VIMPRO: LONG VIDEO UNDERSTANDING VIA PROXY-GUIDED AGENTIC REASONING OVER HIERARCHICAL MEMORY

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-video understanding remains challenging due to lengthy temporal contexts and sparsely distributed question-relevant evidence. Although memory-augmented methods organize long videos into structured memories to preserve and retrieve long-range information, they still face two major challenges. First, they use a static memory as the sole inference context, limiting their ability to adaptively refine or supplement question-specific evidence. Second, similarity-based retrieval may return semantically related yet answer-irrelevant or misleading memories, whereas LLM-based agentic retrieval incurs high inference cost and is difficult to control. To overcome these limitations, we propose VimPro, a proxy-guided framework that constructs and dynamically maintains a hierarchical memory for long-video understanding. During inference, the answer agent reasons over the evolving memory and requests supplementary evidence when needed, while a lightweight proxy, MemLens, filters noisy candidates into compact answer-supporting evidence. MemLens is trained through cold-start supervised fine-tuning followed by answer-gated reinforcement learning with tailored, fine-grained rewards. Extensive experiments demonstrate that VimPro consistently outperforms both closed-source and open-source SOTA methods. All datasets, code, and trained models will be publicly released.

1 INTRODUCTION

Long-video understanding is a highly challenging task whose core objective is to accurately answer diverse user queries based on the entire video content, where task-relevant evidence is often sparsely distributed among numerous redundant frames. This requires models to both preserve long-range visual and semantic information and precisely locate fine-grained spatiotemporal details. Although multimodal large language models (MLLMs) have shown strong capabilities in image and short-video understanding (Maaz et al., 2024; Lin et al., 2024; Wang et al., 2024a; OpenAI, 2024), their effectiveness on long videos remains limited by context length, visual token capacity, and computational cost.

To address these challenges, existing methods mainly explore sparse frame sampling (Maaz et al., 2024; Lin et al., 2024), visual token compression (Li et al., 2024; Jin et al., 2024), agentic long-video reasoning (Yin et al., 2026; Qiu et al., 2026; Jain et al., 2026) and memory-augmented modeling (Song et al., 2024; He et al., 2024). Among them, memory-based methods organize long videos into structured representations instead of feeding entire videos into MLLMs, and have become a promising direction. However, most memory-based methods build a static memory offline (Long et al., 2025; Yeo et al., 2026) (Figure 1a–b): fixed and query-agnostic during inference, making it hard to compensate for missing or coarse-grained evidence and limiting the flexibility of downstream retrieval and reasoning.

Moreover, as illustrated in Figure 1, (i) M3-Agent (Long et al., 2025) retrieves the top-k relevant pieces from a static memory by textual semantic similarity. However, semantic similarity does not necessarily indicate task utility and may retrieve irrelevant or even harmful memories that mislead downstream reasoning. (ii) WorldMM (Yeo et al., 2026) improves retrieval through iterative query generation, but relies on heavy indexing (summarization, entity extraction, and graph construction)

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

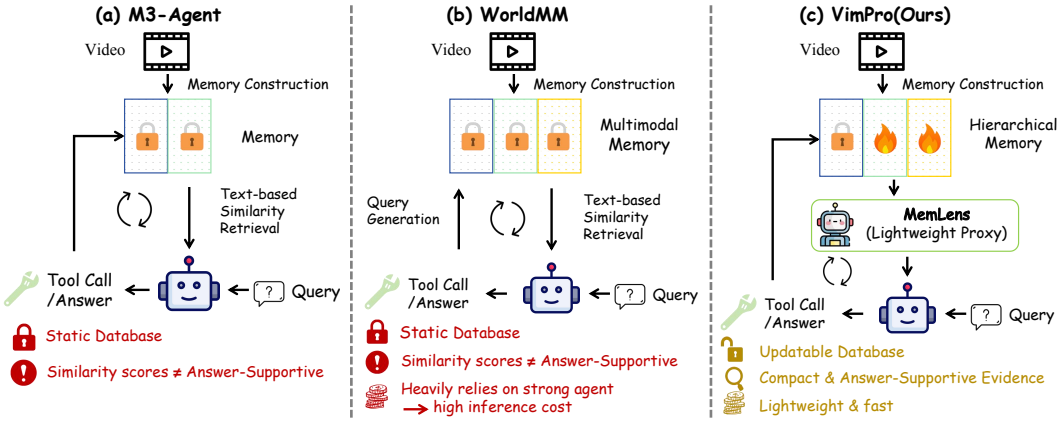


Figure 1: Comparison with memory-based long video understanding methods.

and a strong Answer-LLM for step-by-step search, incurring high inference cost. Therefore, the core bottleneck lies not only in video memory construction, but, more critically, in dynamically maintaining memory and performing task-aware, fine-grained evidence selection.

Motivated by these observations, we propose **VimPro**, a proxy-guided agentic framework with hierarchical memory for long-video understanding. The hierarchical memory comprises a static episodic memory and two dynamically maintained memories. Specifically, the episodic memory is constructed offline to preserve video semantics at multiple temporal scales. The two dynamic memories comprise evidence memory, which stores compact, question-aware evidence retrieved from the episodic memory, and working memory, which is maintained by the answer agent to track the intermediate reasoning state and determine whether supplementary retrieval is required. When the current evidence is insufficient, the answer agent can request supplementary retrieval to refine its memory state, enabling adaptive reasoning without reprocessing the full video.

To bridge large-scale video memory and answer reasoning, VimPro introduces **MemLens**, a lightweight, option-aware proxy model for fine-grained evidence selection. VimPro retrieves candidate evidence from the episodic memory and can further invoke retrieval tools when additional information is needed, after which MemLens jointly evaluates them to identify a compact set of answer-supportive evidence. To endow MemLens with task-aware evidence selection ability, we construct a high-quality dataset comprising 10K samples. We further propose a two-stage training protocol. First, we perform cold-start supervised fine-tuning (SFT) to initialize MemLens with structured generation and basic evidence selection capabilities. We then employ reinforcement learning (RL) with fine-grained rewards—particularly, a signed evidence ranking reward and a downstream answer gate—enabling the model to explore answer-oriented evidence selection policies rather than merely imitating the teacher supervision learned during the SFT stage.

Our contributions are summarized as follows:

- We propose **VimPro**, a novel agentic framework for long video understanding that decouples memory retrieval from answer reasoning by delegating fine-grained memory selection to a lightweight proxy, enabling accurate reasoning with low inference cost.
- We construct a comprehensive dataset and introduce effective training strategies—cold-start SFT and RL to align retrieval with reasoning, optimizing evidence selection for answering performance rather than semantic similarity.
- We conduct extensive experiments on multiple long video understanding benchmarks to evaluate VimPro, and the results show superior performance compared to existing baselines.

2 Related Work

Long Video Understanding. Recent Video LLMs have explored several strategies to scale multimodal understanding from short clips to long-form videos. One line of work extends the context

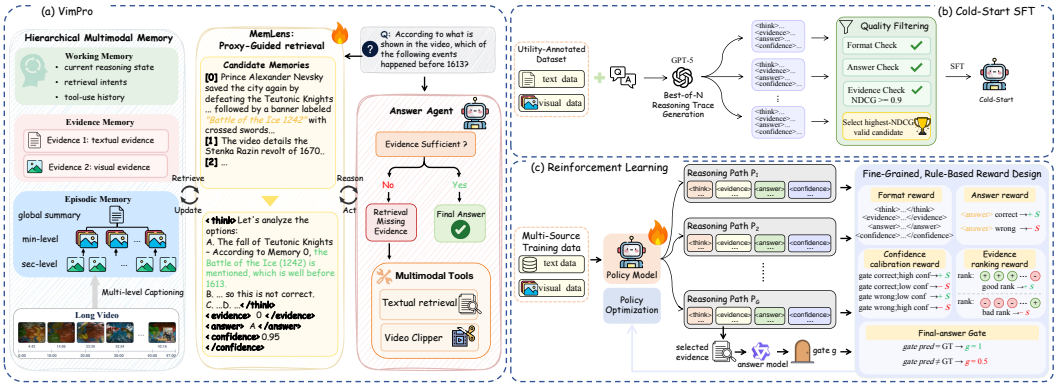


Figure 2: **Overview of VimPro.** (a) Overall architecture of the agent; (b) illustration of the cold-start SFT training process; and (c) illustration of the RL training process.

window to accommodate longer video sequences (Zhang et al., 2025a; Qian et al., 2024), but retains redundant visual information and incurs high computational costs as video duration increases. Another line sparsely samples frames or clips (Zhang et al., 2023; Maaz et al., 2024; Lin et al., 2024; Li et al., 2023; Zhang et al., 2025c) to increase temporal coverage under limited context budgets, but may miss fine-grained or transient evidence between sampled frames. A third line reduces visual token costs through token pruning, temporal pooling, or representation compression (Li et al., 2024; Jin et al., 2024; Shen et al., 2025), at the risk of losing subtle or short-lived cues.

Memory-Based Video LLMs. Memory-based methods provide a promising direction for long video understanding by explicitly storing and reusing historical video information. Early efforts introduce memory mechanisms to alleviate the limited context window of Video LLMs: MovieChat (Song et al., 2024) maintains short-term and long-term memories to compress dense video tokens, and MA-LMM (He et al., 2024) builds an online memory bank to preserve past content during sequential processing. Recent studies further organize long videos into structured memories: VideoTree (Wang et al., 2025b) constructs a query-adaptive tree and performs coarse-to-fine selection over video events. M3-Agent (Long et al., 2025) builds entity-centric episodic and semantic memories for multi-turn reasoning. WorldMM (Yeo et al., 2026) further integrates multiscale episodic, semantic, and visual memories into a unified memory architecture.

Nevertheless, these methods largely rely on static memory and adopt either similarity-based retrieval (Luo et al., 2025) or costly LLM-based search (Wang et al., 2024c; Pang & Wang, 2025; Zhang et al., 2025b). In contrast, VimPro dynamically maintains hierarchical memory and employs a lightweight proxy model to select answer-supportive evidence, decoupling retrieval from expensive answer-model reasoning.

3 Method

3.1 Architecture Overview

VimPro constructs a hierarchical multimodal memory from visual content and dynamically updates its memory state with question-aware evidence during online reasoning to support final answer generation. As illustrated in Figure 2, VimPro consists of three core components: (i) a hierarchical multimodal memory $\mathcal{M}$; (ii) MemLens, a lightweight proxy retrieval model for fine-grained evidence selection; and (iii) an answer agent that reasons over the selected evidence, adaptively invokes tools to capture additional evidence before producing the final answer.

3.2 HIERARCHICAL MEMORY ARCHITECTURE

As discussed in Section 1, an effective memory architecture for long-video understanding should support: 1) persistently preserving long-video information, and 2) dynamically selecting and maintaining answer-supportive evidence throughout reasoning. To achieve this, VimPro adopts a hierarchical architecture consisting of three memory components with complementary roles. Episodic memory serves as the persistent video memory constructed offline; evidence memory is dynamically formed and updated with question-aware evidence selected by MemLens; and working memory is maintained by the answer agent to track the current evidence and reasoning state. This design separates long-term video storage, question-conditioned evidence selection, and online reasoning, enabling VimPro to access long-video information efficiently while preserving the flexibility required for different queries.

Episodic Memory. Episodic memory is an offline, persistent memory that stores long-video semantics at multiple temporal scales. It provides a compact memory bank for subsequent evidence retrieval, covering both fine-grained actions and long-range events without repeatedly processing the full video for each question.

We define the temporal scales used to organize the video as $\mathcal{T} = \{t_0, t_1, \ldots, t_N\}$, where $t_0 < t_1 < \cdots < t_N$. The video is first divided into consecutive, non-overlapping segments of duration t_0, from which frames are uniformly sampled at 1 FPS and captioned by a video LLM. Higher-level memories are then constructed bottom-up by aggregating and summarizing lower-level memories into coarser event-level representations. Each memory item contains a textual description, its retrieval embedding, and the corresponding temporal span.

Formally, episodic memory is defined as $\mathcal{M}^{\text{epi}} = \bigcup_{t_i \in \mathcal{T}} \mathcal{M}^{t_i} \cup \mathcal{M}^{\text{g}}$, where $\mathcal{M}^{t_i}$ denotes the set of memory items at temporal scale t_i, and $\mathcal{M}^{\text{g}}$ denotes the global video-level memory. This hierarchy preserves both local event details and broader temporal context, providing reusable candidates during answer reasoning.

Evidence Memory. Evidence memory is the question-conditioned memory dynamically formed and maintained during inference. Given a question and a coarse candidate set retrieved from $\mathcal{M}^{\text{epi}}$, MemLens selects a compact set of answer-supportive and discriminative evidence. Unlike episodic memory, evidence memory retains only the memory items selected as supportive of the current question.

Evidence memory bridges long-video storage and downstream reasoning. Directly feeding coarse retrieval results into the answer agent may introduce semantically related but answer-irrelevant or even misleading memories. Evidence memory provides the answer agent with a compact evidence context and reduces the risk of reasoning over noisy or contradictory information. During iterative reasoning, evidence memory can be further refined or expanded with supplementary evidence acquired through tool invocation when the current evidence is insufficient.

Working Memory. Working memory is the transient state maintained by the answer agent during inference. It records the intermediate reasoning state, the retrieval intents, and the tool-use trajectory. This design reduces contextual redundancy and the risk of context overflow, allowing subsequent reasoning to focus on the current state and helping the answer agent avoid redundant or ineffective tool calls.

Working memory coordinates evidence retrieval and answer reasoning. When the available evidence is incomplete or ambiguous, the answer agent may request supplementary retrieval or invoke tools according to the working-memory state. This mechanism enables controlled, proxy-guided iterative reasoning while avoiding open-ended agentic search.

3.3 MEMLENS: PROXY-GUIDED EVIDENCE RETRIEVAL AND TRAINING

MemLens is the lightweight proxy retrieval model in VimPro. It performs fine-grained selection over multimodal evidence candidates, including textual memories retrieved from episodic memory and visual evidence acquired through tool invocation. Unlike a standard vector retriever that ranks memories mainly by semantic similarity, MemLens is trained to identify answer-useful evidence for

the current question. MemLens bridges coarse memory retrieval and answer reasoning by filtering irrelevant or misleading candidates and using the selected evidence to construct a compact, question-specific evidence memory.

Problem Formulation. Given a question q, options $\mathcal{O} = \{o_1, o_2, \ldots, o_K\}$, textual candidates $\mathcal{C}^{\text{txt}}$ obtained from episodic-memory retrieval and visual candidates $\mathcal{C}^{\text{vis}}$ acquired through tool invocation, MemLens learns a policy π_θ for structured evidence selection: $y \sim \pi_\theta(\cdot \mid q, \mathcal{O}, \mathcal{C}^{\text{txt}}, \mathcal{C}^{\text{vis}})$, where y consists of `<think>`, `<evidence>`, `<answer>`, and `<confidence>`. The `<think>` field describes the intermediate evidence-selection reasoning process, and `<evidence>` field defines the evidence memory items for the current question, while `<answer>` is only an auxiliary MemLens prediction and is not used as the final output. The `<confidence>` score estimates evidence reliability and sufficiency. Final answering is performed by a separate answer agent over the hierarchical memory.

Coarse-to-Fine Retrieval. VimPro adopts a coarse-to-fine retrieval strategy to reduce the search space before fine-grained evidence selection. Given a question q and candidate options $\mathcal{O}$, the system first retrieves a coarse textual candidate set $\mathcal{C}^{\text{txt}}$ from the episodic memory $\mathcal{M}^{\text{epi}}$ according to semantic similarity: $\mathcal{C}^{\text{txt}} = \text{TopK}_N\left(\{\, s(q, \mathcal{O}, m) \mid m \in \mathcal{M}^{\text{epi}} \,\}\right)$, where $s(\cdot)$ measures the similarity between the query $(q, \mathcal{O})$ and a memory unit m, and N is the retrieval budget. When the available evidence is insufficient, the answer agent may additionally invoke video inspection tools to capture a visual candidate set $\mathcal{C}^{\text{vis}}$. The final candidate set combines both sources, $\mathcal{C} = \mathcal{C}^{\text{txt}} \cup \mathcal{C}^{\text{vis}}$.

MemLens then performs fine-grained, option-aware evidence selection over $\mathcal{C}$. The evidence identified in the `<evidence>` field of its structured output y constitutes the evidence memory $\mathcal{M}^{\text{evi}}$. By combining broad textual-memory recall with targeted visual evidence acquisition, this two-stage design enables MemLens to adaptively select a compact, variable-size set of answer-supportive evidence for downstream reasoning.

Dataset Construction. We sample long-video QA instances from VideoMarathon (Lin et al., 2025) to construct our training dataset. For each instance, we build the episodic memory $\mathcal{M}^{\text{epi}}$ and apply coarse retrieval to obtain textual candidates $\mathcal{C}^{\text{txt}} = \{m_i\}_{i=1}^{N}$. We use a teacher model to assign each textual candidate a signed utility score $u_i \in [-100, 100]$ and to annotate the visual utility of fine-grained clips cropped from the temporal spans of selected memory items. After completing this annotation pipeline over the full QA pool, we sample 10K fully constructed instances for RL training and derive GPT-5-based SFT data from them, retaining 2K instances after filtering. Further details are provided in Appendices A.2 and B.3.

Two-Stage Training. Figure 2 summarizes the two-stage optimization procedure of MemLens. We first perform supervised fine-tuning (SFT) to initialize the structured output format and basic evidence selection behavior. We further optimize MemLens through RL with carefully designed rewards, enabling it to select answer-supportive evidence for downstream reasoning.

Stage 1: Cold-Start Supervised Fine-Tuning. Applying reinforcement learning directly to a general-purpose base model is unstable: initial rollouts rarely produce valid structured outputs or meaningful evidence, yielding sparse and noisy rewards. We use GPT-5 (OpenAI, 2025) to construct supervised targets from the utility-annotated data. For each instance, it generates four structured reasoning traces conditioned on textual and visual candidates. We filter traces by output format, answer correctness, and evidence quality, selecting the highest-scoring valid trace as the SFT target.

Through systematic generation and quality filtering, we construct a cold-start dataset $\mathcal{D}_{\text{cold}}$. We train MemLens with SFT, minimizing the average negative log-likelihood over all tokens in the structured output. This cold-start stage equips MemLens with stable structured generation and basic evidence-selection capabilities, providing a reliable initialization for subsequent reinforcement learning.

Stage 2: Task-Outcome-Guided GRPO Training. We further optimize MemLens through Group Relative Policy Optimization (GRPO) (Shao et al., 2024) with task-outcome-guided rewards, enabling it to explore new evidence-selection strategies under direct feedback from downstream answer outcomes, thereby moving beyond the imitation of teacher-generated selections.

Rather than relying on a single outcome-level signal, we introduce a fine-grained, answer-gated reward framework that provides complementary feedback on evidence utility, downstream answer-

ability, output validity, and confidence calibration. Formally, the reward framework consists of the following components:

- Format reward $\mathcal{R}_{\text{format}}$, which evaluates the structural validity of the model's output. It checks whether all required special tokens appear in the correct order and whether the `<evidence>` fields can be successfully parsed according to the predefined schema.
- Auxiliary-answer reward $\mathcal{R}_{\text{ans}}$ rewards correct predictions in the `<answer>` field, encouraging MemLens to select evidence that is discriminative with respect to the candidate options and promoting consistency between the selected evidence and its auxiliary prediction.
- Confidence calibration reward $\mathcal{R}_{\text{calib}}$. This reward calibrates the predicted confidence against the answer-gate MLLM, enabling MemLens to assign high confidence when the selected evidence enables the MLLM to answer correctly and low confidence otherwise.
- **Evidence ranking reward** $\mathcal{R}_{\text{evi}}$ evaluates both the utility and ordering of the evidence selected by MemLens using a signed variant of normalized discounted cumulative gain (NDCG) (Järvelin & Kekäläinen, 2002). Given the ordered items in the `<evidence>` field, we compute a discounted cumulative gain (DCG), where each selected item contributes its *signed* teacher utility score weighted by a logarithmic positional discount that assigns greater weight to more useful evidence appearing earlier in the sequence. Since the scale of the DCG may vary across questions, we normalize it on a per-question basis to obtain a reward bounded within $[-1, 1]$.
- **Final answer gate.** The teacher-based evidence ranking reward does not guarantee that the selected evidence is sufficient for final answering. We therefore introduce a soft final-answer gate that directly evaluates the answerability of the selected evidence memory. Given the question and selected evidence, an answer-gate MLLM predicts the final answer. The gate is set to $g = 1$ if the prediction matches the ground-truth answer and to $g = 0.5$ otherwise. It modulates both the evidence ranking reward and the auxiliary answer reward, allowing MemLens to receive the full rewards only when the selected evidence supports correct answering, while retaining a discounted learning signal otherwise. Since the evidence ranking reward is signed, the gate is applied to the positive part of that signal alone, so that failing the gate discounts a reward but never forgives the penalty incurred by selecting harmful evidence.

Writing $s = \mathcal{R}_{\text{evi}} + \mathcal{R}_{\text{ans}}$, the final reward is defined as $\mathcal{R} = g \max(s, 0) + \min(s, 0) + \mathcal{R}_{\text{format}} + \mathcal{R}_{\text{calib}}$. Formal definitions of each reward component are provided in Appendix B.4. Our fine-grained, answer-gated reward scheme jointly captures evidence utility and answerability, guiding MemLens to select evidence that is both discriminative and useful for final answering.

Through RL training, MemLens learns to adaptively select answer-supportive evidence from noisy candidate memories, moving beyond generic semantic relevance and constructing a compact, question-aware evidence memory for the answer agent's reasoning.

3.4 Answer Agent

The answer agent performs iterative reasoning by dynamically updating the hierarchical memory until it can produce a reliable final answer. When evidence is insufficient, the answer agent invokes multimodal tools to acquire supplementary observations, enabling progressive evidence refinement and aggregation through MemLens filtering. We next describe the multimodal tools and the iterative reasoning process of the answer agent.

Multimodal Tools. Rather than equipping the answer agent with a large collection of tools, as in prior agentic video methods (Wang et al., 2024c; Zhang et al., 2025b), which may introduce ambiguous tool selection and complex tool-use trajectories, VimPro adopts a minimal yet sufficient multimodal toolset for targeted evidence acquisition. This design simplifies the reasoning process while providing complementary access to textual and visual information.

- Textual retrieval tool. The textual retrieval tool searches the episodic memory for supplementary textual information that is not covered by the current context. Conditioned on the question, candidate options, and current reasoning state, it issues a focused query over memory items at different temporal scales and returns the most relevant textual memories and their timestamps.
- Visual inspection tool. The visual inspection tool accesses localized video spans to recover fine-grained visual or temporal details that may be omitted or inaccurately represented in the caption-

Table 1: Comparison of accuracy with various baselines on long-video QA benchmarks.

Methods	Video-MME (w/o Sub)				LongVideoBench	LVBench	Avg.
	Short	Medium	Long	Overall	Long		
Commercial VLMs							
Gemini 1.5 Pro (Gemini Team, Google, 2024)	81.7	74.3	67.4	75.0	58.6	33.1	55.6
Gemini 2.0 Flash (Google DeepMind, 2024)	–	–	63.0	–	45.7	48.3	–
GPT-5 (OpenAI, 2025)	–	–	67.9	–	64.5	60.1	–
Open-Source VLMs							
Qwen2.5-VL-72B (Qwen Team, 2025a)	81.2	74.2	63.9	73.1	–	–	–
Qwen3-VL-8B (Qwen Team, 2025b)	77.6	68.6	61.3	69.2	52.7	47.2	56.4
AdaReTaKe (Wang et al., 2025a)	80.6	74.9	65.0	73.5	–	53.3	–
Memory-based Methods							
VideoAgent (Wang et al., 2024c)	–	–	–	–	–	29.3	–
VideoTree (Wang et al., 2025b)	–	–	54.2	–	–	28.8	–
MR. Video (Pang & Wang, 2025)	–	–	61.8	–	61.6	60.8	–
DVD (Zhang et al., 2025b)	–	–	67.3	–	68.6	74.2	–
M3-Agent (Long et al., 2025)	70.8	59.1	55.8	61.9	47.6	43.5	51.0
WorldMM (Yeo et al., 2026)	79.1	69.6	66.1	71.6	61.8	52.6	62.0
VimPro (Ours)	**83.3**	**82.1**	**81.0**	**82.1**	**73.4**	**74.7**	**76.7**

based memory. Given a target temporal range, it samples frames or clips from the corresponding video segment and returns localized visual observations for subsequent reasoning.

Iterative Reasoning. Given a question, VimPro first retrieves candidate memories from the pre-constructed episodic memory, based on which MemLens initializes the question-specific evidence memory. The answer agent then begins iterative reasoning over the initial evidence memory while maintaining its working memory. At each step, the answer agent examines the current working memory and evidence memory to assess whether the available evidence is sufficient for answering. If additional information is required, it identifies the missing evidence, selects an appropriate textual or visual tool, and issues a focused retrieval request. The tool output is treated as candidate evidence and passed to MemLens for fine-grained selection before being incorporated into the evidence memory, while the retrieval intent and updated reasoning state are recorded in the working memory. The answer agent repeats this cycle to progressively refine and aggregate evidence until it determines that the evidence is sufficient or reaches the step budget, after which it produces the final answer.

4 Experiments

4.1 Benchmarks

To comprehensively evaluate VimPro's ability, we select three long-form video benchmarks: Video-MME (Fu et al., 2025), LongVideoBench (Long) (Wu et al., 2024), and LVBench (Wang et al., 2024b). All benchmarks consist of multiple-choice questions, with accuracy used as the evaluation metric under the w/o-subtitle setting.

4.2 Implementation Details

To construct the Episodic Memory, we sample video frames at 1 FPS, generate fine-grained visual captions using Qwen3.5-27B (Qwen Team, 2026), and organize the descriptions hierarchically at four temporal scales: 10 seconds, 30 seconds, 3 minutes, and 10 minutes. MemLens is based on Qwen3-VL-4B-Instruct and trained in two stages on 8 NVIDIA A100 GPUs, following Section 3: cold-start SFT followed by GRPO under the task-outcome-guided reward, with the answer-gate MLLM calling the Qwen3.5-27B API. During inference, we employ GPT-5 as the answer agent, with a maximum depth of 5 steps. Full hyperparameters are provided in Appendix B.5.

4.3 Main Results

Accuracy Comparison. We compare VimPro against three groups of baselines: (i) *commercial VLMs*, including Gemini 1.5 Pro (Gemini Team, Google, 2024), Gemini 2.0 Flash (Google

DeepMind, 2024), and GPT-5 (OpenAI, 2025); (ii) *open-source VLMs*, including Qwen2.5-VL-72B (Qwen Team, 2025a), Qwen3-VL-8B (Qwen Team, 2025b), and AdaReTaKe (Wang et al., 2025a); and (iii) *memory-based methods*, including VideoAgent (Wang et al., 2024c), VideoTree (Wang et al., 2025b), MR. Video (Pang & Wang, 2025), DVD (Zhang et al., 2025b), M3-Agent (Long et al., 2025), and WorldMM (Yeo et al., 2026).

Table 1 reports the comparison with representative baselines. VimPro achieves the best performance across all reported benchmark settings. On the short, medium, and long subsets of Video-MME, it achieves accuracies of 83.3, 82.1, and 81.0, surpassing the strongest baselines by 1.6, 7.2, and 13.1 percentage points, respectively. The widening performance margin suggests that VimPro is particularly effective for longer videos, where relevant evidence is more difficult to locate. On LongVideoBench, VimPro achieves 73.4 accuracy, outperforming the strongest baseline by 4.8 percentage points. It also reaches 74.7 on LVBench, the best result among all compared methods.

Moreover, compared with directly applying GPT-5, VimPro improves accuracy by 13.1 points on Video-MME Long, 8.9 points on LongVideoBench, and 14.6 points on LVBench. These consistent gains highlight the effectiveness of the updatable hierarchical memory and proxy-guided evidence retrieval for long-video understanding.

Efficiency Comparison. We randomly sample 10 videos from Video-MME Long for efficiency evaluation, averaging 42.2 minutes and covering 30 questions. For each method, we report the aggregate end-to-end runtime and token consumption over the complete pipeline, including video processing, memory construction, evidence retrieval, and final answer generation. For a fair comparison, all three systems answer with a locally deployed Qwen3-32B (Yang et al., 2025) instead of the GPT-5 agent used in Table 1. As shown in Table 2, VimPro completes the evaluation in 262 minutes while consuming 6.1M tokens. Compared with M3-Agent and WorldMM, it reduces runtime by 54.4% and 64.1%, and token consumption by 73.6% and 70.2%, respectively. These efficiency gains stem from VimPro's dynamically updated memory and proxy-guided retrieval pipeline. By incrementally maintaining a compact memory bank during video processing and further refining coarse candidates into a question-specific Evidence Memory, VimPro avoids repeatedly processing redundant visual content and reduces the context passed to the answer agent. VimPro achieves a favorable trade-off between question-answering performance and computational cost.

Table 2: Computational efficiency comparison. All systems run on the same RTX Pro 6000 GPU with a locally deployed Qwen3-32B answer model.

Method	End-to-End Time(min)	Token Consumption(M)
M3-Agent	574	23.1
WorldMM	730	20.5
VimPro(Ours)	**262**	**6.1**

4.4 Ablations

Hierarchical Multimodal Memory. We ablate the three components of our memory architecture—*episodic*, *evidence*, and *working* memory—by evaluating different combinations of the three modules, treating the offline episodic store as a fixed substrate and progressively adding evidence selection and working memory, as reported in Table 3.

Table 3: Ablation on the hierarchical multimodal memory.

Memory Type			Video-MME (Long)
Episodic	Evidence	Working	
✓			71.2
✓	✓		77.5
✓		✓	71.6
✓	✓	✓	**81.0**

Using episodic memory alone yields an accuracy of 71.2% on the long-video subset of Video-MME. Adding evidence memory substantially improves performance to 77.5%, a gain of 6.3 percentage points, demonstrating the importance of selecting and organizing retrieved evidence before answering. In contrast, adding working memory directly on top of episodic memory produces only a marginal improvement, from 71.2% to 71.6%. Nevertheless, combining all three memory components achieves the best performance of 81.0%, exceeding the episodic–evidence variant by 3.5 percentage points. These results suggest that evidence memory provides the largest individual gain, while working memory becomes beneficial when paired with effective evidence selection. Overall, the three components are complementary, with the full hierarchical memory architecture delivering the strongest long-video question-answering performance.

Significance of MemLens. We isolate the contribution of MemLens by keeping the coarse candidate set $\mathcal{C}$ and the answer agent fixed, while varying only whether fine-grained evidence selection is applied. In the *w/o MemLens* setting, the top-N candidates retrieved by semantic similarity are directly provided to the answer agent. In the *w/ MemLens* setting, MemLens further selects a compact, answer-supporting Evidence Memory from the same candidate set. As shown in Table 4, MemLens improves accuracy on Video-MME Long from 76.6 to 81.0, yielding a gain of 4.4 percentage points. This improvement indicates that semantic similarity alone is insufficient for reliable evidence retrieval, while option-aware filtering helps suppress answer-irrelevant or misleading candidates.

Table 4: Ablation study on our MemLens.

Setting	Video-MME (Long)
w/o MemLens	76.6
w/ MemLens (Ours)	**81.0**

Multi-turn Reasoning. We evaluate the effectiveness of multi-turn reasoning by varying the maximum number of reasoning-and-retrieval rounds T. As shown in Table 5, increasing the reasoning budget consistently improves performance on both benchmarks. In particular, increasing T from 1 to 5 improves accuracy by 1.7 percentage points on Video-MME Long and 3.0 percentage points on LongVideoBench Long. The larger gain on LongVideoBench suggests that multi-turn reasoning is especially beneficial for questions requiring evidence that is difficult to recover in a single retrieval pass. We attribute these improvements to the answer agent's ability to assess the sufficiency of the current Evidence Memory and, when necessary, request supplementary evidence from MemLens. By conditioning each subsequent retrieval on the accumulated Working Memory, the agent can refine its information needs, recover temporally dispersed evidence, and compensate for incomplete or weakly matched initial retrievals. Overall, additional reasoning rounds provide a consistent benefit, with the full five-round configuration achieving the best performance on both benchmarks.

Table 5: Ablation on the number of retrieval turns T.

T	Video-MME (Long)	LongVideoBench (Long)
1	79.3	70.4
3	79.6	73.0
5	**81.0**	**73.4**

Effectiveness of Training Strategies. To assess the contribution of each training stage, we compare four MemLens variants, all using Qwen3-VL-4B-Instruct as the backbone while keeping the rest of VimPro fixed: (1) *Base*, a prompt-driven baseline; (2) *SFT only*, trained solely with cold-start SFT; (3) *RL only*, optimized directly with RL; and (4) *SFT + RL*, which combines cold-start SFT with subsequent GRPO training. As shown in Table 6, task-specific training consistently improves performance over the base model on Video-MME. SFT provides a stable initialization by establishing the structured output format and basic evidence-selection behavior, while RL further improves performance by exploring evidence-selection strategies under direct feedback from downstream answer outcomes. In contrast, applying RL without the cold-start stage is less effective, as the model initially struggles to generate valid structured outputs and informative evidence selections. Combining both stages achieves the best overall performance, demonstrating their complementary roles in learning robust, question-aware, and answer-supportive evidence selection.

Table 6: Ablation study on our training strategies.

Training Strategy	Video-MME (Long)
Base	39.9
SFT only	67.3
RL only	74.9
SFT + RL (Ours)	**81.0**

5 Conclusion

We presented VimPro, a long-video understanding framework built on a dynamically maintained hierarchical memory architecture, enabling the answer agent to iteratively refine question-specific evidence and reasoning states without repeatedly processing the full video. To bridge large-scale video memory retrieval and answer reasoning, we introduced MemLens, a lightweight proxy model that filters noisy candidates into compact, answer-supporting evidence, and trained it through cold-start SFT followed by answer-gated RL. Extensive experiments demonstrate that VimPro achieves strong long-video question-answering performance while improving both runtime and token efficiency over recent memory-based agents. These results highlight the effectiveness of dynamic hierarchical memory and proxy-guided evidence retrieval for scalable long-video understanding.

Ethics Statement

This study follows the ICLR Code of Ethics. Our experiments use existing long-video datasets, including Video-MME, LongVideoBench, LVBench, and VideoMarathon, and involve no human-subject or animal experiments. We did not collect private participant data. The datasets were accessed under their applicable usage terms; as with other public video collections, their content may reflect privacy and representation limitations.

Reproducibility Statement

Anonymized code is included in the supplementary material. Section 3 and Appendices B and C describe the memory construction, MemLens training pipeline, reward functions, model and hardware settings, and inference procedure. The benchmark datasets are available from their original providers. We plan to release our code, trained models, and author-created data to support replication.

AI Use Statement

We used LLMs to assist with language editing, concision, and manuscript revisions. The authors made the final decisions about the study and take responsibility for its methods, results, and conclusions. This writing assistance is separate from the use of LLMs and MLLMs within VimPro itself: as detailed in Section 3, GPT-5 generates cold-start SFT traces, and a teacher model provides candidate utility scores.

References

Glenn W. Brier. Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78(1):1–3, 1950.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. M3-Embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 2318–2335, 2024.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, Peixian Chen, Yanwei Li, Shaohui Lin, Sirui Zhao, Ke Li, Tong Xu, Xiawu Zheng, Enhong Chen, Caifeng Shan, Ran He, and Xing Sun. Video-MME: The first-ever comprehensive evaluation benchmark of multi-modal LLMs in video analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Gemini Team, Google. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

Google DeepMind. Gemini 2.0 Flash. https://deepmind.google/technologies/gemini/, 2024.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Jitesh Jain, Jialuo Li, Zixian Ma, Jieyu Zhang, Chris Dongjoo Kim, Sangho Lee, Rohun Tripathi, Tanmay Gupta, Christopher Clark, and Humphrey Shi. SAGE: Training smart any-horizon agents for long video reasoning with reinforcement learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446, 2002.

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 (CVPR)*, 2024.

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

Yanwei Li, Chengyao Wang, and Jiaya Jia. LLaMA-VID: An image is worth 2 tokens in large language models. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024.

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.

Jingyang Lin, Jialian Wu, Ximeng Sun, Ze Wang, Jiang Liu, Yusheng Su, Xiaodong Yu, Hao Chen, Jiebo Luo, Zicheng Liu, and Emad Barsoum. Unleashing hour-scale video training for long video-language understanding. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Lin Long, Yichen He, Wentao Ye, Yiyuan Pan, Yuan Lin, Hang Li, Junbo Zhao, and Wei Li. Seeing, listening, remembering, and reasoning: A multimodal agent with long-term memory. *arXiv preprint arXiv:2508.09736*, 2025.

Yongdong Luo, Xiawu Zheng, Guilin Li, Shukang Yin, Haojia Lin, Chaoyou Fu, Jinfa Huang, Jiayi Ji, Fei Chao, Jiebo Luo, and Rongrong Ji. Video-RAG: Visually-aligned retrieval-augmented long video comprehension. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz 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 (ACL)*, 2024.

OpenAI. GPT-4o system card. https://openai.com/index/gpt-4o-system-card/, 2024.

OpenAI. GPT-5 system card. https://openai.com/index/gpt-5-system-card/, 2025.

Ziqi Pang and Yu-Xiong Wang. MR. Video: MapReduce as an effective principle for long video understanding. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Rui Qian, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Shuangrui Ding, Dahua Lin, and Jiaqi Wang. Streaming long video understanding with large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Jihao Qiu, Lingxi Xie, Xinyue Huo, Qi Tian, and Qixiang Ye. LongVideo-R1: Smart navigation for low-cost long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Qwen Team. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025a.

Qwen Team. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025b.

Qwen Team. Qwen3.5: Towards native multimodal agents. https://qwen.ai/blog?id=qwen3.5, 2026.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y.K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Xiaoqian Shen, Yunyang Xiong, Changsheng Zhao, Lemeng Wu, et al. LongVU: Spatiotemporal adaptive compression for long video-language understanding. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, 2025.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. HybridFlow: A flexible and efficient RLHF framework. In *Proceedings of the Twentieth European Conference on Computer Systems (EuroSys)*, 2025.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-VL: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024a.

Weihan Wang, Zehai He, Wenyi Hong, Yean Cheng, Xiaohan Zhang, Ji Qi, Xiaotao Gu, Shiyu Huang, Bin Xu, Yuxiao Dong, Ming Ding, and Jie Tang. LVBench: An extreme long video understanding benchmark. *arXiv preprint arXiv:2406.08035*, 2024b.

Xiao Wang, Qingyi Si, Shiyu Zhu, Jianlong Wu, Li Cao, and Liqiang Nie. AdaReTaKe: Adaptive redundancy reduction to perceive longer for video-language understanding. In *Findings of the Association for Computational Linguistics: ACL 2025*, 2025a.

Xiaohan Wang, Yuhui Zhang, Orr Zohar, and Serena Yeung-Levy. VideoAgent: Long-form video understanding with large language model as agent. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024c.

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 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025b.

Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. LongVideoBench: A benchmark for long-context interleaved video-language understanding. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2024.

An Yang et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Woongyeong Yeo, Kangsan Kim, Jaehong Yoon, and Sung Ju Hwang. WorldMM: Dynamic multimodal memory agent for long video reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Yufei Yin, Qianke Meng, Minghao Chen, Jiajun Ding, Zhenwei Shao, and Zhou Yu. VideoARM: Agentic reasoning over hierarchical memory for long-form video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Hang Zhang, Xin Li, and Lidong Bing. Video-LLaMA: An instruction-tuned audio-visual language model for video understanding. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations (EMNLP)*, 2023.

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. *Transactions on Machine Learning Research (TMLR)*, 2025a.

Xiaoyi Zhang, Zhaoyang Jia, Zongyu Guo, Jiahao Li, Bin Li, Houqiang Li, and Yan Lu. Deep video discovery: Agentic search with tool use for long-form video understanding. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025b.

Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. LLaVA-Video: Video instruction tuning with synthetic data. *Transactions on Machine Learning Research (TMLR)*, 2025c.

A Dataset Details

A.1 Benchmark Details

We provide additional details on the three evaluation settings used in the main paper: all three duration subsets of Video-MME (Fu et al., 2025), the validation Long subset of LongVideoBench (Wu et al., 2024), and the full LVBench benchmark (Wang et al., 2024b). All three benchmarks use multiple-choice accuracy. Table 7 distinguishes the size of each complete benchmark from the subset on which we report results.

Table 7: Dataset-level statistics and the subsets evaluated in the main paper.

Benchmark	#Videos	#QA	Evaluated subset
Video-MME (Fu et al., 2025)	900	2,700	Short/Medium/Long
LongVideoBench (Wu et al., 2024)	3,763	6,678	Validation Long
LVBench (Wang et al., 2024b)	103	1,549	Full benchmark

A.1.1 Video-MME

Video-MME (Fu et al., 2025) splits its 900 videos into three equally sized duration tiers, each with 300 videos and 900 questions: *Short* (<2 min, avg. ∼81 s), *Medium* (4–15 min, avg. ∼520 s), and *Long* (30–60 min, avg. ∼2,471 s).

A.1.2 LongVideoBench

LongVideoBench (Wu et al., 2024) uses four progressive duration tiers (8–15 s, 15–60 s, 3–10 min, 15–60 min); we emphasize the longest tier $(900\text{s}, 3600\text{s}]$. We evaluate on its *validation* Long subset, matching the LongVideoBench (Long) column in Table 1. Although the benchmark supplies both shorter tiers and a hidden-label test split, neither is included in this result.

A.1.3 LVBench

LVBench (Wang et al., 2024b) is not tiered by duration: every video exceeds 30 min (avg. ∼68 min, longest ∼2 h), and questions are organized by capability instead. It enforces *modality independence*: all evidence is carried by the visual channel and audio/subtitles are explicitly excluded. Its 26 fine-grained question types are grouped into the six core capabilities defined in Table 8. Entity Recognition (ER) and Event Understanding (EU) account for the largest share of questions, while Summarization (Sum) and Reasoning (Rea) are the hardest, with model accuracy typically below 35%.

A.2 Training Dataset Details

Our training corpus is built on top of VideoMarathon (Lin et al., 2025), a large-scale long-video dataset. We first exclude any videos that overlap at the video level with the evaluation sets of Video-MME (Fu et al., 2025), LongVideoBench (Wu et al., 2024), or LVBench (Wang et al., 2024b), so that no training video appears in any of the three evaluation benchmarks. From the remaining pool we keep only videos longer than 30 minutes. This yields 501 videos spanning 30.0–59.9 minutes (avg. 41.2 minutes) and a pool of 28,102 QA instances, with 35–61 questions per video (avg. 56.1). Table 9 summarizes the source-pool statistics, while Figure 3 shows the QA topic distribution.

Of the 28,102 QA instances, 27,880 have four options, 198 have two options, 2 have three options, and 22 have five options. We run the construction pipeline of Appendix B.3 over this full pool, then sample 10K fully constructed QA instances for RL training and derive the SFT data from them, which yields 2K instances after filtering; the two stages therefore draw on 10K unique QA instances in total. Each selected instance comprises a question and its candidate options, associated textual and visual evidence candidates, and candidate-level utility annotations.

Table 8: The six core capabilities evaluated by LVBench.

Abbr.	Capability	Description
ER	Entity Recognition	Detect/track key entities and relations
EU	Event Understanding	Scene changes, event detection/classification
KIR	Key Information Retrieval	Extract on-screen text, numbers, details
TG	Temporal Grounding	Localize moments, durations, ordering
Rea	Reasoning	Causal, affective, intent, prediction
Sum	Summarization	Cross-video synthesis of the core theme

Table 9: Statistics of the 501-video source pool from which the training data is sampled.

Statistic	**Value**
Videos	501
Video duration range	30.0–59.9 min
Avg. video duration	41.2 min
QA instances	28,102
Avg. QA/video	56.09
QA/video range	35–61

A.2.1 For SFT

The SFT set provides cold-start supervision that teaches MemLens the structured output format and basic evidence-selection behavior. For each RL instance we prompt GPT-5 with the question, the candidate options, the textual and visual candidates, and their utility annotations, and sample four independent structured reasoning traces, each of which is then screened by three filters. First, we validate the format by parsing the trace against the structured output schema used by MemLens at inference time, requiring that all special tokens appear in the expected order and that the `<evidence>` field parses into a well-formed ordered list of candidate identifiers; this is the same structural criterion later enforced at training time by $\mathcal{R}_{\text{format}}$. Second, we require the option predicted in the trace's `<answer>` field to match the ground-truth option, so that the retained supervision never pairs a selection rationale with an incorrect conclusion. Third, we score the ordered evidence list with exactly the signed NDCG used by the evidence ranking reward $\mathcal{R}_{\text{evi}}$, computed from the same teacher utility scores u_i of Appendix B.3 and normalized per question to $[-1, 1]$, and keep only traces scoring above 0.9; because the threshold applies to the normalized score, it admits a trace only when the selected evidence is close to the utility-optimal ordering for that question, ruling out traces that answer correctly while citing weak or mis-ordered evidence. An instance is retained only if at least one of its four traces clears all three filters, in which case we keep the trace with the highest evidence score as its SFT target. This yields a cold-start set $\mathcal{D}_{\text{cold}}$ of 2K supervised instances. Note that these 2K instances are a subset of the 10K RL instances and are therefore revisited during GRPO, which does not leak supervision into the RL stage: cold-start SFT only fits the teacher's output format and basic selection behavior, whereas the RL gradient is driven by the answer gate and the signed utility annotations rather than by the teacher traces.

A.2.2 For RL

The RL set contains 10K sampled QA instances and drives GRPO optimization under the task-outcome-guided reward. Each instance pairs a question with its coarse candidate set and the signed utility annotations, over which MemLens explores evidence-selection policies with feedback from the downstream answer gate.

Coarse retrieval is run separately over the two fine scales (10 s and 30 s) and the two coarse scales (3 min and 10 min), keeping the top-32 memories of each scale. The fine pool therefore holds exactly 64 candidates per question, whereas the coarse pool holds 18.9 on average (13–26; a scale with fewer than 32 memories contributes all of them). These pools are the unit of annotation rather than the unit of training: the teacher scores every candidate in them, but a training instance presents MemLens with 16 candidates, the same retrieval budget $N = 16$ it is given at inference. Table 10 reports how the teacher utility scores are distributed over these candidates. Only about one candidate in ten is strong evidence ($u_i \geq 50$), around half of the pool is near-neutral, and more than a quarter carries negative utility: coarse retrieval returns more actively misleading candidates than decisive

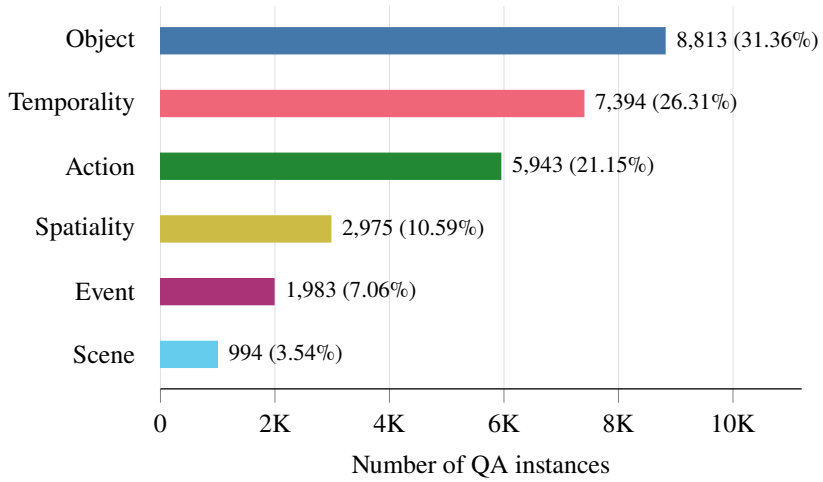


Figure 3: Topic distribution of the 28,102 QA instances in the source pool. Bar labels report the number and proportion of QA instances in each topic.

Table 10: Distribution of the teacher utility scores over the scored candidate pools.

Teacher utility u_i	Fine pool	Coarse pool
Strong positive ($u_i \geq 50$)	9.5%	10.0%
Weak positive ($20 \leq u_i < 50$)	8.5%	6.4%
Near-neutral ($-20 < u_i < 20$)	55.8%	49.5%
Weak negative ($-50 < u_i \leq -20$)	26.1%	33.4%
Strong negative ($u_i \leq -50$)	0.2%	0.8%
Mean / median u_i	-2.7 / -15.0	-8.5 / -15.0
Candidates per question	64.0	18.9

ones. This is the quantitative form of the observation that semantic similarity is a poor proxy for answer usefulness, and it is exactly what MemLens is trained to correct.

Measured per question rather than per candidate, 62.9% of questions have at least one strong candidate ($u_i \geq 50$) in the fine pool and 86.8% have at least one positive candidate; for the coarse pool the corresponding numbers are 46.4% and 70.7%. Questions whose pool contains no useful memory are not discarded. They become the abstain supervision that teaches MemLens to return an empty `<evidence>` list rather than pad its selection with the least bad candidate, and they also bound what any selector can achieve on top of this episodic memory, which we return to in Appendix F.

B Implementation Details

We provide additional details on the baseline setup, the hierarchical memory, the construction of our training data, the reward functions, the training configuration, and the prompts used.

B.1 Baseline Setup

Baselines. We compare VimPro with three groups of baselines: commercial VLMs, including Gemini 1.5 Pro (Gemini Team, Google, 2024), Gemini 2.0 Flash (Google DeepMind, 2024), and GPT-5 (OpenAI, 2025); open-source VLMs, including Qwen2.5-VL-72B-Instruct (Qwen Team, 2025a), Qwen3-VL-8B-Instruct (Qwen Team, 2025b), and AdaReTaKe (Wang et al., 2025a); and memory-based methods, including VideoAgent (Wang et al., 2024c), VideoTree (Wang et al., 2025b), MR. Video (Pang & Wang, 2025), DVD (Zhang et al., 2025b), M3-Agent (Long et al., 2025), and WorldMM (Yeo et al., 2026). Unless otherwise noted, we use results reported in the original papers, official leaderboards, or prior evaluations of the same released checkpoints. Since GPT-5 also serves as VimPro's answer agent, its directly prompted performance provides a con-

trolled reference for measuring the contribution of our memory and retrieval pipeline. We reproduce M3-Agent and WorldMM using their publicly released code and checkpoints for the accuracy comparison in Table 1. For a fair and consistent comparison with VimPro, we use Qwen3.5-27B (Qwen Team, 2026) as WorldMM's captioning model and GPT-5 (OpenAI, 2025) as its answer model. All other implementation details follow the authors' official specifications.

Efficiency Measurement. For the efficiency comparison in Table 2, all systems run on a single RTX Pro 6000 GPU with a locally deployed Qwen3-32B (Yang et al., 2025), ensuring that latency reflects computation rather than API overhead. M3-Agent uses the authors' fine-tuned checkpoint required by its jointly trained reasoning pipeline, while WorldMM and VimPro use the off-the-shelf model.

B.2 Hierarchical Memory

We use the same memory configuration for all three benchmarks. Frames are extracted at 1 FPS and the video is divided into consecutive, non-overlapping 10-second segments. The temporal scales are $\mathcal{T} = \{10\text{s}, 30\text{s}, 3\text{min}, 10\text{min}\}$, with a single video-level memory $\mathcal{M}^{\text{g}}$ on top. Each 10-second segment is captioned by Qwen3.5-27B (Qwen Team, 2026) from its sampled frames, and each coarser level is produced bottom-up by summarizing the units it covers, with summaries budgeted at 300, 1500, and 3000 words for the 3-minute, 10-minute, and video level, and at 150 words for the 30-second level. Each memory item stores a textual description, its temporal span, and a retrieval embedding produced by BGE-M3 (Chen et al., 2024) (1024-dimensional, CLS-pooled, ℓ_2-normalized, no instruction prefix), and $s(\cdot)$ is the cosine similarity between the query and the item embedding. Coarse retrieval returns the $N = 16$ highest-scoring items. The answer agent reads the memory through two tools: `text_retrieve`(query, scale, top_k) issues a new query against $\mathcal{M}^{\text{epi}}$, returns 16 items by default, and can be restricted to one temporal scale; `visual_inspect`(start, end, num_frames) samples 8 frames from a localized span of the source video and yields the visual candidates $\mathcal{C}^{\text{vis}}$. All tool outputs pass through MemLens before entering $\mathcal{M}^{\text{evi}}$. The reasoning loop is capped at $T = 5$ rounds, and tool invocation is disabled in the final round.

B.3 Dataset Construction

This section specifies the pipeline that turns each raw QA instance of the source pool into a fully constructed training instance. The resulting corpus statistics are reported in Appendix A.2.

Memory building and coarse retrieval. For each QA instance we first build the episodic memory $\mathcal{M}^{\text{epi}}$ over its source video, then run coarse retrieval against the question to obtain a textual candidate pool $\mathcal{C}^{\text{txt}}$, taken over a wider retrieval budget than inference uses; Appendix A.2 gives its size. We additionally crop fine-grained clips from the temporal spans of the retrieved memory items, which form the visual candidates scored alongside the textual ones.

Teacher utility scoring. Each candidate is then scored independently by the teacher model, Qwen3.5-27B (Qwen Team, 2026), which outputs a signed utility score $u_i \in [-100, 100]$ for a candidate memory segment m_i given the question q and options $\mathcal{O} = \{o_1, \ldots, o_K\}$:

$$u_i = \text{clip}_{[-100,100]}(p_i^{\text{ans}} + p_i^{\text{topic}} + p_i^{\text{temp}} + p_i^{\text{qual}} - n_i^{\text{off}} - n_i^{\text{temp}} - n_i^{\text{dist}}).$$

Scoring each candidate independently, rather than ranking the candidate set as a whole, makes the resulting scores comparable across questions after the per-question normalization used by $\mathcal{R}_{\text{evi}}$. Figure 4 gives the complete prompt handed to the scorer, including the criterion and numeric range of each of the four positive components and three negative penalties: the system prompt fixes the scoring principles, the seven rubric dimensions with their ranges, the final-score formula, the score-interpretation bands, and the JSON output schema, while the user prompt instantiates one candidate segment together with the question, answer choices, gold answer, and video duration.

Worked examples. Table 11 walks two candidates of the same question through the scorer, one at each end of the signed scale. The first is a strong positive: it carries the evidence the question turns

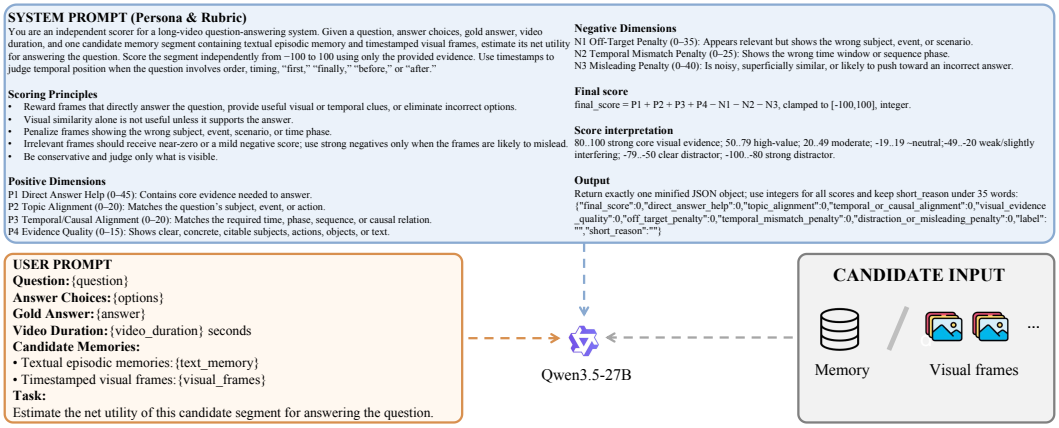


Figure 4: Prompt for candidate utility annotation.

Table 11: Worked examples of utility scoring. The supporting candidate names the gold topic (bold); the misleading candidate describes a helmet at the end of the video and pulls toward distractor option D (red).

Question	What is the main focus of the video in the first 30 seconds?
Options	A. Painting techniques B. Motorcycle repair **C. Demonstrating 3D printing** (gold) D. Helmet cleaning
Strongly answer-supporting candidate	$u_i = +92$
Description	A man in a white t-shirt and backwards cap speaks to the camera, explaining that a tutorial he has wanted to do for a long time is finally here. The scene cuts to **a 3D printer fabricating a bright green object** with the text **"3D PRINTING"** on screen. [. . .]
Strongly misleading candidate	$u_i = -85$
Description	A man completes the final assembly of a custom Iron Man helmet and presents the finished product. [. . .] Wearing white gloves, he rotates the object to display the front faceplate, sides, and back, acknowledging that the build is complete.

on, so the positive components dominate and no penalty applies. The second is a strong negative: it is superficially similar to the question's subject and would be ranked highly by semantic retrieval, but it describes the wrong occurrence of that subject and therefore pushes toward a distractor option. Contrasting the two makes explicit that a negative u_i is not simply the absence of relevance, which would score near zero, but active harm to downstream answering.

B.4 Reward Functions

Reasoning-format reward. $\mathcal{R}_{\text{format}}$ checks that the rollout output y places `<think>`, `<evidence>`, `<answer>`, and `<confidence>` in the prescribed order, and that `<evidence>` parses under the predefined schema into an ordered list of valid candidate identifiers $\mathcal{E} = (e_1, \ldots, e_{|\mathcal{E}|})$. The list may be empty: the literal `none` offered by the prompt parses as $\mathcal{E} = \varnothing$ and is a well-formed output, so abstaining is never penalized as a format violation.

$$\mathcal{R}_{\text{format}} = \begin{cases} 1, & \text{if all fields appear in the} \\ & \text{correct order and } \mathcal{E} \text{ parses,} \\ 0, & \text{otherwise.} \end{cases}$$

Auxiliary-answer reward. $\mathcal{R}_{\text{ans}}$ scores the option $\hat{o}$ that MemLens predicts in `<answer>` from the evidence it just selected, against the ground-truth option $o^\star$. Rewarding it pushes the selection toward evidence that discriminates among the options rather than evidence that is merely on-topic:

$$\mathcal{R}_{\text{ans}} = \begin{cases} 1, & \text{if } \hat{o} = o^\star, \\ 0, & \text{otherwise.} \end{cases}$$

Evidence ranking reward. $\mathcal{R}_{\text{evi}}$ scores both the utility and the ordering of $\mathcal{E}$ against the teacher utility scores $u_i \in [-100, 100]$ of Appendix B.3, using a signed variant of NDCG (Järvelin & Kekäläinen, 2002):

$$\mathrm{DCG}(\mathcal{E}) = \sum_{k=1}^{|\mathcal{E}|} \frac{u_{e_k}}{\log_2(k+1)}.$$

Because u_i is signed, DCG can take either sign, and its scale depends on how much genuinely useful evidence a given question admits. We therefore normalize per question against the best and worst orderings of the same size: DCG^+ takes the $|\mathcal{E}|$ candidates of highest utility in decreasing order, DCG^- the $|\mathcal{E}|$ candidates of lowest utility in increasing order:

$$\mathcal{R}_{\text{evi}} = \begin{cases} \dfrac{\mathrm{DCG}(\mathcal{E})}{\mathrm{DCG}^+}, & \text{if } \mathrm{DCG}(\mathcal{E}) \geq 0, \\ \dfrac{\mathrm{DCG}(\mathcal{E})}{|\mathrm{DCG}^-|}, & \text{otherwise,} \end{cases}$$

which bounds $\mathcal{R}_{\text{evi}}$ within $[-1, 1]$ and makes it comparable across questions with different candidate pools. Textual and visual candidates never appear in the same list: MemLens is queried either on textual candidates or on visual ones, under the same four-tag protocol and the same template with the numbered captions replaced by timestamped frames. Everything else is identical — both modalities are scored by the same teacher on the same signed scale, candidates are referred to by their 0-indexed position in whichever list is shown, and $\mathcal{E}$ and DCG are computed the same way.

Abstention. A rollout may legitimately select nothing. Both $\mathrm{DCG}(\mathcal{E})$ and DCG^+ vanish on an empty selection, so $\mathcal{R}_{\text{evi}}$ is defined on it directly: abstention scores $+1$ when the candidate pool contains no positive-utility candidate, and -1 otherwise. Any non-empty selection on such a pool scores at most 0, since its DCG is non-positive. Rewarding the empty selection only where no useful evidence exists is what makes abstention a distinct target rather than a safe way of avoiding a negative score: on a pool that does admit useful evidence, returning nothing is penalized as heavily as returning the worst possible ordering. For the remaining degenerate case in which a non-empty selection meets $\mathrm{DCG}^+ = 0$ — every candidate in the pool is exactly neutral — we set $\mathcal{R}_{\text{evi}} = 0$.

Final answer gate. The teacher utilities say which candidates *look* useful, but not whether the selected set is jointly sufficient to answer. We therefore query the answer-gate MLLM f_{gate} with the question q, its options $\mathcal{O}$, and the selected evidence only, and compare its prediction against the ground-truth option $o^\star$:

$$g = \begin{cases} 1, & \text{if } f_{\text{gate}}(q, \mathcal{O}, \mathcal{E}) = o^\star, \\ 0.5, & \text{otherwise.} \end{cases}$$

An empty selection is never sent to the gate: with no evidence to read, f_{gate} could only guess, and on a four-option question it would return the gold answer a quarter of the time. The rollout's own `<answer>` and `<confidence>` are just as uninformed on such a question. We therefore drop both $\mathcal{R}_{\text{ans}}$ and $\mathcal{R}_{\text{calib}}$ whenever the candidate pool admits no positive-utility candidate, and score the rollout by $\mathcal{R}_{\text{evi}}$ and $\mathcal{R}_{\text{format}}$ alone, so that a correct abstention earns its full $+1$ instead of a score that turns on a coin flip. The gate plays no role on these questions in any case: an abstention bypasses it, and any non-empty selection has $s \leq 0$ and is therefore never multiplied by g. The gate is soft rather than binary on purpose: a rollout whose evidence does not yet support a correct answer still retains half of its ranking and answer signal, which keeps early training from collapsing to zero reward before MemLens produces answerable selections. Because $\mathcal{R}_{\text{evi}}$ is signed, however, a gate applied to the raw sum would halve penalties as well as rewards, so that a rollout selecting harmful evidence would be forgiven half of its penalty precisely when the gate also fails on it. We therefore gate the positive part of the signal only. Writing $s = \mathcal{R}_{\text{evi}} + \mathcal{R}_{\text{ans}}$, the composite reward optimized by GRPO is

$$\mathcal{R} = \begin{cases} \mathcal{R}_{\text{evi}} + \mathcal{R}_{\text{format}}, & \text{if the pool admits no positive-utility candidate,} \\ g\max(s, 0) + \min(s, 0) & \\ \quad + \mathcal{R}_{\text{format}} + \mathcal{R}_{\text{calib}}, & \text{otherwise,} \end{cases}$$

with $\mathcal{R}_{\text{format}}, \mathcal{R}_{\text{ans}} \in \{0, 1\}$, $\mathcal{R}_{\text{evi}} \in [-1, 1]$ and $\mathcal{R}_{\text{calib}} \in [-1, 0]$.

[SYSTEM] You are a professional annotator specializing in visual descriptions of video clips.
You will be given a sequence of still images sampled from a short video, arranged chronologically and accompanied by their corresponding time positions. Your task is to produce one concise caption that accurately summarizes what is visually happening throughout the clip.
Requirements:

- Identify the visible subjects, their movements, relevant objects, and the surrounding environment.
- Describe events in the same order in which they occur.
- Base the description only on visual evidence.
- Do not infer dialogue, music, sound effects, or other audio information.
- Use objective, neutral language and avoid unsupported assumptions about motives, feelings, or relationships.
- Do not refer to timestamps, individual images, sampling, or the input format.
- Return only the completed caption, with no introduction, explanation, or additional formatting.

[USER] Segment window: {t0} to {t1}
The following frames are ordered chronologically within the segment.
Frame timestamp: {ts}
⟨*the frame image*⟩
... (one timestamp/image pair per sampled frame)
Write the caption for this segment.

Figure 5: Prompt for base-level segment captioning.

Confidence calibration reward. $\mathcal{R}_{\text{calib}}$ ties the value $c \in [0, 1]$ reported in `<confidence>` to whether the gate was in fact able to answer from the selected evidence, $z = \mathbf{1}[f_{\text{gate}}(q, \mathcal{O}, \mathcal{E}) = o^\star] \in \{0, 1\}$, using the Brier score (Brier, 1950). It is defined only on questions whose pool admits useful evidence, since elsewhere the gate is never queried and the term is dropped:

$$\mathcal{R}_{\text{calib}} = -\,(c - z)^2 \in [-1, 0].$$

The target is the binary gate outcome rather than the softened gate value g, so that the score remains proper over the whole interval $[0, 1]$: it is maximized only by reporting the true probability that the selection is sufficient, and neither constant optimism nor constant pessimism can exploit it. Calibrating against g instead would place the optimal report at $0.5 + 0.5\,p$ for a selection sufficient with probability p, leaving the lower half of the confidence scale unusable. The term is deliberately left outside the gate: confidence should be rewarded for being *accurate*, including when it correctly reports that the selected evidence is insufficient.

B.5 Training

MemLens is trained in two stages on 8 NVIDIA A100 (80 GB) GPUs with a maximum context length of 32K tokens. The cold-start SFT stage uses a learning rate of 1×10^{-5}, a total batch size of 256, and runs for 2 epochs. The GRPO stage is implemented with the VERL framework (Sheng et al., 2025), using a batch size of 256, 8 rollouts per question, a constant learning rate of 1×10^{-6}, and 2 epochs; the answer gate queries the external Qwen3.5-27B API.

B.6 Prompts

This section gives the prompts that drive the components of VimPro. Two of them build the episodic memory offline: Figure 5 captions a base-level segment and Figure 6 aggregates lower-level memories into a coarser level. The other two are used at inference time: Figure 7 is the structured-output template of MemLens, and Figure 8 is the system prompt of the answer agent. Placeholders in braces are instantiated per call. The prompt of the teacher utility scorer used during training-data construction is shown separately in Figure 4.

[SYSTEM] You are a professional summarizer specializing in event-level descriptions of long videos. You will be given ordered summaries of consecutive sub-segments, each accompanied by its own time range, that together cover a longer span of a single video. Your task is to consolidate them into one chronological event summary of that span.
Requirements:

- Merge related content into major events in chronological order, giving each event a clear beginning, progression, and key turning points.
- Preserve the subjects' decisions and actions in full, including the purpose and method of each key action.
- Keep discussions concise, retaining only the core topics and conclusions rather than incidental exchanges.
- Base the summary only on the sub-segment summaries, without adding atmosphere, emotion, or interpretation beyond them.
- Do not refer to sub-segments, summaries, timestamps, or the input format.
- Keep the entire output strictly below {max_words} words, and do not report the word count.
- Use plain text with one paragraph per major event, opening each paragraph with a single summarizing sentence and following it with the key actions and details in order.
- Return only the completed summary, with no introduction, explanation, or additional formatting.

[USER] Segment window: {t0} to {t1}
Sub-segment summaries:
[{c1_start}s-{c1_end}s] {c1_summary}
[{c2_start}s-{c2_end}s] {c2_summary}
...
Write the event summary for this window.

Figure 6: Prompt for bottom-up hierarchical aggregation.

C VimPro Inference Pipeline

Algorithm 1 details the complete inference procedure. We use Select_θ to denote MemLens evidence selection, applied separately to the textual and to the visual candidates of a round, since the two modalities are never shown in the same list. Evidence accumulates across rounds: what MemLens keeps is appended to the evidence memory and is not re-examined later, so each call filters only the candidates that round's tool returned. The working memory update records the latest reasoning state, retrieval intent, and tool-use trajectory. In the algorithm, Split partitions a mixed tool output into textual and visual candidates before MemLens processes them, $\mathrm{ParseToolRequest}$ reads the `<call>` and `<crop>` tags of the agent's output to obtain the requested tool h_t and its arguments r_t, and $\mathrm{FinalDecode}$ re-queries the answer agent with both tag options removed from the prompt, which leaves it no option but to commit to an answer. The indicator τ_t withholds the tools in the final round, so a run issues at most $T-1$ tool calls, and $\mathrm{FinalDecode}$ is reached only when the agent has still committed to no answer by then.

D Qualitative Analysis

Figure 9 contrasts VimPro with two baselines on a question that asks *why* a row of dogs lies by a window. Handed the video directly as 16 uniformly sampled frames, GPT-5 sees neither of the two brief staged photos — the first, spanning 720–760 s, falls between the samples at 667 and 815 s, and the second lies past the final sample at 2297 s — and answers "begging for food" from the ambient cafe footage that survives; the evidence is present in the video but not on the grid. WorldMM instead issues two rounds of visual search and answers from 66 retrieved frames, yet its second query restates a distractor as a premise — "dogs lined up by a window, eyes closed, resting on beds in a sunbeam" — and then searches for frames matching it. Pixels returned against such a query can never contradict it, because a frame asserts nothing on its own, so the baseline retrieves more and more evidence while converging on the wrong option. VimPro instead retrieves a 30 s memory that already reports four people kneeling behind a row of corgis "to pose for a photograph",

[SYSTEM] You are a proxy evidence selector for a long-video question answering system. You are NOT the final answerer; you select evidence and provide an auxiliary answer prediction based only on the selected evidence.
You are given a multiple-choice question, its answer options, and a numbered list of candidate memory segments retrieved from a long video. Each memory is a textual summary of a video segment. Some memories are useful for answering the question; others are irrelevant, weakly useful, contradictory, or misleading.
Do BOTH of the following:

1. Select the memories that genuinely help answer the current question, and order them from most useful to least useful. Omit memories that are irrelevant, off-topic, redundant, contradictory, or misleading. Keep only answer-useful memories; there may be far fewer than the total, and there may be none.
2. Based only on the selected memories, provide an auxiliary answer prediction.

Output format (STRICT) — exactly these four tags, in this order, nothing else:
¡think¿brief evidence-selection reasoning: why the kept memories support the answer and why the others are omitted; keep it concise, no long chain-of-thought¡/think¿
¡evidence¿comma-separated 0-indexed memory numbers worth keeping, most useful first; use none if no memory is useful; no duplicates¡/evidence¿
¡answer¿the letter of the predicted option¡/answer¿
¡confidence¿a number from 0 to 1 indicating how reliable and sufficient the selected evidence is for answering the question¡/confidence¿
Example:
¡think¿Memory 2 directly supports option B; memory 0 provides useful context. Memories 1 and 3 are off-topic or misleading.¡/think¿
¡evidence¿2,0¡/evidence¿ ¡answer¿B¡/answer¿ ¡confidence¿0.85¡/confidence¿

[USER] Question: {question}
Options: {options}
Candidate memories ({n} total):
[0] {memory_0}
[1] {memory_1}
...
Important: Do not select a memory only because it is semantically related to the question. Select it only if it helps distinguish the correct option from the distractors.
Select the useful memories (most useful first, omit useless/misleading ones) and give your auxiliary answer, using the required four-tag format: ¡think¿...¡/think¿ ¡evidence¿kept memory numbers from 0..{n-1}, or none¡/evidence¿ ¡answer¿predicted option letter¡/answer¿ ¡confidence¿0-1¡/confidence¿.

Figure 7: Prompt for proxy evidence selection with MemLens.

which localizes the span and makes its single tool call directed rather than speculative; the three 10 s memories that call returns each bind the dogs to the kneeling humans and the raised smartphone in one sentence, and the agent answers correctly. The difference is not retrieval effort — VimPro issues one call where WorldMM issues two — but form: what VimPro retrieves is an assertion that can be refuted, and the reason the dogs are lined up is *recorded* at memory-construction time rather than inferred at answer time.

E Case Study

We present three cases, each isolating one design decision, and report the full trajectory of VimPro rather than only its final answer: the coarse candidate set retrieved from the episodic memory, the evidence MemLens keeps and discards, any tool call the answer agent issues, and the resulting answer.

Case 1: End-to-end trajectory. Table 12 traces a question that VimPro answers without any tool call. It illustrates the intended division of labour: coarse retrieval recalls broadly at the cost of precision, and MemLens compresses that set into a small evidence memory before the answer agent ever sees it. The gap between the size of $\mathcal{C}^{\mathrm{txt}}$ and that of $\mathcal{M}^{\mathrm{evi}}$ is the context the answer agent never

[SYSTEM] You are answering a multiple-choice question about a long video. You do not see the video itself; instead you are given two memories:

- Evidence Memory — the video segments MemLens has selected as supportive of the current question, each tagged with its time window [start s - end s].
- Working Memory — your own reasoning state so far: the time windows you have already drilled into, the frame sets you have already cropped, and the remaining tool budget.

If the Evidence Memory lacks the detail needed, you may request more (at most {max_calls} call(s) total; START/END = integer seconds):

- ¡call¿sec-level, START, END¡/call¿ (finer text evidence)
- ¡crop¿START, END¡/crop¿ (actual video frames to look at)

Everything you request is filtered by MemLens and merged into the Evidence Memory, and your retrieval intent is appended to the Working Memory. Plan before each call and reflect on every result. Do not guess — use the tools to gather evidence, and give the final answer only when you are confident. Each cropped frame has its timestamp (M:SS or H:MM:SS) burned in white text at the top-left corner.

[USER] The video is {duration} long ({duration_s}s in total); any tool time window must fall within 0-{duration_s}s.
Evidence Memory ({n} segments):
— Segment 1 [{t_start}s-{t_end}s] —
Summary: {summary}
...
Working Memory:

- Time windows already drilled (do not repeat them): [{s}-{e}s], ...
- {k} cropped frame set(s) are attached below for you to look at.
- Tool calls remaining: {budget}

Question: {question}
Choices: (A) ... (B) ... (C) ... (D) ...
If the Evidence Memory above is enough, answer with a single letter (A/B/C/D) only. Otherwise request second-level evidence with ¡call¿sec-level, START, END¡/call¿ or view actual frames with ¡crop¿START, END¡/crop¿.

Figure 8: Prompt for the tool-using answer agent.

has to read: here MemLens keeps 2 candidates, and the answer agent reads only those two. The six discarded candidates shown in the table are all on topic – the same mother and cub, on a riverbank, in a tree, or hunting – but none of them says anything about the male, which is what the question turns on.

Case 2: Discarding a misleading candidate. Coarse retrieval ranks by semantic similarity, so a memory that mentions the subject of the question at the wrong moment in the video is exactly the kind of candidate it scores highly. Table 13 shows such a case: the top-ranked candidate is not merely uninformative but points to a distractor option, and MemLens drops it in favour of a lower-ranked candidate that resolves the question. This is the failure mode the evidence memory is designed to prevent, and it is why the utility scores of the training data are signed rather than non-negative: a candidate of this type must be pushed below the irrelevant ones, not merely below the useful ones. Feeding the same coarse candidate set directly to the answer agent flips the prediction to the distractor, which is the per-question view of the retriever ablation.

Case 3: A tool call resolving an under-specified caption. Coarse retrieval works at the scale at which a question is phrased, and a question about one specific moment can retrieve a set of candidates that are all about the right topic at the wrong granularity. Table 14 shows such a question: every retrieved candidate describes the food of the episode, none describes the meeting the question asks about, and MemLens reports a low confidence on the selection it can make. The answer agent reads that as a signal to localize rather than to answer, and drills the opening span of the video for second-level evidence. The case makes two points. First, the tool is directed rather than exploratory: the agent names what it is missing and the span it expects to find it in. Second, the returned candi-

Algorithm 1 VimPro Inference Pipeline

Require: video V, question q, options $\mathcal{O}$, episodic memory $\mathcal{M}^{\text{epi}}$, MemLens π_θ, answer agent f_{ans}, tools $\mathcal{H}$, maximum rounds T
Ensure: final answer a

$\mathcal{C}_0^{\text{txt}} \leftarrow \text{RetrieveText}(q, \mathcal{O}, \mathcal{M}^{\text{epi}})$
$\mathcal{M}_0^{\text{evi}} \leftarrow \text{Select}_\theta(q, \mathcal{O}, \mathcal{C}_0^{\text{txt}})$
$\mathcal{M}_0^{\text{work}} \leftarrow \text{Initialize}(q, \mathcal{O}, \mathcal{M}_0^{\text{evi}})$
for $t = 1, \ldots, T$ **do**
  $\tau_t \leftarrow \mathbf{1}[t < T]$ {the tools are withheld in the final round}
  $d_t \leftarrow f_{\text{ans}}(q, \mathcal{O}, \mathcal{M}_{t-1}^{\text{evi}}, \mathcal{M}_{t-1}^{\text{work}}, \tau_t)$
  if d_t contains a final answer a **then**
    return a
  end if
  if d_t requests no tool **then**
    break
  end if
  $(h_t, r_t) \leftarrow \text{ParseToolRequest}(d_t)$
  $\mathcal{C}_t \leftarrow h_t(r_t; V, \mathcal{M}^{\text{epi}})$, where $h_t \in \mathcal{H}$
  $(\mathcal{C}_t^{\text{txt}}, \mathcal{C}_t^{\text{vis}}) \leftarrow \text{Split}(\mathcal{C}_t)$
  $\mathcal{E}_t^{\text{txt}} \leftarrow \text{Select}_\theta(q, \mathcal{O}, \mathcal{C}_t^{\text{txt}})$, $\mathcal{E}_t^{\text{vis}} \leftarrow \text{Select}_\theta(q, \mathcal{O}, \mathcal{C}_t^{\text{vis}})$
  $\mathcal{M}_t^{\text{evi}} \leftarrow \mathcal{M}_{t-1}^{\text{evi}} \cup \mathcal{E}_t^{\text{txt}} \cup \mathcal{E}_t^{\text{vis}}$
  $\mathcal{M}_t^{\text{work}} \leftarrow \text{Update}(\mathcal{M}_{t-1}^{\text{work}}, d_t, h_t, r_t, \mathcal{M}_t^{\text{evi}})$
end for
$a \leftarrow \text{FinalDecode}(f_{\text{ans}}, q, \mathcal{O}, \mathcal{M}^{\text{evi}}, \mathcal{M}^{\text{work}})$
return a

dates enter the evidence memory only after passing through MemLens, which cuts 16 second-level candidates to 2 and raises its confidence from 0.6 to 0.9.

F LIMITATIONS

VimPro can only reason over what the episodic memory recorded and what coarse retrieval surfaces from it: the memory is caption-based, so a detail the captioner omits or states incorrectly is invisible to MemLens however the evidence is selected, and the visual inspection tool recovers detail only from a span that textual evidence has already localized rather than searching for one retrieval never returned. A question whose supporting segment is missed at the coarse stage is therefore unanswerable regardless of the reasoning budget, and the temporal scales of the hierarchy are fixed in advance rather than adapted to the event structure of each video. Our end-to-end results additionally depend on a proprietary model: GPT-5 both generates the cold-start SFT traces and serves as the answer agent, so MemLens inherits whatever notion of evidence usefulness the SFT teacher encodes, and although MemLens itself is a 4B open model whose selections are reproducible from the released checkpoint, the reported accuracies are not reproducible without API access. Finally, all three benchmarks are multiple-choice and the design is specialized to that format — the auxiliary answer field, the answer gate, and the calibration reward all assume a discrete option set — so transferring the method to open-ended answering, temporal grounding, or dense captioning would require replacing this outcome signal, which we have not verified.

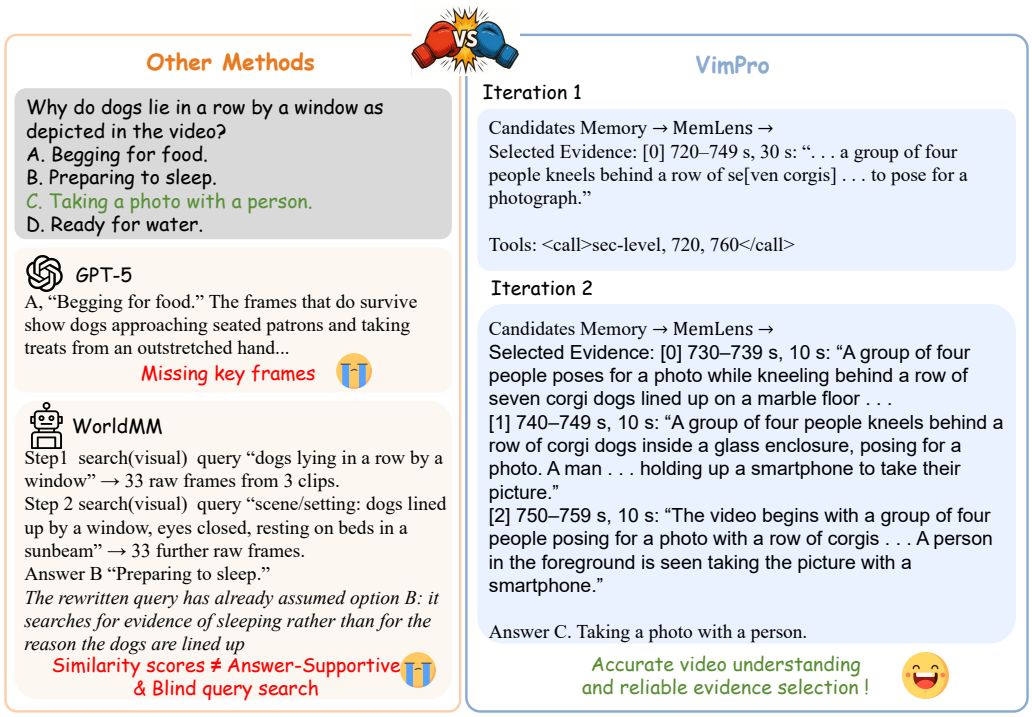


Figure 9: Qualitative comparison on a Video-MME Long question. GPT-5's uniform frame grid misses the staged photo and WorldMM's rewritten query already assumes a distractor, whereas VimPro localizes the span from a coarse memory and confirms it with one directed tool call.

Table 12: Visualization of a question answered from the initial evidence memory alone.

Stage	Content
Question	Why is there no male jaguar near the cub?
Options	A. Male and female jaguars part after mating. B. Male jaguar is dead. C. Female jaguar brings its cub out to find food. D. Male jaguar goes out to find food.
$\mathcal{C}^{\text{txt}}$	**[0]** 480–509 s, 30 s: "A mother jaguar and her cub are shown perched on a tree branch … the narration clarifies that the identity of the cub's father is unknown because **male and female leopards do not remain together after mating**." [1] 1000–1009 s, 10 s: "A mother jaguar and her cub are seen walking along a muddy riverbank … the mother jaguar is facing a new challenge and must hunt for two." **[2]** 360–539 s, 3 min: "A mother jaguar and her six-month-old cub traverse a dirt path and riverbank … narration explains that **male and female leopards do not remain together after mating**, leaving the cub's paternity unknown …" [3] 1170–1199 s, 30 s: "A jaguar emerges from dense vegetation onto a sandy riverbank … determines that the odds of a successful hunt are too low …" [4] 1010–1019 s, 10 s: "… she must hunt for two, a second, smaller jaguar, identified as her cub, is visible in the background …" [5] 2460–2469 s, 10 s: "A jaguar stands over a large carcass … a jaguar cub emerges from the grass, followed shortly by a second cub …" [6] 350–359 s, 10 s: "A jaguar cub lies resting on a sandy patch of ground while an adult jaguar walks past it … jaguar cubs stick with their mothers for the first few years." [7] 470–479 s, 10 s: "A jaguar stands on a thick tree branch … a narrator states that the mother and cubs share the space." … (8 of the 16 candidates shown)
`<think>`	"The question asks why there is no male jaguar near the cub. I need to find memories that explain the absence of the male jaguar. Memory 0 directly addresses this question. It states that male and female leopards (jaguars) do not remain together after mating, and the cub's paternity is unknown because of this behavior. This directly supports option A. Memory 2 also mentions that male and female leopards do not remain together after mating. Memory 0 is the most useful as it explicitly explains the reason for the male's absence (post-mating separation). Other memories discuss mother–cub interactions or hunting but do not address the specific question about the male jaguar's absence."
`<evidence>`	`0,2`
`<answer>`	A (auxiliary)
`<confidence>`	0.95
Answer	A (ground truth A); no tool call was issued

Table 13: Visualization of MemLens discarding a top-ranked but misleading candidate.

Stage	Content
Question	Which state's cuisine is consistently highly praised by the three people in the video?
Options	A. Oklahoma (ground truth) B. Wisconsin C. North Carolina D. Hawaii (the distractor the top-ranked candidate supports)
$\mathcal{C}^{\text{txt}}$	[0] 400–409 s, 10 s: "… a tier list ranking various states, with Wyoming placed in the *A* tier alongside New Mexico and Hawaii. … The scene transitions to a cooked elk chop …" **[1]** 870–879 s, 10 s: "… a cursor hovering over the outline of Wisconsin, which is currently placed in the *S* tier … three people sitting at a table eating tater tots …" [2] 390–419 s, 30 s: "The three men conclude their discussion on the previous food item by assigning it an A tier rating, noting it would be S tier if they were in Hawaii …" [3] 330–339 s, 10 s: "… New Jersey, Minnesota, and West Virginia in the *C* tier, followed by a transition to a map of North Carolina … Lexington-Style BBQ Pork …" [4] 1810–1819 s, 10 s: "… a leaderboard showing California in first place with a score of 19.5 … three golden-brown crab cakes …" [5] 1830–1839 s, 10 s: "… Josh gives a score of 6.0 … a leaderboard graphic showing Maryland in first place …" [6] 1860–1869 s, 10 s: "… the clam chowder tasted familiar … a scoreboard graphic … Maryland in the lead …" **[7]** 870–899 s, 30 s: "… moving Wisconsin from the S tier to the A tier … three people eating tater tots that receive an A grade …" … (candidates [8]–[12], [14], [15] omitted) **[13]** 1800–2031 s, 10 min: "The judges evaluate a series of regional dishes …" – the segment that announces the Fried Onion Burger from **Oklahoma** scoring 9 and 7 and winning
Why retrieval ranks [0] first	It is a tier-list frame naming several states next to a praise-valued tier label, so it matches the question's surface form – states, ranking, praise – more closely than any single tasting segment.
Why it is harmful	The tier shown is a passing intermediate ranking, not the final verdict, and the only state it names in a high tier is the distractor; [2] compounds this by naming Hawaii next to the phrase "S tier".
`<think>`	"… Memory 13 explicitly mentions the Fried Onion Burger from Oklahoma receiving scores of 9 and 7, securing 25 points, and announces Oklahoma as the winner. … Memory 1 and 7 are useful for Wisconsin but focus on tater tots rather than consistent praise of the state's cuisine in the same way Oklahoma is praised for its winning dish. Memory 3 mentions North Carolina but does not describe consistent high praise by all three people. Other memories discuss different states and dishes but do not match the criteria …"
`<evidence>`	`13,1,7` – the rank-1 candidate is absent, and the decisive one is ranked 14 of 16
`<answer>`	A (auxiliary)
`<confidence>`	0.95
Answer with MemLens	A (ground truth A)
Answer without MemLens	D – the same answer agent, given the top-ranked candidates directly, follows the tier-list frames to the distractor

Table 14: Visualization of a tool call recovering second-level evidence.

Stage	Content
Question	What food is sampled by the main character when meeting a modeling agent for the first time in the video?
Options	**A. Some uncommon foods (ground truth)** B. High cholesterol foods C. Historic food D. Roast suckling pig
Round 1 $\mathcal{C}^{\text{txt}}$	**[0]** 1470–1479 s: "… a dish featuring meat and vegetables in a rich orange sauce …" **[1]** 1480–1489 s: "… a large, golden-brown roasted pig served on a banana leaf …" **[2]** 1760–1769 s: "… a food preparation area where whole pigs are being roasted over charcoal …" **[3]** 2160–2339 s, 3 min: "… a man and a woman eating roasted pork outdoors … they are on their second lechon …" **[4]** 1630–1639 s: "… a plate of fried pork topped with pickles …" [5] 970–979 s: "… chopped onions and garlic being added to a hot wok …" **[6]** 2450–2459 s: "… a hand holding a piece of fried pork, revealing its crispy skin …" [7] 180–189 s: "… the interior of a restaurant, panning across walls decorated with framed newspaper clippings …" … (8 of the 16 candidates shown; the omitted [11] and [12] are among those MemLens keeps)
Round 1 output	`<evidence> 0,1,2,3,4,6,11,12; <answer>` D; `<confidence>` 0.6. The `<think>` states the problem outright: "None of the memories explicitly describe the main character meeting a modeling agent … Given the lack of specific evidence … I must conclude that the question might be referring to the prominent food featured in the video (roast suckling pig)."
Agent plan	"I will retrieve the dialogue from the beginning of the video to see how the guest is introduced … and what food is sampled at that first meeting."
Tool call	`<call>sec-level, 0, 180</call>`
Round 2 $\mathcal{C}^{\text{txt}}$	16 second-level candidates inside 0–180 s, including **[2]** 70–79 s: "A woman named **Patricia Lacuesta** is shown eating a piece of fried food at an outdoor food stall … sitting at a metal table across from a man wearing a red bandana …" **[3]** 60–89 s, 30 s: "A man wearing a red bandana and a woman named Patricia Lacuesta sit at a bustling outdoor food stall to sample **golden-brown fried snacks**, where the man … jokes that the over-fried item resembles pork rind …"
Round 2 output	`<evidence> 3,2; <answer>` A; `<confidence>` 0.9. `<think>`: "Memory 2 explicitly mentions Patricia Lacuesta … trying various foods for the first time … Memory 3 … identifies Patricia as the model agent … The fried snacks sampled are considered 'uncommon' …"
Answer	A (ground truth A)
Answer without the tool	B – the single-round variant answers from the round-1 captions alone