1, f_2, \ldots, f_T\}$, Memento encodes each frame f_t using a ViT-based (Radford et al., 2021) encoder. The result $v_t \in \mathbb{R}^{(1+h_p \times w_p) \times C}$ contains a global [CLS] token and $h_p \times w_p$ spatial tokens.

Instead of directly projecting v_t into the language space via an MLP projector as in LLaVA (Liu et al., 2023; 2024a), we first process it through the Dynamic Memory (DM). At each step t, the current v_t and historical memory $\mathcal{M}_{t-1}$ are fused according to a Remember-and-Forget (R&F) strategy. It decides whether to retain the original information, and produces the updated memory



(a) Memento Framework (b) Dynamic Memory (DM) (c) Query-related Memory Selection (QMS)

Figure 2: **Overall architecture of Memento.** (a) Memento receives user queries and historical responses with the current memory state, achieving proactive interaction over ultra-long video streams. (b) Details of the DM mechanism, which mainly utilizes similarity-based retention and aggregation. (c) Details of the QMS module using query-conditioned gating and masking.

$\mathcal{M}_t = \text{DM}(v_t, \mathcal{M}_{t-1})$ as *R&F Memory*. Then, for all the user queries $q = \{q_1, q_2, \ldots, q_n\}$ in the past, the current R&F memory $\mathcal{M}_t$ is filtered by the Query-related Memory Selection (QMS) to retrieve the most relevant subset $\mathcal{M}'_t \subset \mathcal{M}_t$. The selected memory $\mathcal{M}'_t$ is fed into the LLM to generate the next-token distributions P, enabling both reactive and proactive responses.

Finally, considering that the fused memory changes across frames but lacks per-frame token structure, we apply Step-Aware Memory Attention (SAMA) to restrict attention to available memory at each time step during training. Thanks to this alignment, the supervision objective from VideoLLM-online (Chen et al., 2024a) can be directly adopted to train the memory-based framework:

$$\mathcal{L} = \frac{1}{N}\sum_{j=1}^{N}(\underbrace{-\log l_{j+1} P_j^{[\text{Txt}_{j+1}]}}_{LM\ Loss} - \underbrace{\log f_j P_j^{[\text{EOS}]}}_{Streaming\ Loss}), \tag{1}$$

where l_j is 1 if the j-th token is a language response token, and 0 otherwise. f_j is 1 if both (1) the j-th token is the last token in $\mathcal{M}'_t$, and (2) l_{j+1} is 0. $P_j^{[\text{Txt}_{j+1}]}$ is the probability on the $j+1$-th text token, output from the large language model head of the j-th token, and $P_j^{[\text{EOS}]}$ is the probability for the EOS token.

3.2 Dynamic Memory

To update R&F memory $\mathcal{M}_t$, we aim to balance between retaining essential information and fusing redundant content, which may arise in the short term (adjacent frames with little change) or in the long term (repeated scenes or actions), as shown in Fig. 2 (b). To handle both, we compute two relevance scores: (1) a short-term score δ, based on cosine similarity (Wang et al., 2024d;e) between the current frame v_t and the last memory $m_{t-1} \in \mathbb{R}^{(1+h_p \times w_p) \times C}$ in $\mathcal{M}_{t-1}$; and (2) a long-term score σ, obtained via cross-attention (Vaswani et al., 2017) between v_t and all flattened historical memory tokens $M_{t-1} \in \mathbb{R}^{N_{t-1}(1+h_p \times w_p) \times C}$ in Eq. 2. A fixed threshold ϵ controls memory update.

$$\begin{aligned} \text{Attn}(v_t, \mathbf{M}_{t-1}) &= \text{softmax}\left(\frac{(v_t W_q)(M_{t-1} W_k)^\top}{\sqrt{d}}\right), \\ \sigma &= \psi\left((\text{Attn}(v_t, M_{t-1}) \cdot (M_{t-1} W_v))\, W_o\right), \end{aligned} \tag{2}$$

where W_q, W_k, W_v, and W_o are projection matrices. $\psi(\cdot)$ denotes a summation followed by a sigmoid activation (LeCun et al., 1998), yielding a scalar score $\sigma \in \mathbb{R}$.

The R&F gate selects the memory update strategy based on a relevance threshold ϵ. If $\delta > \epsilon$, the current frame is considered locally redundant and is fused into the last memory token using Eq. 3.

$$\begin{aligned} &\text{score} = \text{softmax}(\text{Attn}(m_{t-1}, v_t)), \quad w = \text{score} \cdot u, \\ &\tilde{m}_{t-1} = m_{t-1} \cdot (1 - \text{sum}(w)) + w^\top v_t, \quad \mathcal{M}_t = \text{Concat}(\mathcal{M}_{t-1}^{[:N_{t-1}-1]}, \tilde{m}_{t-1}), \end{aligned} \tag{3}$$

where score $\in \mathbb{R}^{(1+h_p \times w_p)\times(1+h_p \times w_p)}$ is the normalized attention weight in spatial, $u \in \mathbb{R}$ is a fixed scalar update ratio. $\tilde{m}_{t-1}$ is the fused token, and $\mathcal{M}_{t-1}^{[:N_{t-1}-1]}$ denotes the first $N_{t-1}-1$ memory. If $\delta \leq \epsilon$ while $\sigma > \epsilon$, the frame is semantically aligned with long-term memory content; we thus reuse the same update strategy but compute attention scores by treating M_{t-1} as queries and v_t as keys and values, enabling soft updates across all memory slots. Finally, if both $\delta \leq \epsilon$ and $\sigma \leq \epsilon$, the frame is considered distinct and directly appended to memory, namely $\mathcal{M}_t = \text{Concat}(\mathcal{M}_{t-1}, v_t)$.

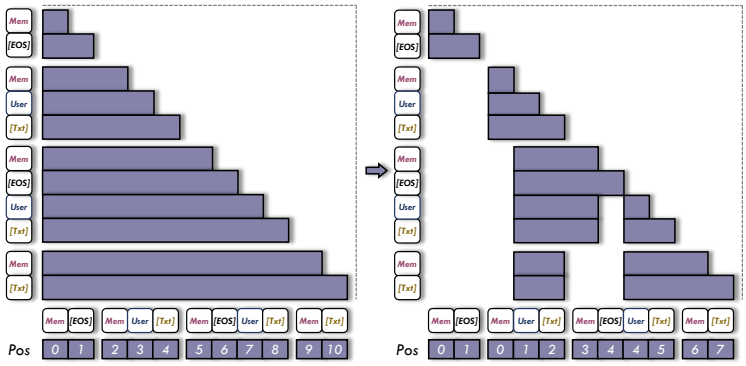
This gated update mechanism enables Memento to forget redundant content via token fusion, and remember distinct information. Different from token-based methods, this mechanism could avoid unacceptable growth in memory usage and computational cost. Compared with the fixed-length memory banks, it dynamically expands for novel content. This design maintains a compact yet expressive representation across ultra-long video streams.

3.3 Query-related Memory Selection

To reduce memory consumption while preserving response quality, we filter the current R&F memory $\mathcal{M}_t$ according to user queries q in Fig. 2 (c). Specifically, we transform $\mathcal{M}_t$ into $M_t \in \mathbb{R}^{N_t \times (1+h_p \times w_p) \times C}$, and compute cross-attention with user tokens Q as keys and values, following Eq. 2, to yield the score $R \in \mathbb{R}^{N_t}$ for each memory frame. QMS then applies a top-k gating strategy to select the most relevant $k = r_{\text{qms}} \cdot N_t$ tokens, $\mathcal{M}'_t = \text{TopK}(M_t, R, k)$. The selected compact memory $\mathcal{M}'_t$ is then passed to the LLM for generation. Our QMS ensures query-aware generation while decreasing the cost of full-memory attention, thereby enabling scalable reasoning over ultra-long temporal sequences.

3.4 Step-Aware Memory Attention

Unlike token-based models with frame-wise accumulation, the memory bank lacks explicit alignment with video steps. Thus, prior standard training methods in (Chen et al., 2024a; Wu et al., 2024b; Li et al., 2025a) with causal attention are inapplicable. As shown in Fig. 3 (a), this attention will allow access to expired memory. In contrast, our proposed SAMA in Fig. 3 (b) introduces a masking scheme to align with frame-wise visibility.



(a) Causal Attention (b) Step-Aware Memory Attention (SAMA)

Figure 3: **Causal attention vs. SAMA.** Causal attention (left) permits access to all past tokens, including expired memory. SAMA (right) restricts attention to valid memory, excluding irrelevant tokens.

Specifically, an example of input sequence is:

$$\texttt{tokens} = [\mathcal{M}'_1,\ \texttt{[EOS]},\ \mathcal{M}'_2,\ q_1,\ \texttt{[Txt]}_1,\ \mathcal{M}'_3,\ \texttt{[EOS]},\ q_2,\ \texttt{[Txt]}_2,\ \mathcal{M}'_4,\ \texttt{[Txt]}_3]_L. \quad (4)$$

A binary attention mask $A \in \{0,1\}^{L \times L}$ is built, where token x_i is allowed to attend to token x_j if:

$$A_{ij} = \begin{cases} 1, & x_j \in \mathcal{M}'_s \cup q \cup \{\texttt{[Txt]}_k\}_{k=1,2,..},\ i \geq j,\ x_j \neq \texttt{[EOS]} \\ 1, & i = j,\ x_i = \texttt{[EOS]} \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

Here, $s = \text{step}(x_i)$ denotes the video frame index when token x_i is added to the sequence. Furthermore, we reassign correct position ids for each token to ensure that tokens within the same frame share a base offset. This aligns positional encoding with the token visibility defined by the mask.

During inference, we maintain the same masking structure so that only previous dialog tokens are stored as key-value cache (Dao et al., 2023; Ge et al., 2024), allowing efficient streaming decoding with minimal computation.

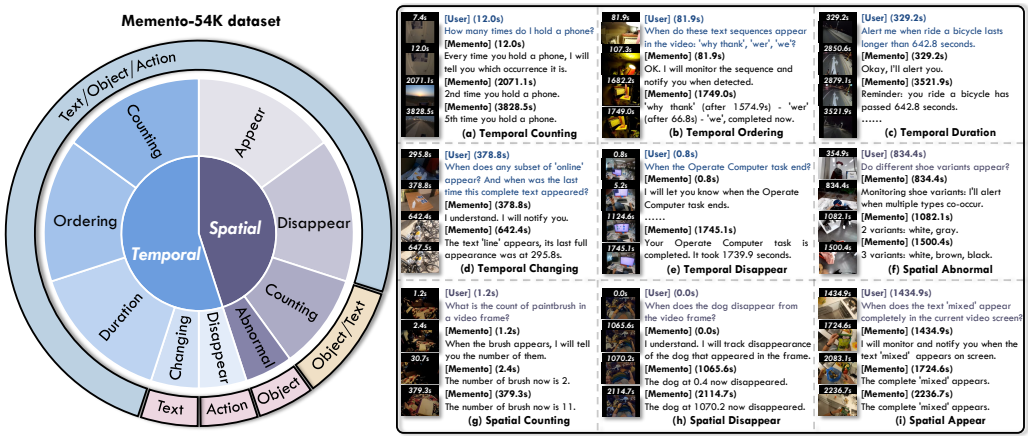


Figure 4: **Overview of Memento-54k.** Left: the 9 task types categorized by spatial vs. temporal, and by modality (text, object, action). Right: example QA instances for each task type.

4 Dataset and Benchmark: Memento-54k and MementoBench

4.1 Memento-54k Dataset Construction

Video Filtering and Sampling. To support long-duration, proactive interaction, we construct Memento-54k based on Ego4D (Grauman et al., 2022). We filter all videos to retain those between 5 minutes and 7 hours, to ensure long-term context. To reduce sample imbalance, we downsample videos from overrepresented scenarios (e.g., cooking), yielding a subset of 4,466 daily-life videos.

Task Annotation. As illustrated in Fig. 4, we define 9 task types spanning spatial and temporal reasoning, where spatial tasks focus on short-term perception (e.g., object presence), and temporal tasks require long-range memory (e.g., repeated actions or text changing). These tasks are designed for three modalities: action, object, and text. Each sample is annotated as a streaming QA pair, including a question and multiple assistant responses with timestamps. For each modality, we first obtain timestamp-level labels, and then generate QA pairs:

- ***Action.*** Based on Ego4D timestamp narrations, we prompt GPT-4o to generate QA pairs such as repeated actions of Temporal Counting on the right side of Fig. 4, see Appendix for details.
- ***Object.*** We extract objects at 2 FPS using ChatReX (Jiang et al., 2024), a category-agnostic detector. QA pairs are then generated via rule-based scripts. For example, in temporal duration tasks, we track object appearance and disappearance timestamps to identify presence for producing response.
- ***Text.*** On-screen text is detected by Qwen2-VL at 2 FPS. Text annotations are similar to object, such as temporal changing tasks identify cases where a previously seen full text is later partially disappeared, and once the subset is matched, a response is triggered to form a QA pair.

Streaming QA Formatting. For each task, up to 9 instances are annotated per video, each focusing on a distinct action, object, or text. Failed or invalid cases are manually corrected. Then, QA pairs are grouped by randomly selecting 1-5 user queries with their timestamped responses to form new streaming samples. This forms the final release of the Memento-54k dataset, and the specific distribution is as shown in Table. 2.

Split	Duration	Videos	Samples	Responses
Train	Total	4,426	53.6k	2.5M
Test	5-10 min	10	62	2.6k
	10-30 min	10	67	3.6k
	30-60 min	15	56	4.8k
	$>$ 60 min	5	13	2.5k
	Total	40	198	13.5k

Table 2: **Distribution of Memento-54k.**

Especially, streaming QA must scan entire videos, making evaluation expensive. Though it contains only 40 videos, the test set covers over 13k responses, which is sufficient for robust evaluation.

4.2 MementoBench Evaluation

To evaluate models under the proactive long-term understanding setting, we identify three essential requirements for this task: temporal alignment, answer quality, and minimal redundancy.

TimeRecall. *TimeRecall* is the fraction of ground-truth responses for which the model produces at least one response within a 5-second window, reflecting the ability to anticipate when to respond.

Score. *Score* measures the generation quality by comparing all the model responses within the above window with ground-truth answers. We use GPT-3.5-turbo-0125 to assign score from 1 to 10 and take the maximum among multiple outputs. Scoring details are provided in the Appendix.

Redundancy. *Redundancy* captures the extent of unnecessary generation, defined as the proportion of model responses outside the time window in *TimeRecall*.

The most closely related benchmark to ours is Ego4D Narration Stream (Chen et al., 2024a; Lin et al., 2022), which evaluates the temporally align performance on generated descriptions with visual events in streaming egocentric videos. However, it focuses only on the current narration, overlooking tasks that require long-term past information. In addition, its evaluation relies on exact text match, whereas MementoBench supports free-form outputs, enabling more flexible and robust assessment.

Notably, existing online benchmarks (Li et al., 2025b; Wu et al., 2024a) such as OVO-Bench, which appear to evaluate proactive interaction, in fact offer an offline-form question and predefined response timestamps during inference. All past video frames before each timestamp are provided, which ideally should be judged by the model. As a result, such benchmarks emphasize response accuracy for specified questions, allowing non-proactive models to be evaluated under this setting. In contrast, MementoBench compares whether models can proactively interact at the right time with the correct content, enabling more accurate evaluation of the desired capabilities in real-world proactive settings.

5 Experiments

5.1 Implementation Details

In this work, we implement our Memento following the VideoLLM-online framework (Chen et al., 2024a). Unless otherwise stated, we use SigLIP-ViT-L/384 (Zhai et al., 2023) as the vision encoder, which extracts frame-wise features at 2 FPS, and set $h_p = w_p = 3$. For the LLM module, we use LLaMA-3.1-8B-Instruct (Grattafiori et al., 2024). Following (Chen et al., 2024a), we train 1 epoch for our model in the DeepSpeed Zero-2 (Rajbhandari et al., 2020) configuration, with LoRA (Hu et al., 2022) to all linear layers in the LLM with a rank of 128 and a scaling factor of 256. For our DM module, we set the relevance threshold $\epsilon = 0.7$, and update ratio $u = 0.2$. In the QMS module, the top-k ratio $r_{\text{qms}} = 50\%$. We use AdamW optimizer (Loshchilov & Hutter, 2019) with a learning rate of $1e$-4 and cosine decay. All experiments are conducted on 4 NVIDIA A100 GPUs (80GB). Please refer to Appendix for inference details with a dynamic correction strategy.

5.2 Main Results

We compare our method with VideoLLM-online (Chen et al., 2024a) using MementoBench. To ensure fairness, we train VideoLLM-online with our Memento-54k dataset using the same training schedule, denoted as VideoLLM-online*.

To assess runtime scalability, Figure 5 (right) shows GPU memory usage during streaming video inference. VideoLLM-online quickly accumulates tokens and runs into OOM at about 25 minutes, with memory peaking at 80.5 GB. In contrast, Memento maintains bounded usage under 45.3 GB across the entire 4-hour streaming videos, demonstrating its advantages for proactive response to ultra-long videos with stable memory and no interruption. The occasional rises correspond to dense response periods and are reduced afterward as temporary variables are released.

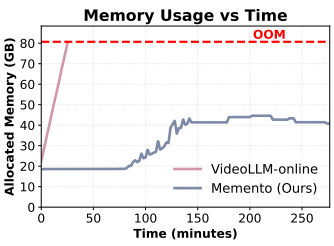


Figure 5: **Memory Usage.**

The results on MementoBench are shown in Table 3. The original VideoLLM-online performs poorly across all aspects, with only 6.1% spatial and 11.8% temporal recall, and nearly 0% beyond 25 minutes due to memory overflow. Even after supervised fine-tuning (SFT) on Memento-54k (VideoLLM-online*), average recall only rises to 8.9%, with long-term recall still at 0.3%. While it reports a higher score of 5.32 and lower redundancy of 21.3%, this is largely because it triggers

Method	*TimeRecall* ↑				*Score* ↑			*Redund.* ↓
	Sp.	*Temp.*	*Long (> 25min)*	*Avg.*	*Sp.*	*Temp.*	*Avg.*	
Online Video LLMs								
VideoLLM-online	6.1%	11.8%	0.1%	8.1%	1.55	1.21	1.40	56.4%
VideoLLM-online*	7.9%	11.6%	0.3%	8.9%	5.11	5.68	5.32	21.3%
Ours								
Memento*	**45.9%**	**51.3%**	**35.2%**	**47.5%**	4.31	4.02	4.22	64.5%

Table 3: **Evaluation on MementoBench.** *Sp.* and *Temp.* denote spatial and temporal task types, where *Temp.* requires long-term visual reasoning. *Long* marks responses beyond 25 minutes, for assessing understanding persistence under ultra-long video streams, independent of task type. In particular, VideoLLM-online is the only model with available open-source online inference code.

Memory Schema	*TimeRecall* ↑			*Score* ↑			*Redund.* ↓
	Sp.	*Temp.*	*Avg.*	*Sp.*	*Temp.*	*Avg.*	
Fixed Memory							
Len=8	14.8%	22.1%	16.9%	4.61	4.65	4.64	55.5%
Len=32	20.4%	26.4%	22.1%	5.14	5.04	5.12	53.7%
Len=128	28.1%	31.2%	29.0%	4.77	4.74	4.76	52.7%
Dynamic Memory							
ϵ=*0.6*	23.1%	25.5%	23.8%	**5.12**	**5.25**	**5.16**	**50.9%**
ϵ=*0.7*	38.2%	**46.7%**	40.4%	4.36	4.67	4.39	56.2%
ϵ=*0.8*	**43.9%**	46.6%	**44.7%**	4.59	4.05	4.43	61.4%

Table 4: **Ablation on memory schema.** "Len" indicates the fixed memory bank size. Our dynamic memory consistently yields better recall and offers superior trade-offs in others.

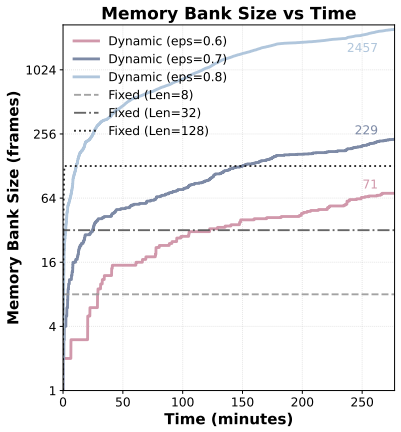


Figure 6: **Memory Size Comparison.**

very few responses, often staying silent when answers are expected. The resulting low recall makes it unsuitable for real-world applications. In contrast, Memento achieves 45.9% spatial, 51.3% temporal, and 35.2% long-duration recall, while maintaining a solid score of 4.22. Although its redundancy increases to 64.5%, given the substantial gain in recall (+38.6%), we consider this a worthwhile trade-off, as ensuring timely and consistent response is critical in ultra-long online scenarios.

5.3 Ablation Study

We conduct three ablation studies to evaluate the core design components of Memento. Our analysis focuses on three aspects: memory mechanism (with 1+2×2 frame tokens, r_{qms}=100%), frame token configuration (with ϵ=0.7, r_{qms}=100%) and QMS top-k ratio (with ϵ=0.7, 1+2×2 frame tokens).

Memory Mechanism. To examine the impact of memory structure and hyperparameter on long-term reasoning, we compare fixed-length memory banks with our dynamic memory mechanism, as shown in Table 4. Increasing the fixed memory size improves recall from 16.9% to 29.0% and slightly reduces redundancy. In comparison, dynamic memory achieves notably higher recall (up to 44.7% at $\epsilon = 0.8$) while maintaining comparable score and redundancy (up to 5.16 and 50.9% at $\epsilon = 0.6$). Notably, for temporal tasks that require long-range memory, recall improves significantly from 31.2% (fixed) to 46.7% at $\epsilon = 0.7$. Figure 6 further shows that dynamic memory scales naturally with video length, enabling long-range context retention. However, $\epsilon = 0.8$ results in nearly 10× larger memory than $\epsilon = 0.7$ with marginal gain in all the metrics, so we adopt the default $\epsilon = 0.7$.

Frame Token Configuration. We futher analyze different frame tokens in Table 5. $1 + 3 \times 3$ offers a better balance, which achieves the highest recall of 68.9%, while maintaining a reasonable score of 3.78 and moderate redundancy at 66.6%. Fewer tokens achieve a too low recall of 40.4%. $1 + 4 \times 4$ increases redundancy without improving recall.

Frame Token	*TimeRecall* ↑	*Score* ↑	*Redund.* ↓
$1 + 2 \times 2$	40.4%	**4.39**	**56.2%**
$1 + 3 \times 3$	**68.9%**	3.78	66.6%
$1 + 4 \times 4$	60.9%	3.93	67.2%

Table 5: **Ablation on frame tokens.**

r_{qms}	*TimeRecall* ↑			*Score* ↑			*Redund.* ↓	*Memory Usage* ↓
	Sp.	*Temp.*	*Avg.*	*Sp.*	*Temp.*	*Avg.*		
10%	38.7%	31.5%	33.6%	4.31	**4.67**	4.33	63.0%	39.53 GB
50%	**54.7%**	**59.5%**	**56.1%**	3.96	3.86	3.93	66.7%	45.19 GB
90%	49.0%	52.8%	50.1%	4.15	3.97	4.10	63.7%	53.50 GB
100%	38.2%	46.7%	40.4%	**4.36**	**4.67**	**4.39**	**56.2%**	55.44 GB

Table 6: **Ablation on QMS top-k ratio.** We exclude textual KV cache in "Memory Usage" reporting, as dialogue history size varies with response behavior and is independent of r_{qms}.

Figure 7: **Qualitative results of Memento on ultra-long streaming video.** The scene involves rock climbing over a 1.5-hour timeline, with three user queries issued at 0, 0, and 44 minutes, respectively. These queries cover the tasks of spatial appear, temporal disappear and temporal ordering for action.

QMS Top-k Ratio. To assess how QMS filtering affects retrieval relevance, we adjust the top-k selection ratio r_{qms} in the QMS module. As shown in Table 6, selecting all memory slots (r_{qms}=100%) results in suboptimal performance: although it achieves the highest score of 4.39 and lowest redundancy of 56.2%, its recall is notably lower compared to the best $r_{\text{qms}} = 50\%$ setting by 15.7%. This highlights that overly broad memory access may introduce irrelevant context and distract attention from key visual evidence. Meanwhile, too few slots (r=10%) limits context recall and harms performance. The 50% configuration strikes the best trade-off across all metrics, demonstrating that QMS effectively prioritizes relevant memory and improves response alignment.

5.4 Visualization of Memento

Figure 7 showcases Memento's performance on a 1.5-hour streaming video with temporally distant queries. The model identifies "holding the rope" at 27 minutes in response to an initial query and triggers "pulling the rope completed" 8 minutes later. It also tracks the ordered occurrence of "look up", "look around", and "hold the rope" before issuing a final response. Moreover, it remains proactive across the entire duration, generating correct responses even after 80 minutes, demonstrating its robustness in ultra-long streaming scenarios.

6 Conclusion

In this paper, we present Memento, a proactive vision-language framework for ultra-long streaming video. It introduces dynamic memory, query-related selection and step-aware attention for scalable long-term context modeling and temporally aligned training. Moreover, we construct Memento-54k and MementoBench for training and evaluation. Experiments show that Memento enables effective proactive interaction. Declaration of LLM usage will be discussed in Appendix.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pp. 18407–18418, 2024a.

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, 2024b.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24185–24198, 2024c.

Zesen Cheng, Sicong Leng, Hang Zhang, Yifei Xin, Xin Li, Guanzheng Chen, Yongxin Zhu, Wenqi Zhang, Ziyang Luo, Deli Zhao, and Lidong Bing. Videollama 2: Advancing spatial-temporal modeling and audio understanding in video-llms. *arXiv preprint arXiv:2406.07476*, 2024. URL https://arxiv.org/abs/2406.07476.

Tri Dao, Daniel Haziza, Francisco Massa, and Grigory Sizov. Flash-decoding for long-context inference. https://pytorch.org/blog/flash-decoding/, 2023.

Yue Fan, Xiaojian Ma, Rujie Wu, Yuntao Du, Jiaqi Li, Zhi Gao, and Qing Li. [inline-graphic not available: see fulltext]videoagent: A memory-augmented multimodal agent for video understanding. In *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part XXII*, volume 15080, pp. 75–92, 2024.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *J. Mach. Learn. Res.*, 23:120:1–120:39, 2022.

Zhangwei Gao, Zhe Chen, Erfei Cui, Yiming Ren, Weiyun Wang, Jinguo Zhu, Hao Tian, Shenglong Ye, Junjun He, Xizhou Zhu, et al. Mini-internvl: a flexible-transfer pocket multi-modal model with 5% parameters and 90% performance. *Visual Intelligence*, 2(1):1–17, 2024.

Suyu Ge, Yunan Zhang, Liyuan Liu, Minjia Zhang, Jiawei Han, and Jianfeng Gao. Model tells you what to discard: Adaptive KV cache compression for llms. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*, 2024.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, Miguel Martin, Tushar Nagarajan, Ilija Radosavovic, Santhosh Kumar Ramakrishnan, Fiona Ryan, Jayant Sharma, Michael Wray, Mengmeng Xu, Eric Zhongcong Xu, Chen Zhao, Siddhant Bansal, Dhruv Batra, Vincent Cartillier, Sean Crane, Tien Do, Morrie Doulaty, Akshay Erapalli, Christoph Feichtenhofer, Adriano Fragomeni, Qichen Fu, Christian Fuegen, Abrham Gebreselasie, Cristina Gonzalez, James Hillis, Xuhua Huang, Yifei Huang, Wenqi Jia, Weslie Khoo, Jachym Kolar, Satwik Kottur, Anurag Kumar, Federico Landini, Chao Li, Yanghao Li, Zhenqiang Li, Karttikeya Mangalam, Raghava

Modhugu, Jonathan Munro, Tullie Murrell, Takumi Nishiyasu, Will Price, Paola Ruiz Puentes, Merey Ramazanova, Leda Sari, Kiran Somasundaram, Audrey Southerland, Yusuke Sugano, Ruijie Tao, Minh Vo, Yuchen Wang, Xindi Wu, Takuma Yagi, Yunyi Zhu, Pablo Arbelaez, David Crandall, Dima Damen, Giovanni Maria Farinella, Bernard Ghanem, Vamsi Krishna Ithapu, C. V. Jawahar, Hanbyul Joo, Kris Kitani, Haizhou Li, Richard Newcombe, Aude Oliva, Hyun Soo Park, James M. Rehg, Yoichi Sato, Jianbo Shi, Mike Zheng Shou, Antonio Torralba, Lorenzo Torresani, Mingfei Yan, and Jitendra Malik. Ego4d: Around the World in 3,000 Hours of Egocentric Video. In *IEEE/CVF Computer Vision and Pattern Recognition (CVPR)*, 2022.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Bo He, Hengduo Li, Young Kyun Jang, Menglin Jia, Xuefei Cao, Ashish Shah, Abhinav Shrivastava, and Ser-Nam Lim. MA-LMM: memory-augmented large multimodal model for long-term video understanding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pp. 13504–13514, 2024.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. Metagpt: Meta programming for A multi-agent collaborative framework. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*, 2024.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *Proc. Int. Conf. Learn. Represent.*, 2022.

Zhenpeng Huang, Xinhao Li, Jiaqi Li, Jing Wang, Xiangyu Zeng, Cheng Liang, Tao Wu, Xi Chen, Liang Li, and Limin Wang. Online video understanding: Ovbench and videochat-online. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pp. 3328–3338. Computer Vision Foundation / IEEE, 2025.

Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. Adaptive mixtures of local experts. *Neural Comput.*, 3(1):79–87, 1991.

Qing Jiang, Yuqin Yang, Yuda Xiong, Yihao Chen, Zhaoyang Zeng, Tianhe Ren, Lei Zhang, et al. Chatrex: Taming multimodal llm for joint perception and understanding. *arXiv preprint arXiv:2411.18363*, 2024.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proc. IEEE*, 86(11):2278–2324, 1998.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. Gshard: Scaling giant models with conditional computation and automatic sharding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*, 2021.

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*, 2024a.

Feng Li, Renrui Zhang, Hao Zhang, Yuanhan Zhang, Bo Li, Wei Li, Zejun Ma, and Chunyuan Li. Llava-next-interleave: Tackling multi-image, video, and 3d in large multimodal models. *arXiv preprint arXiv:2407.07895*, 2024b.

Wei Li, Bing Hu, Rui Shao, Leyang Shen, and Liqiang Nie. Lion-fs: Fast & slow video-language thinker as online video assistant. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025a.

Yanwei Li, Chengyao Wang, and Jiaya Jia. Llama-vid: An image is worth 2 tokens in large language models. In *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part XLVI*, volume 15104, pp. 323–340, 2024c.

Yifei Li, Junbo Niu, Ziyang Miao, Chunjiang Ge, Yuanhang Zhou, Qihao He, Xiaoyi Dong, Haodong Duan, Shuangrui Ding, Rui Qian, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, and Jiaqi Wang. Ovo-bench: How far is your video-llms from real-world online video understanding?, 2025b. URL https://arxiv.org/abs/2501.05510.

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. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pp. 5971–5984, 2024.

Kevin Qinghong Lin, Jinpeng Wang, Mattia Soldan, Michael Wray, Rui Yan, Eric Zhongcong Xu, Difei Gao, Rong-Cheng Tu, Wenzhe Zhao, Weijie Kong, Chengfei Cai, Hongfa Wang, Dima Damen, Bernard Ghanem, Wei Liu, and Mike Zheng Shou. Egocentric video-language pretraining. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.

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

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26286–26296, 2024a.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, January 2024b. URL https://llava-vl.github.io/blog/2024-01-30-llava-next/.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Fan Ma, Xiaojie Jin, Heng Wang, Yuchen Xian, Jiashi Feng, and Yi Yang. Vista-llama: Reducing hallucination in video language models via equal distance to visual tokens. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pp. 13151–13160. IEEE, 2024.

Muhammad Maaz, Hanoona Abdul 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), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pp. 12585–12602, 2024.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, 2022.

Pranav Putta, Edmund Mills, Naman Garg, Sumeet Motwani, Chelsea Finn, Divyansh Garg, and Rafael Rafailov. Agent q: Advanced reasoning and learning for autonomous ai agents. *arXiv preprint arXiv:2408.07199*, 2024.

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

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763, 2021.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. Zero: memory optimizations toward training trillion parameter models. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 20, 2020.

Shuhuai Ren, Linli Yao, Shicheng Li, Xu Sun, and Lu Hou. Timechat: A time-sensitive multimodal large language model for long video understanding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pp. 14313–14323, 2024.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc V. Le, Geoffrey E. Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.

Enxin Song, Wenhao Chai, Guanhong Wang, Yucheng Zhang, Haoyang Zhou, Feiyang Wu, Haozhe Chi, Xun Guo, Tian Ye, Yanting Zhang, Yan Lu, Jenq-Neng Hwang, and Gaoang Wang. Moviechat: From dense token to sparse memory for long video understanding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pp. 18221–18232, 2024a.

Enxin Song, Wenhao Chai, Tian Ye, Jenq-Neng Hwang, Xi Li, and Gaoang Wang. Moviechat+: Question-aware sparse memory for long video question answering. *arXiv preprint arXiv:2404.17176*, 2024b.

Reuben Tan, Ximeng Sun, Ping Hu, Jui-Hsien Wang, Hanieh Deilamsalehy, Bryan A. Plummer, Bryan C. Russell, and Kate Saenko. Koala: Key frame-conditioned long video-llm. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pp. 13581–13591, 2024.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

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.

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.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 5998–6008, 2017.

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

Xiao Wang, Qingyi Si, Jianlong Wu, Shiyu Zhu, Li Cao, and Liqiang Nie. Adaretake: Adaptive redundancy reduction to perceive longer for video-language understanding. *ArXiv*, abs/2503.12559, 2025. URL https://api.semanticscholar.org/CorpusID:277066378.

Xiaohan Wang, Yuhui Zhang, Orr Zohar, and Serena Yeung-Levy. Videoagent: Long-form video understanding with large language model as agent. In *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part LXXX*, volume 15138, pp. 58–76, 2024b.

Yu Wang, Zeyuan Zhang, Julian McAuley, and Zexue He. Lvchat: Facilitating long video comprehension. *arXiv preprint arXiv:2402.12079*, 2024c.

Yuxuan Wang, Cihang Xie, Yang Liu, and Zilong Zheng. Mllm pressure test: Needle in a video haystack, 2024d. URL https://github.com/bigai-nlco/NeedleInAVideoHaystack.

Yuxuan Wang, Cihang Xie, Yang Liu, and Zilong Zheng. Videollamb: Long video understanding with recurrent memory bridges. *arxiv*, 2024e.

Yuetian Weng, Mingfei Han, Haoyu He, Xiaojun Chang, and Bohan Zhuang. Longvlm: Efficient long video understanding via large language models. In *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part XXXIII*, volume 15091, pp. 453–470, 2024.

Cheng-Kuang Wu, Zhi Rui Tam, Chieh-Yen Lin, Yun-Nung Chen, and Hung-yi Lee. Streambench: Towards benchmarking continuous improvement of language agents. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024a.

Shiwei Wu, Joya Chen, Kevin Qinghong Lin, Qimeng Wang, Yan Gao, Qianli Xu, Tong Xu, Yao Hu, Enhong Chen, and Mike Zheng Shou. Videollm-mod: Efficient video-language streaming with mixture-of-depths vision computation. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024b.

Huajian Xin, Z. Z. Ren, Junxiao Song, Zhihong Shao, Wanjia Zhao, Haocheng Wang, Bo Liu, Liyue Zhang, Xuan Lu, Qiushi Du, Wenjun Gao, Haowei Zhang, Qihao Zhu, Dejian Yang, Zhibin Gou, Z. F. Wu, Fuli Luo, and Chong Ruan. Deepseek-prover-v1.5: Harnessing proof assistant feedback for reinforcement learning and monte-carlo tree search. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*, 2025.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zhihao Fan. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024a.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024b.

Jingkang Yang, Shuai Liu, Hongming Guo, Yuhao Dong, Xiamengwei Zhang, Sicheng Zhang, Pengyun Wang, Zitang Zhou, Binzhu Xie, Ziyue Wang, Bei Ouyang, Zhengyu Lin, Marco Cominelli, Zhongang Cai, Yuanhan Zhang, Peiyuan Zhang, Fangzhou Hong, Joerg Widmer, Francesco Gringoli, Lei Yang, Bo Li, and Ziwei Liu. Egolife: Towards egocentric life assistant, 2025. URL https://arxiv.org/abs/2503.03803.

Linli Yao, Yicheng Li, Yuancheng Wei, Lei Li, Shuhuai Ren, Yuanxin Liu, Kun Ouyang, Lean Wang, Shicheng Li, Sida Li, Lingpeng Kong, Qi Liu, Yuanxing Zhang, and Xu Sun. Timechat-online: 80 URL https://arxiv.org/abs/2504.17343.

Xiangyu Zeng, Kunchang Li, Chenting Wang, Xinhao Li, Tianxiang Jiang, Ziang Yan, Songze Li, Yansong Shi, Zhengrong Yue, Yi Wang, Yali Wang, Yu Qiao, and Limin Wang. Timesuite: Improving MLLMs for long video understanding via grounded tuning. In *The Thirteenth International Conference on Learning Representations*, 2025.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pp. 11941–11952, 2023.

Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. *arXiv preprint arXiv:2306.02858*, 2023. URL https://arxiv.org/abs/2306.02858.

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

Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. Video instruction tuning with synthetic data, 2024b. URL https://arxiv.org/abs/2410.02713.

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

A APPENDIX

CONTENTS

A.1 Detailed Behavior of SAMA Module

The Step-Aware Memory Attention (SAMA) module defines how attention and position encodings are assigned for memory-based streaming video modeling, as illustrated in Fig. 1. Specifically:

- Only the current memory is retained, and all earlier memory tokens are discarded.
- All previous user queries and assistant responses remain visible to support long-term reasoning.
- Any `[EOS]` token is masked out from future attention—no later token can see it.

These constraints ensure that meaningful interactions remain persistently accessible, while non-response markers such as `[EOS]` are excluded to prevent memory accumulation and interference during decoding. To preserve temporal consistency, position IDs are propagated from the most recent remembered dialogue turn, enabling the model to build upon prior context. If no memory is retained, position IDs are instead reinitialized to avoid drift from irrelevant history.

```
offset = 0
# (1) Causal attention and local position ids
for block in causal_tokens:
    start, end = block_range(block)
    attention_mask[start:end, start:end] = tril(1)
    position_ids[start:end] = range(0, end - start)
# (2) Global offset shift for remembered QA (always-attend dialog)
for start, end in remembered_QA_tokens:
    attention_mask[end:, start:end] = 1
    position_ids[end:] += position_ids[end - 1] + 1 - offset
    offset = position_ids[end - 1] + 1
# (3) Mask out [EOS] and reset position alignment
for start, end in eos_tokens:
    attention_mask[end:, start:end] = 0
    if end < seq_len and tokens[end] is not Memory:
        # shift to align with last remembered content
        position_ids[end:] -= end - start
```

Figure 1: Algorithmic illustration of attention masking and position encoding in SAMA.

Position logic. In implementation, SAMA updates position IDs based on the latest remembered dialog (e.g., from a past user turn). The position ID of the first token in the current memory continues from the end of that remembered span. If a token follows a masked `[EOS]`, it continues from the most recent valid memory, skipping over the `[EOS]` as if it never existed in the context timeline.

A.2 Inference Details

To balance the timing of model responses in streaming video, VideoLLM-online introduces a correction strategy. Specifically, if the predicted probability of `[EOS]` falls below a fixed threshold θ, it is forced to zero; otherwise, it remains silent. However, we observe this approach to be highly sensitive in practice, even minor changes in training configuration may cause over-response in one case and under-response in another. This makes consistent evaluation difficult and requires expensive manual tuning per model. To address this, we design a simple dynamic adjustment strategy during inference, from an inverse perspective: instead of suppressing `[EOS]`, we explicitly require the probability of generating a `[Txt]` token to exceed a threshold θ' before triggering a response. Then, the threshold θ' will be initialized at 0.5 and increased by $\Delta_{\theta'}$=0.1 after 10 consecutive response frames to reduce over-generation. If no responses occur for 30 frames, θ' resets. This adaptive mechanism stabilizes behavior across models and is consistently applied in all our evaluations.

A.3 Performance on OVBench Benchmark

Beyond our long-form and proactive setting, we further evaluate our Memento on established online benchmarks for assessing its generalization. To this end, we conducted additional experiments on OVBench (Huang et al., 2025), a recently proposed benchmark for streaming vision-language understanding, covering diverse online tasks, as shown in Table 7.

Method	AVG	FP			THV			PM			SP		STP		TP		
		AA	GSP	MP	AP	SV	OP	AR	PR	TR	AL	OP	AT	OT	AS	SL	OES
★ VideoChat-Online	**54.9**	64.1	59.7	16.6	63.1	58.3	**62.8**	**42.2**	54.4	70.6	**54.1**	24.8	88.7	48.5	73.0	25.9	**71.7**
VideoChat-Online	53.9	56.4	63.0	15.6	57.1	57.9	61.9	39.1	54.2	**73.9**	41.3	29.7	92.2	**53.1**	69.8	27.3	69.9
Gemini-1.5-Flash	50.7	**71.4**	53.6	21.9	56.5	**60.8**	40.6	36.7	47.9	62.5	32.3	37.5	87.0	50.0	**83.3**	22.3	46.9
Qwen2-VL	49.7	60.3	**66.1**	22.1	54.9	51.5	51.1	37.8	64.4	69.3	35.3	28.5	97.0	49.4	65.1	30.8	11.7
LLaVA-OneVision	49.5	68.0	62.7	**35.9**	58.4	50.3	46.5	29.4	60.7	58.0	43.1	14.2	86.5	49.7	70.7	28.1	30.2
InternVL2-7B	48.7	52.6	60.2	27.6	57.5	52.0	58.5	38.8	**67.1**	58.3	38.1	31.3	87.4	37.0	75.4	31.4	5.9
InternVL2-4B	44.1	57.7	57.0	14.4	59.2	49.4	60.0	30.3	61.8	46.3	30.9	20.1	83.0	32.3	70.7	29.4	3.4
LongVA	43.6	64.1	56.5	29.5	54.9	51.9	34.8	35.3	55.6	57.7	31.6	3.4	67.4	44.7	80.0	26.7	4.0
LLaMA-VID	41.9	43.6	50.9	19.6	64.0	47.5	46.8	29.4	48.9	51.2	31.9	11.2	75.7	24.8	59.1	26.0	40.0
MiniCPM-V 2.6	39.1	33.3	35.9	15.0	59.2	50.8	55.1	25.0	37.4	41.7	26.6	11.8	**98.3**	36.3	66.1	26.4	6.2
VTimeLLM	33.1	37.2	23.4	15.0	**64.8**	43.8	53.2	25.9	38.8	32.5	25.9	20.4	40.9	6.8	48.4	**43.5**	8.6
★ Flash-Vstream	31.2	26.9	37.6	23.9	60.1	41.9	40.0	23.4	35.3	26.1	24.7	28.8	27.0	21.4	29.8	25.6	26.8
★ MovieChat	30.9	23.1	27.5	23.6	58.4	43.9	40.3	25.6	31.1	23.9	26.9	39.6	24.4	28.9	29.3	25.5	21.9
LITA	20.4	19.2	24.5	19.9	40.8	48.9	24.9	3.1	27.3	6.4	6.9	14.6	35.2	23.9	27.4	0.5	3.4
TimeChat	12.8	7.7	15.3	18.7	20.6	15.7	11.7	9.1	14.7	9.8	7.5	19.5	13.9	10.3	9.3	10.1	10.8
★ VideoLLM-Online	9.6	0.0	1.8	20.9	5.2	5.9	32.6	0.0	2.3	26.7	0.6	26.6	0.9	19.9	0.9	1.7	8.3
★ **Memento (Ours)**	48.5	48.7	59.9	35.6	57.9	53.7	60.5	32.2	57.6	57.4	36.3	**40.1**	64.8	36.5	64.9	36.8	33.6

Table 7: **Comparison on OVBench.** ★ indicates the input is streaming video. ***FP*** (Future Prediction) includes *AA* (Action Anticipation), *GSP* (Goal/Step Prediction) and *MP* (Movement Prediction). ***THV*** (Temporal Hallucination Verification) includes *AP* (Action Persistence), *SV* (Step Verification) and *OP* (Object Presence). ***PM*** (Past Memory) includes *AR* (Action Retrieval), *PR* (Procedure Recall), and *TR* (Trajectory Retrieval). ***SP*** (Spatio Perception) includes *AL* (Action Location) and *OP* (Object Position). **STP** (Spatio-Temporal Perception) includes *AT* (Action Trajectory) and *OT* (Object Trajectory). **TP** (Temporal Perception) includes *AS* (Action Sequence), *SL* (Step Localization) and *OES* (Object Existence State).

Method	Training	Testing	*TimeRecall* ↑				*Score* ↑			*Redund.* ↓
			Sp.	*Temp.*	*Long (> 25min)*	*Avg.*	*Sp.*	*Temp.*	*Avg.*	
Memento+ (Ours)	w/o A + B	A	33.3%	30.1%	22.0%	31.1%	4.90	5.01	4.96	35.8%
Memento+ (Ours)	w/o A + B	B	28.7%	34.9%	26.9%	31.6%	4.43	3.67	4.10	57.0%

Table 8: **Zero-shot evaluation on MementoBench.** A corresponds to *Crafting / Knitting / Sewing / Drawing / Painting* scenario, B corresponds to *Cooking* scenario.

OVBench includes 16 tasks for streaming visual-language understanding. To ensure fair comparison, we trained our model using a subset of the VideoChat-Online training data, specifically VideoChat-Online-1T and 0.27M samples from VideoChat2-1T (approximately 1/7 of the full dataset used by some advanced methods). Despite using only a fraction of the training data compared to larger baselines, our method achieves strong performance, outperforming long-context models such as LongVA and MovieChat, and matching the performance of InternVL2. Notably, our model surpasses online understanding methods like Flash-VStream and VideoLLM-online. While we fall short of VideoChat-Online with 6.4%, we attribute this gap largely to training scale. Most importantly, we emphasize that this experiment serves to demonstrate compatibility and generalization. Our primary goal remains enabling proactive assistance in ultra-long video streams, the capability not captured by existing online benchmarks. In fact, neither the tasks nor the baselines in OVBench are designed to measure such behavior.

A.4 Zero-shot Evaluation on MementoBench

Regarding zero-shot generalization performance, due to the unique nature of proactive streaming assistance over ultra-long videos, no directly compatible benchmark currently exists, so we design a strict zero-shot setting within the Memento-54k dataset. Specifically, we removed the two most frequent scene categories of "Crafting/knitting/sewing/drawing/painting" and "Cooking" (labeled by original Ego4D dataset) from training, reducing the training set size from 53.6k to 42.6k samples, and tested exclusively on these unseen scenarios. The results is shown in Table 8 This reduces the training data by 21% with an expected performance drop. Nevertheless, Memento retained stable long-range proactive response behavior and competitive scores. This experiment demonstrates that its core proactive capabilities can extend, to a certain degree, beyond the training scenes.

A.5 MEMENTO-54K ANNOTATION

A.5.1 ACTION MODALITY

Our action-oriented annotation comprises six types spanning short-term perception and long-term temporal understanding. Based on timestamped narrations from Ego4D, we employ GPT-4o to automatically construct 37,024 fine-grained question-answer (QA) pairs for action modality.

Action Spatial Appear. This task aims to identify the frame where an instance of a countable user action becomes visually observable in the video. The prompt for data generation is detailed below.

The Prompt for Spatial Appear Task in Action QA Generation

You are an excellent expert in understanding long video descriptions. Please follow the instructions below and, based on the provided video captions, help me label the data:

———————————————————

Please strictly follow these requirements for annotation:
1. In the provided video captions, the pronoun **"you"** refers to the user (i.e., "I").
2. In assistant responses, **"you"** refers to the user.
3. Automatically identify all countable events in the video captions.
- An event is countable if it refers to an action or occurrence that can be quantified.
4. For each identified event, generate the following in JSON format:
   - A user question inserted at the timestamp of the first occurrence minus 1 second (use 0 if result is negative), asking about the timing of the event.
   - For each occurrence of the event, insert an assistant response indicating that the user's action has occurred.
5. The final output must conform to the following JSON format:

```
[{
    "event": "Event Name",
    "data": [{
        "user": "User question related to identifying the event occurrences.",
        "time": Insertion time of the question (seconds)
      },{
        "assistant": "Assistant acknowledgment response.",
        "time": Insertion time of the question (seconds)
      },{
        "assistant": "Your action of [event] has come up.",
        "time": Time when the first occurrence of the event happens (seconds)
      }, ...
    ]}, ...
]
```

———— Reference Video Descriptions ————

Action Spatial Disappear. Following the spatial appear task, the spatial disappear task aims to detect the point at which an individual action instance becomes no longer visually observable. GPT-4o is prompted to identify such moments and return the end timestamp for each individual occurrence.

The Prompt for Spatial Disappear Task in Action QA Generation

You are an excellent expert in understanding long video descriptions. Please follow the instructions below and, based on the provided video captions with timestamps, help me label the data:

———————————————————

Annotation Guidelines:
1. In the provided video captions, the pronoun **"you"** refers to the user (i.e., "I").
2. When generating user questions, replace **"you"** with **"I"**.
3. In assistant responses, **"you"** refers to the user.
4. **Identify Events with Timestamps:**
   - Extract all events from the video captions along with their corresponding timestamps.
   - Ensure that the events are listed in chronological order.
5. **Select Relevant Events:**
   - Identify events that occur two or more times in the video.
   - Select up to **three** such events.
6. **Determine Event End Timestamps:**
   - For each occurrence, the end timestamp is the time of the next event that is different.
   - If the event is the last one, its end timestamp equals its start timestamp.
   - If the same event occurs back-to-back, treat each as a separate occurrence.
7. **Format the Output:**
   - Insert a user question at the beginning (time = 0) asking when the event ends.
   - Insert an assistant acknowledgment response at the same timestamp.
   - For each occurrence, add an assistant response stating that the event has ended.
8. The final output must strictly conform to the following JSON format: ...

———— Reference Video Descriptions ————

Action Temporal Duration. It focuses on estimating the duration of a single narrated event as it continuously occurs in the video. GPT-4o is prompted to determine when each event starts and ends.

The Prompt for Temporal Duration Task in Action QA Generation

You are an excellent expert in understanding long video descriptions. Please follow the instructions below and, based on the provided video captions with timestamps, help me label the data:

Annotation Guidelines:
1. In the provided video captions, the pronoun **"you"** refers to the user (i.e., "I").
2. When generating user questions, replace **"you"** with **"I"**.
3. In assistant responses, **"you"** refers to the user.
4. **Identify Events with Timestamps:**
   - Extract all events from the video captions with their corresponding timestamps.
   - Ensure events are ordered chronologically.
5. **Select Relevant Events:**
   - Identify events that occur **two or more times**.
   - Select up to **three** such events.
   - Event names must match their wording in the video captions.
6. **Determine Start and End Timestamps:**
   - For each occurrence, define the **start** as the current timestamp.
   - The **end** is the timestamp of the next different event.
   - If the event is last in sequence, its end equals its start.
   - Back-to-back identical events are treated as separate occurrences.
7. **Calculate Duration:**
   - For each occurrence, compute the duration as `end - start` in seconds.
8. **Format the Output:**
   - At the beginning, insert:
     - A user question at `time = 0`: `"How long did each [event] last?"`
     - An assistant acknowledgment at `time = 0`: `"I will inform you of the duration each time [event] ends."`
   - For each event occurrence, add:
     - `start` and `end` timestamps
     - An assistant message indicating when the event ended and how long it lasted
     - A `time` field showing the duration in seconds
9. The final output must strictly conform to the following JSON format: ...

———— Reference Video Descriptions ————

Action Temporal Disappear. This task focuses on identifying when a high-level action disappears, as indicated by a semantically coherent sequence of events bounded by a clear starting and ending event. Since the ending event alone is insufficient, the model must reason over prior context to determine whether the action has concluded. This task evaluates temporal abstraction and context-aware understanding in streaming long-form videos.

The Prompt for Temporal Disappear Task in Action QA Generation

You are an expert in analyzing video captions to identify tasks with distinct start and end events, and calculating the duration of these tasks. Please follow the instructions below to help me label the data based on the provided video captions.

Annotation Guidelines:
1. **Identify Tasks with Distinct Start and End Events:**
   - Identify tasks in the video captions.
   - Each task must have a **unique start event** and a **distinct end event**.
   - Select up to **three** tasks from the video.
   - Use the **exact wording** of the start and end events as described in the captions.
2. **Calculate Task Durations:**
   - For each task, compute the time difference between the start and end event timestamps.
3. **Generate Data for Each Task:**
   - At the task's start time (or `time = 0` for the first task), insert:
     - A user question: `"When will the [task] task end?"`
     - An assistant response: `"I will let you know when the [task] task ends."`
   - At the task's end time, insert:
     - A response indicating task completion and total duration: `"Your [task] task has been completed. It took [duration] seconds."`
4. **Format the Output:**
   - The final output should be a JSON array with entries for each task.
   - Each entry includes the task name, start/end event names, and a list of time-stamped user/assistant interactions.
5. The output must strictly follow the structure below: ...

———— Reference Video Descriptions ————

Action Temporal Counting. It focuses on tracking and responding to repeated occurrences of identifiable actions in a video. Below is the prompt used to generate the annotations.

The Prompt for Temporal Counting Task in Action QA Generation

You are an excellent expert in understanding long video descriptions. Please follow the instructions below and, based on the provided video captions, help me label the data:

Annotation Guidelines:
Important Notes:
- In the provided video captions, the pronoun **"you"** refers to the user (`"I"`).
- Replace **"you"** with **"I"** when generating user questions.
- In assistant responses, **"you"** refers to the user.

1. **Identify Countable Events:**
   - Detect all countable events from the video captions.
   - An event is countable if it refers to an action or occurrence that can be clearly quantified.
2. **Select Relevant Events:**
   - Only include events that occur **two or more times**.
   - Select up to **three** such events.
3. **Generate Data for Each Event:**
   - Insert a user question and assistant acknowledgment **one second before** the first occurrence (or at `time = 0` if the result is negative):
     - `User: "When does the event of me [event] occur?"`
     - `Assistant: "I understand. Every time you [event], I will remind you."`
   - For each subsequent occurrence:
     - Add: `Assistant: "You have [event]."`
     - Include the corresponding timestamp.
4. **Format the Output:**
   - The final output is a JSON array, one entry per event.
   - Each entry includes the event name and a sequence of time-stamped interactions.
5. The output must strictly follow the structure below: ...

———— Reference Video Descriptions ————

Action Temporal Ordering. It emphasizes the recognition of consecutive event sequences that may span long temporal intervals, and requires the assistant to respond upon their completion. It evaluates the model's ability to capture long-range temporal dependencies and track consistent action orderings.

The Prompt for Temporal Ordering Task in Action QA Generation

You are an excellent expert in understanding long video descriptions. Please follow the instructions below and, based on the provided video captions, help me label the data:

Annotation Guidelines:
Important Notes:
- In the provided video captions, the pronoun **"you"** refers to the user (`"I"`).
- Replace **"you"** with **"I"** when generating user questions.
- In assistant responses, **"you"** refers to the user.

1. **Identify All Events and Their Timestamps:**
   - Extract all events along with their corresponding timestamps.
   - Only consider events with **identical wording** as the same event.
   - Sort events chronologically.
2. **Identify the Two Most Frequent Event Pairs:**
   - A valid event pair is formed when **two different events occur consecutively** with no other events in between.
   - Identify the **two most frequent** such pairs across the captions.
3. **Generate Data for Each Event Pair:**
   - At `time = 0`, insert:
     - `User: "When did I have these consecutive events of [event 1] and then [event 2]?"`
     - `Assistant: "I understand. Every time you have these consecutive events, I will remind you."`
   - For each valid pair occurrence:
     - Insert: `Assistant: "The [event 1] and then [event 2] has occurred."`
     - Include both event timestamps and set the message time to the second event's timestamp.
4. **Format the Output:**
   - The final output should be a JSON array with two entries—one per frequent event pair.
   - Each entry includes the pair description, timestamps, and user/assistant dialogue.
5. The output must strictly follow the structure below: ...

———— Reference Video Descriptions ————

A.5.2 OBJECT MODALITY

In this section, we describe our seven object-oriented tasks for generating QA pairs based on dense object-level narrations, extracted using ChatReX at 2 FPS. Each narration includes a timestamped list of objects, such as *"<69, 278, 659, 539><wooden cabinet>"*. Based on these structured annotations, we generate initial 74,742 QA pairs by applying a fixed user-assistant interaction pattern, where placeholders are filled in accordingly. All QA pairs are stored in structured JSON format before further human correction.

Below is an example of the predefined QA pattern used for the Temporal Ordering Task:

- **User:** *"When do these object sequences appear in the video: '[A]', '[B]', '[C]'?"*
- **Assistant:** *"I understand. I will monitor the object sequence: '[A]' → '[B]' → '[C]' and notify you when detected."*
- **Assistant (at detection time):** *"Object sequence detected: '[A]' (after [X]s) → '[B]' (after [Y]s) → '[C]', completed at [T]s."*

Object Spatial Appear. Its goal is to identify when a specific object appears in the video frame. Objects are temporally counted along the video timeline, and only those with a moderate frequency of occurrence are retained for QA generation.

Object Spatial Disappear. Its goal is to identify when a specific object disappears from the video frame. We compute the maximum continuous visible duration for each object and retain only those whose presence lasts sufficiently long.

Object Spatial Counting. Its goal is to identify how many instances of a specific object appear in the current video frame. For robustness, we retain objects with a moderate number of total appearances, which are based on total frame count, with defaults set to [10, 30] for short videos.

Object Temporal Duration. It evaluates a model's temporal sensitivity by detecting both the duration of continuous visibility and the duration of absence between object reappearances. We retain objects with long visible or invisible durations and a moderate number of total appearances.

Object Temporal Counting. Its goal is to count how many times a specific object appears throughout the video. We retain objects with a moderate number of total appearances.

Object Temporal Ordering. Its goal is to detect when a specific sequence of three distinct objects appears in a consistent temporal order. We identify valid triplets that occur repeatedly (e.g., 7–20 times) with proper temporal spacing, and track their appearance timings across the video.

Object Temporal Abnormal. Its goal is to identify co-occurrence of multiple variants of the same base object (e.g., different colors or sizes of "cup") within a single frame. We retain base objects that exhibit 2 or more distinct variants co-occurring at 2–10 distinct time points.

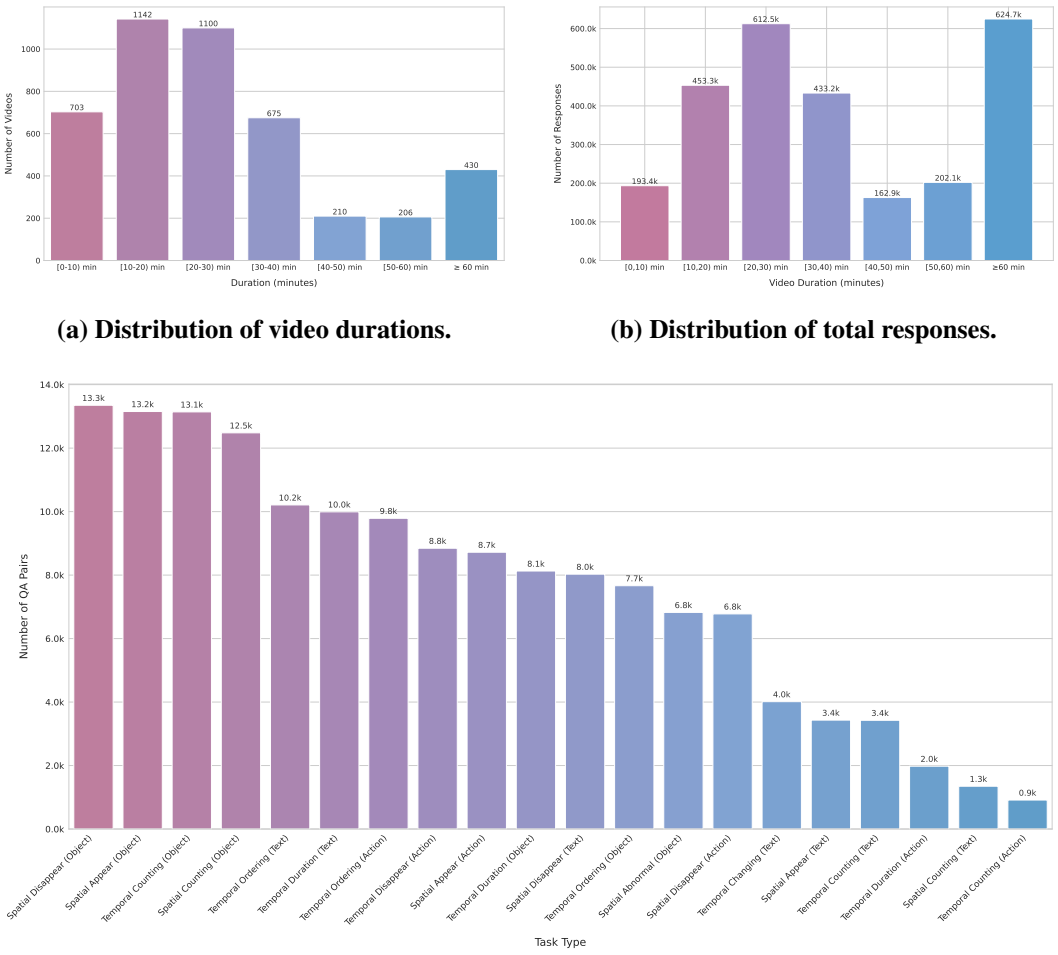
A.5.3 TEXT MODALITY

In this section, we describe our seven text-oriented tasks for generating QA pairs based on dense timestamped OCR results, extracted using Qwen2.5-VL at 2 FPS. Each frame-level annotation provides a list of visible text strings, such as *"Welcome to the Museum"*. After normalization and filtering for different languages, we generate initial 40,447 QA pairs by applying a fixed user-assistant interaction pattern, where placeholders are filled with the selected text content. All QA pairs are stored in structured JSON format before further human refinement.

Text Spatial Appear. Its goal is to identify when a specific text appears completely in the video frame. We retain text spans that are semantically complete and appear with moderate frequency (e.g., 7-20 times).

Text Spatial Disappear. Its goal is to detect when a specific text starts to leave the video screen. We retain text spans with complete structure and disappearance durations exceeding 30 seconds.

Text Spatial Counting. Its goal is to count how many instances of a specific text appear in each video frame. We keep text blocks that appear in multiple frames, and at least one frame contains multiple instances of this text.



(a) Distribution of video durations. **(b) Distribution of total responses.**

(c) Number of QA pairs per task type.

Figure 9: **Statistics of the Memento-54K dataset.** (a) shows the distribution of video durations ; (b) illustrates the number of responses across different duration; and (c) breaks down the total number of QA pairs by task type.

Text Temporal Duration. It detects the duration of continuous visibility and the duration of absence between complete text reappearances. We retain texts with the longest visible or invisible duration.

Text Temporal Counting. Its goal is to track how many times a complete text appears across the full video timeline. Texts with moderate appearance counts are retained.

Text Temporal Ordering. Its goal is to detect ordered sequences of three distinct texts appearing in succession. We retain frequent sequences and report the time intervals between their elements.

Text Temporal Changing. Its goal is to detect when a previously complete text block changes into only part of itself in later frames. The assistant alerts the user whenever such a partial form appears, along with the timestamp of the last full appearance.

A.5.4 Dataset Summary

In summary, Figure 9 presents the statistics of the Memento-54K dataset. As shown in Fig. 9 (a), the dataset covers a wide range of long-video durations, from 5 minutes to over 7 hours. Notably, Fig. 9 (b) shows that videos exceeding 1 hour account for the largest number of response (from assistant) annotations, highlighting the high density of temporal supervision required for streaming long-form

Task Name	Evaluation Focus
object_temporal_ordering	Focus on overall description, object targets, and their correct sequence.
object_spatial_appear	Focus on overall description and whether the object is correctly identified.
object_spatial_disappear	Focus on overall description and whether the object disappearance is correctly captured.
object_temporal_counting	Focus on overall description, object identity, and correct appearance count.
object_spatial_counting	Focus on overall description, object identity, and quantity in the frame.
object_spatial_abnormal	Focus on object count and presence of distinct types or variants.
object_temporal_duration	Focus on correct object identity and time duration since last occurrence.
text_spatial_appear	Focus on correctness of textual content appearance.
text_spatial_disappear	Focus on correctness of textual content disappearance.
text_spatial_counting	Focus on correct textual content and count.
text_temporal_counting	Focus on correct textual content and appearance frequency.
text_temporal_duration	Focus on textual identity and elapsed time since disappearance.
text_temporal_changing	Focus on content variations and the timing of changes.
text_temporal_ordering	Focus on correctness of textual sequence.
action_spatial_appear	Focus on correctness of the action appearance.
action_spatial_disappear	Focus on the overall disappearance of an event.
action_temporal_ordering	Focus on sequence and identity of events.
action_temporal_duration	Focus on action correctness and duration.
action_temporal_disappear	Focus on correct identification of the action disappearance.
action_temporal_counting	Focus on event identity and number of occurrences.

Table 9: Task-specific evaluation focus used in MementoBench scoring.

video understanding. Fig. 9 (c) further breaks down the QA distribution by task type, demonstrating the coverage across diverse spatial, temporal, and multimodal understanding categories.

In fact, the dataset is hierarchically structured into three levels: (1) **Responses**, each representing an individual assistant reply to an online user query; (2) **QA Pairs**, each composed of a single user question and its corresponding set of responses; (3) **Samples**, formed by randomly selecting 1-5 QA pairs from a single video. These samples comprise the 53,824 final entries in the Memento-54K.

A.6 MEMENTOBENCH EVALUATION SCORING

As described in the main paper, the *Score* metric in MementoBench is designed to assess the quality of model responses within the temporal window used by *TimeRecall*. To ensure consistency and interpretability, we use GPT-3.5-turbo-0125 as an expert judge with a structured prompt.

Scoring Prompt Used in MementoBench Evaluation

You are an expert evaluator responsible for assessing the **Answer Accuracy** of AI-generated responses based on a given user question (if any), multiple AI outputs, and a reference answer.
Each evaluation is guided by a **Task Name** and its **Key Evaluation Focus**, which indicate the specific goal and assessment priorities for this task. Please review the responses accordingly, with emphasis on the core evaluation points.

Note: Aspects not included in the core evaluation focus are considered supplementary. These are not required for the response to be deemed correct, but their accurate inclusion may enhance the overall quality.
For tasks involving estimates (e.g., time, quantity), a certain degree of deviation is acceptable.

Task Name: {task_name}
Key Evaluation Focus: {task_focus}
Scoring Criteria (1–10):
- **1–2:** Response is incorrect or contains major factual/logical errors.
- **3–4:** Partially correct, with notable inaccuracies or omissions.
- **5–6:** Mostly correct, but lacks completeness or task-specific focus.
- **7–8:** Accurate, detailed, and clearly aligned with the task goals.
- **9–10:** Fully correct, well-structured, and adds value beyond expectations.

Instructions:
- Assign a score (1–10) to each response.
- Choose the highest score as the **Overall Score**.
- Provide a concise explanation, referencing correctness and task alignment.

Output Format:

```
{"Overall Score": <score>, "Explanation": "<concise rationale>"}
```

Each scoring instance provides the task name, a user query (if available), multiple generated responses, and a reference answer. The judge is guided by a task-specific evaluation focus and is instructed to (1) assign a score between 1 and 10 to each response and (2) provide a concise explanation grounded in task relevance and factual accuracy. The final score is taken as the maximum across all candidate responses.

This explanation-enhanced evaluation enables finer-grained judgment, especially in borderline cases, and improves transparency for evaluation. The task-specific focuses are listed in Table. 9.

A.7 Declaration of LLM usage

Large Language Models (LLMs) were not involved in the conception, design, or implementation of the core methodology in this research. Their usage was limited to assisting with language polishing and improving the clarity of writing. No original scientific contributions, technical innovations, or non-standard components relied on LLMs.