VIDEOSPY: EVOLVING THROUGH EXPERIENCE FOR ADAPTIVE VIDEO UNDERSTANDING

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent approaches to agentic video understanding are often tailored to domain-specific evidence requirements and task settings, limiting their adaptability across domains. A central bottleneck is that their task-solving strategies remain largely fixed after design, without a systematic mechanism for learning and evolving from execution experience. In this work, we investigate whether a foundational video agent can evolve from execution experience to adapt to different domains without redesigning its framework. To this end, we first build a foundational VideoSpy agent with three complementary visual tools: `overview`, `clip_skim`, and `frame_inspect`. A Reflector combines task-grouped reflection and evidence-based review to extract reusable experience from training execution records and ground-truth references. An Agent Optimizer loads the resulting *Skill* and source-grounded *Memory* into the shared agent harness, enabling the same foundational agent to specialize independently for each domain. To assess this mechanism, we establish a unified protocol for domain-specific evolution and held-out evaluation on LVBench, EgoLifeQA, and MedVidBench, covering general long videos, egocentric daily life, and medical videos. With Qwen3.5-4B, evolution raises accuracy by 3.8 and 4.2 percentage points on LVBench and EgoLifeQA, improves nine of ten MedVidBench leaderboard metrics, and reduces inference tokens by 9.7–27.1%. These results show that a shared video agent can adapt to different domains by learning from execution experience.

1 INTRODUCTION

Video understanding requires interpreting events, actions, and temporal relationships from visual content and, when available, accompanying language (Huang et al., 2024; Fu et al., 2025). Multimodal models have advanced visual-language alignment (Lin et al., 2024; Bai et al., 2025) and long-video processing (Shen et al., 2025). Agentic video understanding builds on these capabilities to gather evidence and reason through iterative tool use (Wang et al., 2025b; Fan et al., 2025b; Jeoung et al., 2024). A video agent must locate relevant evidence, organize its reasoning, and select appropriate tools, particularly when information is distributed across long sequences (Zhang et al., 2025b; Lin et al., 2026). Its effectiveness therefore depends on both the underlying models and the strategies that guide their use.

These strategies must accommodate diverse settings, including general long videos (Wang et al., 2025a; Wu et al., 2024), egocentric daily-life videos (Yang et al., 2025), and medical or surgical videos (Su et al., 2026; Wang et al., 2026). Different requirements for evidence retrieval, temporal reasoning, and specialized knowledge often motivate domain-specific agent designs (Zhang et al., 2025b; Fan et al., 2025a; Low et al., 2026). Although these agents can adapt their actions within a task, the strategies that guide execution are often fixed by design. This raises a central question: *Can a foundational video agent evolve from execution experience to specialize in different domains without redesigning its framework?*

Our work draws inspiration from animal learning: through trial and error, animals can accumulate memories and acquire skills while their overall body structure remains largely unchanged. A similar distinction between stable structure and adaptive state appears in neural network training (Figure 1(a)), where the architecture is typically fixed while weights are updated using prediction errors on labeled batches. We extend this distinction to video agents in Figure 1(b), with reflection

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.

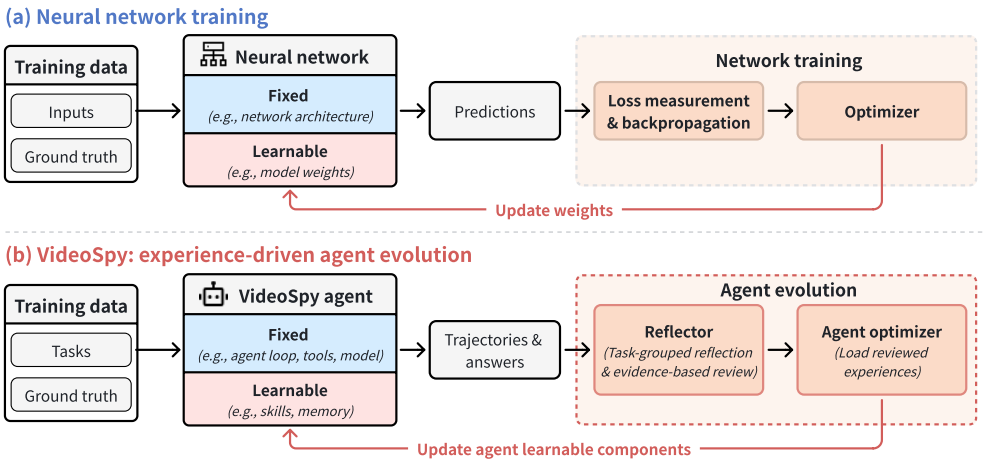


Figure 1: From neural network training to experience-driven agent evolution. (a) Loss measurement and backpropagation guide the optimizer to update network weights within a fixed architecture. (b) The Reflector combines task-grouped reflection with evidence-based review, and the Agent Optimizer loads reviewed experience, enabling domain specialization within a shared agent harness.

on execution experience providing feedback for updates to *Skill* and *Memory*. Model weights and the core *agent harness*, which coordinates the agent loop, tool sets, and context assembly, remain fixed. *Skill* guides how to investigate, while *Memory* supplies source-grounded facts and reusable cases. Ground-truth references support outcome assessment, and recorded observations ground the lessons drawn from each trajectory.

Turning execution experience into strategies that generalize to new tasks requires preserving task context and filtering unsupported lessons. To this end, we introduce **VideoSpy**, a framework for experience-driven agent evolution that enables a common foundational video agent to specialize in different domains. We first build a foundational agent (VideoSpy$^{\mathbf{0}}$) around a ReAct (Yao et al., 2023) loop and three complementary visual tools: `overview`, `clip_skim`, and `frame_inspect`. The **Reflector** performs task-grouped reflection, comparing successful decisions with failure patterns across trajectories of the same task type and using ground-truth references to assess outcomes. Evidence-based review checks whether the extracted guidance and reasoning cases are supported and applicable. The **Agent Optimizer** selects relevant guidance and retrieves eligible facts and cases, integrating reviewed experience into subsequent execution to form the evolved agent (VideoSpy$^{\mathcal{E}}$).

We establish a unified evolution and held-out evaluation protocol on LVBench (Wang et al., 2025a), EgoLifeQA (Yang et al., 2025), and MedVidBench (Su et al., 2026). Each domain evolves independently from the same foundational agent, and evolved agents are frozen before test evaluation. Under matched evaluation conditions, the evolved Qwen3.5-4B agent improves task performance and reduces inference token usage across all three benchmarks. Ablations show that ungrouped reflection can harm accuracy and efficiency, while task-grouped reflection and evidence-based review improve both. Cross-model experiments further support pairing experience matched to the target agent with stronger reflection: a 27B Reflector achieves 46.3% LVBench accuracy using 4B trajectories with inference fixed to 4B.

Our main contributions are: (1) A video-agent evolution framework that adapts a common foundational agent to different domains through execution experience without redesigning its harness. (2) A Reflector that combines task-grouped reflection with evidence-based review, and an Agent Optimizer that integrates the reviewed experience into subsequent task execution. (3) A unified evolution and held-out evaluation protocol across three video domains, demonstrating performance and efficiency gains and assessing cross-model experience transfer.

2 Related Work

Video understanding tasks. Given a video and a question or instruction, video understanding requires reasoning over visual events, temporal relationships, and optional language context to produce answers, descriptions, or spatiotemporal localizations (Li et al., 2024; Ren et al., 2024; Huang et al., 2024; Wang et al., 2026; Fu et al., 2025; Wu et al., 2024). Long-video benchmarks (e.g., LVBench (Wang et al., 2025a)) test evidence retrieval, multimodal integration, and temporal reasoning, while CG-Bench (Chen et al., 2025) explicitly grounds answers in supporting video evidence. Egocentric benchmarks EgoSchema (Mangalam et al., 2023) and EgoLifeQA (Yang et al., 2025) emphasize extended activity understanding and retrieval of personal experiences. Medical benchmarks MedVidBench (Su et al., 2026) and SurgCoT (Wang et al., 2026) additionally require precise localization, procedural reasoning, and domain knowledge. These benchmarks expose different evidence requirements, motivating domain-specific evolution of a common foundational agent.

Agentic video understanding. Large models, including GPT-4o (Hurst et al., 2024), GPT-5 (Singh et al., 2025), and Qwen3.5 (Qwen Team, 2026a), provide perception and reasoning capabilities for video agents. Approaches differ in how they organize and acquire evidence. Decomposed reasoning spans caption-based inference in LLoVi (Zhang et al., 2024a), stepwise reasoning in Video-of-Thought (Fei et al., 2024), and role-based coordination in VideoMind (Liu et al., 2026). Retrieval methods structure evidence as hierarchies, documents, or video collections (Wang et al., 2025c; Ma et al., 2025). Active search methods, including VideoAgent (Wang et al., 2025b), Deep Video Discovery (Zhang et al., 2025b), and VideoSeek (Lin et al., 2026), iteratively select frames or tools. Memory-based agents, including VideoAgent (Fan et al., 2025b), GraphVideoAgent (Chu et al., 2025), M3-Agent (Long et al., 2026), and WorldMM (Yeo et al., 2026), retain structured evidence for retrieval and reasoning. VideoSpy builds on these evidence-gathering capabilities by using training execution records and ground-truth references to learn reusable *Skill* and *Memory* for domain specialization.

Agent reflection and self-improvement. Agent reflection uses execution feedback to diagnose errors and revise subsequent behavior, as in Reflexion (Shinn et al., 2023), ExpeL (Zhao et al., 2024), and Agent-Pro (Zhang et al., 2024c). Self-improvement extends iterative output refinement (Madaan et al., 2023) to persistent context updates in ACE (Zhang et al., 2026b) and optimization of prompts in TextGrad (Yuksekgonul et al., 2025), functions (Zhang et al., 2024b), and agent designs or workflows (Hu et al., 2025; Zhang et al., 2025a). GEPA (Agrawal et al., 2026) uses reflection on execution traces to propose and evaluate prompt updates. Recursive self-improvement (RSI) further targets the improvement mechanism itself. STOP (Zelikman et al., 2024) and Promptbreeder (Fernando et al., 2024) optimize code improvers and prompt-mutation strategies, respectively, while the Darwin Gödel Machine (Zhang et al., 2026a) evolves agents that modify their own code. MetaVideoAgent (Cui et al., 2026) brings automated evolution to video understanding through distribution-aware diagnosis and iterative code optimization. Our work studies domain specialization through reusable *Skill* and *Memory*. Task-grouped reflection and evidence-based review ground this experience in task context and recorded evidence, while the Agent Optimizer integrates it into a shared agent harness.

3 The VideoSpy Evolution Framework

Figure 2 summarizes the framework from domain-specific training tasks to domain-adapted agents. The foundational agent (VideoSpy$^{\mathbf{0}}$) executes each training task once using complementary visual tools, producing *execution records* comprising trajectories, tool observations, and answers. The Reflector analyzes these records alongside ground-truth references to extract and review *Skill* and *Memory*. The Agent Optimizer performs experience selection and context assembly to obtain VideoSpy$^{\mathcal{E}}$.

3.1 Problem Formulation

Given training tasks $\mathcal{D}_{\text{train}} = \{(x_i, y_i^*)\}_{i=1}^{N}$, each input $x_i = (v_i, s_i, q_i)$ contains a video, optional subtitles, and a question. The corresponding y_i^* is a ground-truth reference for the task outcome. The agent receives only x_i during execution and generates its own trajectory, which is subsequently assessed using y_i^* during reflection. We write $\text{VideoSpy}^{\mathbf{0}} = (F, \Theta_0)$, where F comprises the fixed

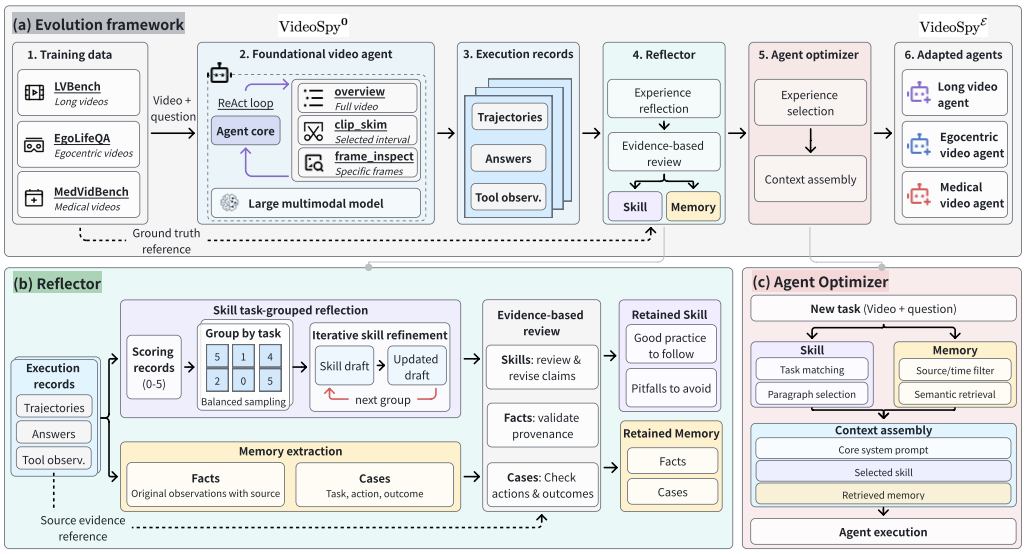


Figure 2: The VideoSpy evolution framework. Training tasks are executed by VideoSpy$^{\mathbf{0}}$ with complementary visual tools to collect execution records. The Reflector extracts and verifies *Skill* and *Memory*, and the Agent Optimizer loads them into the prompt to obtain domain-adapted agents. Each domain evolves independently from the same foundational agent.

model and core agent harness, including the agent loop, prompt assembly, and tool interfaces, and $\Theta_0 = (S_0, M_0)$ is its initial *Skill* and *Memory* state. The complete adaptation process is

$$(\hat{y}_i, \tau_i) = \text{VideoSpy}^{\mathbf{0}}(x_i), \qquad i = 1, \ldots, N, \tag{1}$$

$$\mathcal{T} = \{(x_i, \hat{y}_i, \tau_i, y_i^*)\}_{i=1}^{N}, \tag{2}$$

$$(S^*, M^*) = R(\mathcal{T}; S_0, M_0), \tag{3}$$

$$\text{VideoSpy}^{\mathcal{E}} = U(\text{VideoSpy}^{\mathbf{0}}; S^*, M^*). \tag{4}$$

Here, $\hat{y}_i$ is the predicted answer and τ_i is the execution record for task i. $\mathcal{T}$ collects these records together with the task inputs, predictions, and ground-truth references. R denotes the Reflector, which performs reflection and review over $\mathcal{T}$, and U denotes the Agent Optimizer, which loads the resulting experience into VideoSpy$^{\mathbf{0}}$ to obtain VideoSpy$^{\mathcal{E}}$.

3.2 Foundational Video Agent

The foundational VideoSpy$^{\mathbf{0}}$ agent provides the evidence-gathering and reasoning capabilities on which evolution builds. A large multimodal model follows a ReAct loop to collect visual evidence through three complementary tools: `overview` samples the full video, `clip_skim` examines a selected interval, and `frame_inspect` checks original-resolution frames at specified timestamps. Each tool takes a text query to guide visual inspection, with an explicit time interval for `clip_skim` and timestamps for `frame_inspect`. Let h_t denote the execution history before step t, initialized with the agent prompt and task input. Each evidence-gathering step follows

$$a_t \sim \pi_F(\cdot \mid h_t), \qquad o_t = \text{Exec}(a_t; v), \qquad h_{t+1} = h_t \oplus (a_t, o_t), \tag{5}$$

where π_F is the fixed model's tool-use policy, a_t contains the selected tool calls and their arguments, o_t contains the returned observations, and $\oplus$ denotes appending to the history. When evidence collection ends or the step limit is reached, a separate model call without tools produces the final answer from the accumulated history. For task i, the tool calls and observations across multiple rounds, together with the final answer, form the execution record τ_i used for reflection.

The same agent harness supports all target domains. Evolution from scratch starts with $S_0 = M_0 = \emptyset$. Evolution from VideoSpy$^{\mathbf{0}}$ to VideoSpy$^{\mathcal{E}}$ changes the loaded *Skill* and *Memory* content while keeping model weights, tool interfaces, and configured execution hyperparameters fixed.

3.3 REFLECTOR: TASK-GROUPED REFLECTION AND EVIDENCE-BASED REVIEW

The Reflector converts fixed execution records into *Skill* and *Memory* through task-grouped reflection, memory extraction, and evidence-based review (Figure 2(b)). *Skill* summarizes good practices to follow and pitfalls to avoid, while *Memory* retains reusable facts and cases with sources.

Task-grouped reflection. The Reflector assigns each trajectory a score from 0 to 5 based on answer correctness, reasoning soundness, evidence support, and execution efficiency. Within each task type (e.g., region captioning in MedVidBench (Su et al., 2026)), a bounded sample balances the available score levels and is divided into groups mixing higher- and lower-scoring trajectories when available, without reuse. Each group includes the original trajectories, predictions, scores, and assessment reasons.

For task type k, the Reflector processes groups $G_{k,1}, \ldots, G_{k,J_k}$ sequentially, maintaining one working *Skill* document:

$$W_{k,j} = \mathrm{Reflect}(W_{k,j-1}, G_{k,j}), \qquad j = 1, \ldots, J_k, \tag{6}$$

where $W_{k,0}$ is the initial guidance or empty when evolving from scratch. Each group refines the draft by comparing effective practices and reasoning failures, preserving supported prior guidance while adding, qualifying, or removing claims as warranted by the evidence. The updated draft passes to the next group and remains linked to its sources. Without a substantive update, $W_{k,j} = W_{k,j-1}$.

Memory extraction. *Memory* comprises facts and cases, extracted independently of *Skill* sampling. Facts preserve original tool observations with their sources for reuse within the same video. Cases capture tasks, actions, and outcomes from successful, evidence-supported trajectories, focusing on how an action resolved a relevant uncertainty. They provide methodological examples that can transfer across videos.

Evidence-based review. After all groups are processed, *Skill* is reviewed against its cited decisions and observations:

$$S_k^* = \mathrm{Review}(W_{k,J_k}, E_k), \tag{7}$$

where E_k contains the cited evidence in its trajectory context. Review accepts cautions grounded in observed failures but rejects claims that untried remedies are effective. Unsupported claims, overgeneralizations, and scene-specific answers are removed or revised while preserving supported content. The retained content forms S^*.

For *Memory*, facts are checked against their source observations, while cases are reviewed against recorded actions and outcomes. These checks assess recorded evidence without re-inspecting the video.

3.4 AGENT OPTIMIZER: INTEGRATING EXPERIENCE INTO EXECUTION

The Agent Optimizer converts the Reflector's outputs into the inference context of VideoSpy$^{\mathbf{0}}$, yielding VideoSpy$^{\mathcal{E}}$. As shown in Figure 2(c), experience selection combines task matching and paragraph selection for *Skill* with source/time filtering and semantic retrieval for *Memory*. Context assembly appends the selected experience to the core system prompt before execution.

Skill loading. For a new task, the agent selects the matching *Skill* content. The selected content is appended to the core system prompt in complete paragraphs. This guidance helps direct evidence gathering and judge when to answer, while the current question, observations, and tool constraints remain authoritative.

Memory retrieval. Facts and reviewed cases are deduplicated within compatible sources, embedded, and stored with their provenance. Before a task starts, retrieval filters entries by source and temporal eligibility, ranks the eligible entries by semantic similarity to the query, and appends selected top-k (3 by default) entries to the system prompt. Factual observations apply to their source video or timeline. Historical cases can transfer across videos within the same benchmark, but their scene facts do not describe the current video. Current observations take precedence, and memories unavailable at the query time are excluded.

Table 1: Datasets, evaluation splits, and metrics. MCQ: multiple-choice questions.

Benchmark	Tasks	Training	Test	Main metrics
LVBench (Wang et al., 2025a)	Long-video MCQ (6 types)	83 videos / 1,234 questions	20 videos / 315 questions	Accuracy
EgoLifeQA (Yang et al., 2025)	Egocentric video MCQ (5 types)	Days 1–5 / 405 questions	Days 6–7 / 95 questions	Accuracy
MedVidBench (Su et al., 2026)	Medical video tasks (8 types)	5,146 tasks	Local: 1,124 tasks Leaderboard: 6,245 tasks	10 official task metrics

Writing $e(m, x) \in \{0, 1\}$ for the source and temporal eligibility of memory entry m for input x, the retrieval candidate set is $M_x = \{m \in M^* : e(m, x) = 1\}$. With a fixed prompt assembly function Γ, the resulting initial prompt is

$$P(x) = \Gamma\big(P_{\text{core}}, \text{MatchSkill}(S^*, x), \text{RetrieveMemory}(M_x, x)\big). \tag{8}$$

Here, RetrieveMemory ranks eligible entries by semantic similarity and selects content within the memory budget. Both components are loaded before task execution. VideoSpy$^{\mathcal{E}}$ then follows the same reasoning and tool-use loop as VideoSpy$^{\mathbf{0}}$, guided by the learned experience.

3.5 Evolution and Evaluation Protocol

Our protocol separates experience collection and evolution from held-out evaluation. Each domain starts from the same VideoSpy$^{\mathbf{0}}$. For domain d, let $\mathcal{T}^{(d)}_{\text{train}}$ contain only the execution records and ground-truth references from its training split. Each domain learns its own state while retaining the shared structure:

$$\Theta^{(d)} = R(\mathcal{T}^{(d)}_{\text{train}}; S_0, M_0), \qquad \text{VideoSpy}^{\mathcal{E}}_d = U(\text{VideoSpy}^{\mathbf{0}}; S^{(d)}, M^{(d)}), \tag{9}$$

where $\Theta^{(d)} = (S^{(d)}, M^{(d)})$. This state is frozen before evaluation on a test set separated from training by video, date, or source video.

4 Experiments

4.1 Experimental Setup

Datasets and metrics. To evaluate experience-driven evolution, we independently evolve the same foundational VideoSpy$^{\mathbf{0}}$ for each domain using execution records and ground-truth references from training tasks. The Reflector extracts reusable experience, which the Agent Optimizer loads to form VideoSpy$^{\mathcal{E}}$. We then freeze the evolved state and compare the two agents on held-out tasks. We evaluate on LVBench (Wang et al., 2025a) for long-video evidence reasoning, EgoLifeQA (Yang et al., 2025) for egocentric daily activities, and MedVidBench (Su et al., 2026) for medical videos. Table 1 summarizes the training and test splits. We measure evolution gains using accuracy on LVBench and EgoLifeQA and the ten official task metrics on MedVidBench. We also report average inference rounds and token costs per task.

Baselines. We compare with DVDAgent (Zhang et al., 2025b), VideoSeek (Lin et al., 2026), and WorldMM (Yeo et al., 2026) that are primarily designed for long-video understanding. WorldMM further supports ultra-long egocentric video streams in EgoLifeQA through episodic, semantic, and visual memory. DVDAgent and WorldMM require offline construction of a video text index, whereas VideoSeek and our VideoSpy$^{\mathbf{0}}$ gather evidence directly from raw video. For the MedVidBench leaderboard test, we additionally include publicly reported results including Qwen2.5-VL-7B$_{\text{Surg-CholecT50}}$ (NVIDIA, 2025) specialized for surgical video understanding.

Implementation details. We use Qwen3.5-4B (Qwen Team, 2026a) and Qwen3.8-27B (Qwen Team, 2026b), abbreviated as 4B and 27B in tables. Within each execution, the agent and visual tools share the same backbone. The Reflector uses the same backbone by default, except in the cross-model experiments. We reproduce DVDAgent, WorldMM, and VideoSeek on our test splits. For

Table 2: Ablations of the evolution framework with Qwen3.5-4B. Checkmarks indicate enabled components. Bold marks the best result in each metric column.

Configuration	Reflector		Agent Optimizer		LVBench			EgoLifeQA		
	Task-grouped reflection	Evidence-based review	Skill	Memory	Acc. (%) ↑	Avg. rounds ↓	Avg. tokens ↓	Acc. (%) ↑	Avg. rounds ↓	Avg. tokens ↓
VideoSpy$^{\mathbf{0}}$	–	–	–	–	39.4	11.7	80.3k	32.6	14.7	129.0k
(a) Reflector										
Skill only (ungrouped)	–	✓	✓	–	36.5	12.4	83.2k	31.6	14.9	135.5k
Skill+Memory (no review)	✓	–	✓	✓	39.7	12.8	84.5k	33.7	15.2	139.3k
(b) Agent Optimizer										
Skill only	✓	✓	✓	–	41.3	9.8	74.2k	34.7	12.2	121.5k
Memory only	✓	✓	–	✓	40.3	10.9	77.7k	34.7	13.1	129.8k
VideoSpy$^{\mathcal{E}}$	✓	✓	✓	✓	**43.2**	**9.3**	**69.7k**	**36.8**	**11.7**	**116.5k**

Table 3: Accuracy and inference cost on LVBench and EgoLifeQA. Bold marks the best value in each metric column within the online group (b). * denotes reproduced baselines (Avg. tokens cost per task for prebuilt indices (LVBench/EgoLifeQA): DVDAgent 14.1k/23.9k (reused GPT-4.1-generated LVBench captions), WorldMM 405.7k/reused).

Method	Model	LVBench			EgoLifeQA		
		Acc. (%) ↑	Avg. rounds ↓	Avg. tokens ↓	Acc. (%) ↑	Avg. rounds ↓	Avg. tokens ↓
(a) Offline caption or memory construction (requires a prebuilt text index)							
DVDAgent* (Zhang et al., 2025b)	4B	55.9	10.6	437.5k	33.7	14.1	104.3k
WorldMM* (Yeo et al., 2026)	4B	36.5	2.9	39.1k	37.9	2.7	25.8k
(b) Online active evidence gathering (raw video, no offline index)							
VideoSeek* (Lin et al., 2026)	4B	33.7	10.9	244.9k	26.3	13.3	202.5k
VideoSpy$^{\mathbf{0}}$ *(ours)*	4B	39.4	11.7	80.3k	32.6	14.7	129.0k
VideoSpy$^{\mathcal{E}}$ *(ours)*	4B	43.2	**9.3**	**69.7k**	36.8	11.7	116.5k
VideoSpy$^{\mathbf{0}}$ *(ours)*	27B	56.8	11.0	86.2k	41.1	12.1	102.5k
VideoSpy$^{\mathcal{E}}$ *(ours)*	27B	**60.6**	10.2	71.1k	**42.1**	**11.2**	**100.2k**

DVDAgent on LVBench, we reuse existing GPT-4.1-generated offline captions. Across VideoSpy and all reproduced baselines, we use Qwen *text-embedding-v4* (1,024 dimensions) wherever text embeddings are required for indexing and retrieval. VideoSpy uses a default limit of 16 rounds, after which it must produce a final answer without further tool calls. On the MedVidBench local test split, we consistently use Qwen3.5-4B as the judge for all LLM-based metrics.

4.2 Ablation Studies

Reflector. Table 2 shows that *Skill* from ungrouped reflection lowers accuracy and increases token usage relative to VideoSpy$^{\mathbf{0}}$, suggesting that extracted guidance can be counterproductive. With review and *Skill*-only loading fixed, task-grouped reflection raises accuracy from 36.5% to 41.3% on LVBench and from 31.6% to 34.7% on EgoLifeQA, while reducing token usage and outperforming the foundational agent on both measures. With task grouping and *Skill+Memory* loading fixed, reviewing *Skill* and cases improves accuracy by 3.5 and 3.1 percentage points and reduces token usage by 17.5% and 16.4% on the two benchmarks, respectively. These results highlight the importance of organizing and verifying experience before reuse.

Agent Optimizer. With both Reflector components enabled, *Skill* and *Memory* each improve accuracy over VideoSpy$^{\mathbf{0}}$, while combining them achieves the highest accuracy with the fewest rounds and tokens on both benchmarks (Table 2(b)). This supports their complementary roles: *Skill* guides how to investigate, while *Memory* supplies reusable facts and cases.

4.3 Performance and Efficiency across Domains

We assess whether domain-specific experience improves held-out performance and inference efficiency within a shared agent harness. Each evolved agent is compared with its foundational counterpart under fixed models, tools, and execution budgets. Tables 3 and 4 report these comparisons alongside reproduced baselines and published leaderboard results.

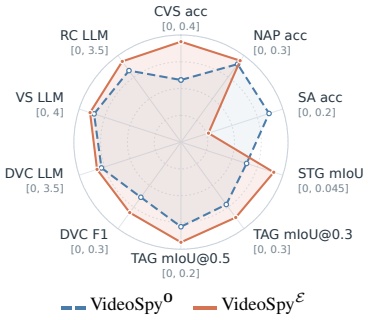
Table 4: MedVidBench results on local and leaderboard splits. All leaderboard baselines use 1-shot prompting. Bold marks the best value in each metric column within each split. * denotes reproduced baselines (Avg. tokens cost per task for prebuilt indices: DVDAgent 183.2k, WorldMM 174.5k.)

Method	Model	CVS acc	NAP acc	SA acc	STG mIoU	TAG_{mIoU}@0.3	TAG_{mIoU}@0.5	DVC F1	DVC llm	VS llm	RC llm
(a) Local test split (1,124 samples).											
DVDAgent* (Zhang et al., 2025b)	4B	0.205	0.169	0.080	0.031	0.027	0.024	0.121	2.184	3.925	2.708
WorldMM* (Yeo et al., 2026)	4B	0.732	0.202	0.204	0.011	0.028	0.012	0.156	1.790	2.760	1.640
VideoSeek* (Lin et al., 2026)	4B	0.289	0.274	**0.225**	0.033	0.185	0.157	0.184	2.061	1.754	2.204
VideoSpy$^{\mathbf{O}}$ *(ours)*	4B	0.393	0.250	0.000	0.049	0.232	0.177	0.227	1.835	3.036	3.044
VideoSpy$^{\mathcal{E}}$ *(ours)*	4B	0.581	**0.325**	0.000	0.036	0.212	0.166	0.232	1.972	3.224	3.422
VideoSpy$^{\mathbf{O}}$ *(ours)*	27B	0.257	0.298	0.056	0.071	0.314	0.266	0.337	2.487	4.031	3.246
VideoSpy$^{\mathcal{E}}$ *(ours)*	27B	**0.794**	0.226	0.142	**0.074**	**0.324**	**0.267**	**0.368**	**2.688**	**4.085**	**3.589**
(b) Leaderboard split (6,245 samples)											
Qwen2.5-VL-7B$_{\text{Surg-CholecT50}}$ (NVIDIA, 2025)	7B	0.000	**0.302**	0.000	0.000	0.019	0.013	0.051	1.945	2.101	2.986
1-shot prompting (Su et al., 2026)	GPT-4.1	0.018	0.250	0.087	0.014	0.096	0.005	0.101	2.438	2.490	2.080
1-shot prompting (Su et al., 2026)	4B	0.309	0.231	**0.276**	**0.051**	0.074	0.040	0.142	2.699	3.491	3.037
VideoSpy$^{\mathbf{O}}$ *(ours)*	4B	0.232	0.270	0.173	0.029	0.216	0.158	0.191	2.731	3.411	2.888
VideoSpy$^{\mathcal{E}}$ *(ours)*	4B	**0.375**	0.282	0.054	0.041	**0.261**	**0.187**	**0.244**	**2.883**	**3.569**	**3.260**

Table 5: Cross-model evolution on LVBench. Models for trajectory collection and reflection vary independently, with inference fixed to Qwen3.5-4B.

Evolution models		LVBench		
Trajectory collection	Reflector	Acc. (%) ↑	Avg. rounds ↓	Avg. tokens ↓
4B	4B	43.2	**9.3**	69.7k
4B	27B	**46.3**	9.6	**68.9k**
27B	4B	40.3	10.9	72.6k
27B	27B	42.9	10.5	71.5k

Figure 3: Qwen3.5-4B on MedVidBench leaderboard test, with metric-specific scales.



Long-video and egocentric understanding. Evolution improves accuracy while reducing token usage on both benchmarks with both backbones (Table 3), supporting more effective evidence use through learned experience. LVBench gains around 4 percentage points at both 4B and 27B, showing that experience remains useful with a stronger backbone. EgoLifeQA gains are 4.2 and 1.0 points, respectively, indicating that the benefit varies with the domain and backbone. These improvements support adapting evidence-gathering strategies within a shared harness. Under the matched 4B backbone, VideoSpy$^{\mathcal{E}}$ also outperforms VideoSeek in accuracy and token usage. DVDAgent and WorldMM provide reference results under a different setting with prebuilt offline indices.

Medical video understanding. On the local MedVidBench split, gains in CVS (Critical View of Safety) scoring, captioning, and summarization across both backbones support adaptation to medical tasks within the shared harness (Table 4). On the leaderboard test, 4B evolution improves nine of ten metrics (Figure 3) while reducing average token usage by 27.1% (see Appendix), demonstrating more accurate and efficient specialization without changing model weights or tools. The evolved 4B agent exceeds same-backbone one-shot prompting on eight metrics, and GPT-4.1 and the surgical Qwen2.5-VL-7B$_{\text{Surg-CholecT50}}$ model on nine each.

These baselines use one-shot prompting with a task-specific question–answer format example (Su et al., 2026). VideoSpy uses no manually supplied question–answer demonstration and instead uses frozen *Skill* and *Memory* to guide iterative evidence gathering. The gains across most metrics support the effectiveness of experience-guided investigation, with task-specific exceptions: some metrics favor one-shot baselines, and skill assessment declines after evolution. Skill assessment requires judging execution quality, while grounding depends on precise spatial and temporal boundaries (Su et al., 2026). MedVidBench's fixed pre-extracted frames may omit cues needed for these judgments, constraining the benefits of learned guidance.

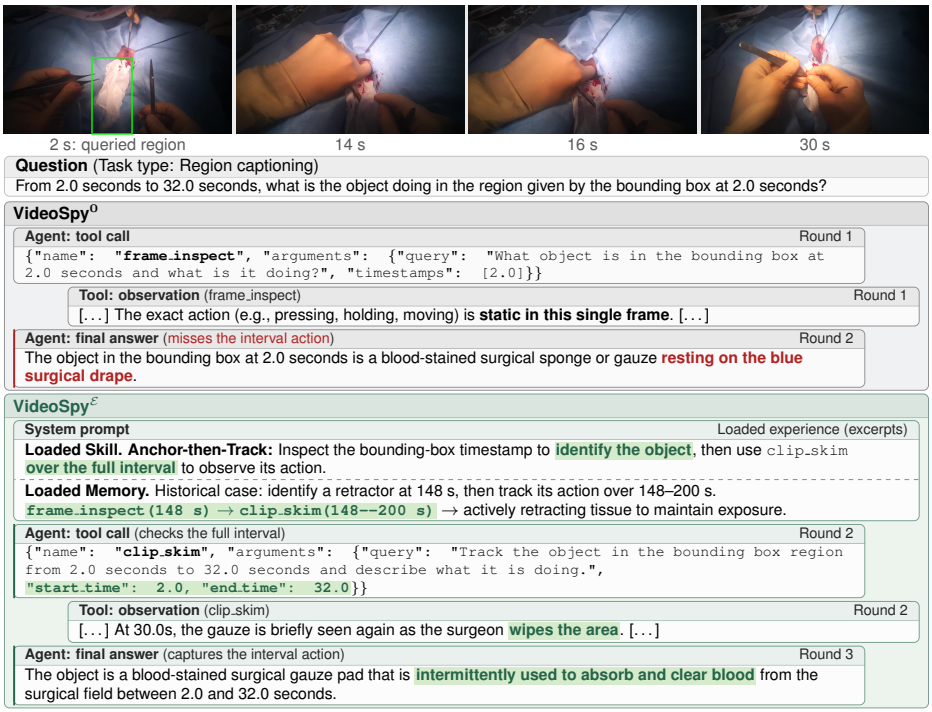


Figure 4: **Experience-guided evidence gathering.** Skill and Memory guide VideoSpy$^{\mathcal{E}}$ to inspect the full 2–32 s interval and identify the gauze's blood-clearing action, which VideoSpy$^{\mathbf{0}}$ misses from a single frame. Full case: Appendix E.3.

4.4 Experience Transfer and Behavioral Adaptation

Cross-model experience transfer. Table 5 fixes inference to Qwen3.5-4B and varies the execution backbone and Reflector independently. Experience from 27B trajectories reaches 40.3–42.9% accuracy, exceeding the 39.4% foundational baseline and demonstrating transfer across these backbones. However, 4B trajectories perform better with either Reflector, suggesting the value of experience matched to the target agent's behavior. A 27B Reflector improves both trajectory sources, with the best pairing reaching 46.3% using 4B trajectories. These results suggest complementary roles for experience matched to the target agent and stronger reflection.

Experience-guided evidence gathering. Figure 4 illustrates how experience aligns evidence collection with the question's temporal scope on a MedVidBench sample. VideoSpy$^{\mathbf{0}}$ answers an action question from a single frame, describing only where the gauze rests. The loaded *Skill* specifies identifying the object before inspecting its action over the requested interval, while a historical case in *Memory* provides a concrete example. VideoSpy$^{\mathcal{E}}$ additionally inspects 2–32 s and identifies the gauze's blood-clearing function, matching the reference. The key change is to verify the requested action before stopping, using the same backbone and tools.

5 Conclusion

We presented VideoSpy, a framework that enables a common foundational video agent to specialize in different domains through execution experience. The Reflector combines task-grouped reflection and evidence-based review to extract reusable *Skill* and *Memory*, which the Agent Optimizer integrates into subsequent execution. Under a unified held-out evaluation protocol, the evolved Qwen3.5-4B agent improves task performance and reduces inference token usage on LVBench, EgoLifeQA, and MedVidBench. Ablations show that ungrouped reflection can hurt performance, highlighting the value of task-grouped reflection and evidence-based review. Cross-model results further suggest that experience matched to the target agent complements stronger reflection. These findings support experience-driven domain adaptation within a shared video agent.

AI USE STATEMENT

In this work, we used generative AI tools to assist with code implementation, grammar correction, formatting, and improving the clarity of writing. We have not used generative AI tools for generating research ideas, designing experiments, or creating experimental results. We have reviewed all AI-assisted work. Specifically, we checked AI-assisted text for accuracy and consistency with the intended meaning. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

Our experiments use only publicly available datasets: LVBench, EgoLifeQA, and MedVidBench. We follow the applicable licenses, terms of use, and access requirements of each dataset, including restrictions concerning privacy and redistribution. These datasets are used solely for the research purposes described in this paper.

REPRODUCIBILITY STATEMENT

To support reproducibility, Sections 3.5 and 4.1 describe our evolution and evaluation protocols and experimental settings. Appendices A–C provide detailed method descriptions, core excerpts of the original prompts, complete visual-tool schemas, and Skill and Memory loading procedures. We provide the project source code in the supplementary materials. All datasets are publicly available, and our agent backbones use open weights. Reused captions and external embedding services are described in Section 4.1.

REFERENCES

Lakshya A Agrawal, Shangyin Tan, Dilara Soylu, Noah Ziems, Rishi Khare, Krista Opsahl-Ong, et al. GEPA: Reflective prompt evolution can outperform reinforcement learning. In *International Conference on Learning Representations*, 2026.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, et al. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Guo Chen, Yicheng Liu, Yifei Huang, Baoqi Pei, Jilan Xu, Yuping He, et al. CG-Bench: Clue-grounded question answering benchmark for long video understanding. In *International Conference on Learning Representations*, 2025.

Meng Chu, Yicong Li, and Tat-Seng Chua. GraphVideoAgent: Enhancing long-form video understanding with entity relation graphs. In *ACM International Conference on Multimedia*, 2025.

Benlei Cui, Ruize Wang, Junjie Li, Jinhao Chen, Longtao Huang, Yinghao Chen, et al. MetaVideoAgent: Automated video-agent evolution for long-form video understanding. *arXiv preprint arXiv:2608.04587*, 2026.

Yue Fan, Xiaojian Ma, Rongpeng Su, Jun Guo, Rujie Wu, Xi Chen, and Qing Li. Embodied videoagent: Persistent memory from egocentric videos and embodied sensors enables dynamic scene understanding. In *IEEE/CVF International Conference on Computer Vision*, 2025a.

Yue Fan, Xiaojian Ma, Rujie Wu, Yuntao Du, Jiaqi Li, Zhi Gao, et al. VideoAgent: A memory-augmented multimodal agent for video understanding. In *European Conference on Computer Vision*, 2025b.

Hao Fei, Shengqiong Wu, Wei Ji, Hanwang Zhang, Meishan Zhang, Mong-Li Lee, et al. Video-of-Thought: Step-by-step video reasoning from perception to cognition. In *International Conference on Machine Learning*, 2024.

Chrisantha Fernando, Dylan Banarse, Henryk Michalewski, Simon Osindero, and Tim Rocktäschel. Promptbreeder: Self-referential self-improvement via prompt evolution. In *International Conference on Machine Learning*, 2024.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, et al. Video-MME: The first-ever comprehensive evaluation benchmark of multi-modal LLMs in video analysis. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Shengran Hu, Cong Lu, and Jeff Clune. Automated design of agentic systems. In *International Conference on Learning Representations*, 2025.

De-An Huang, Shijia Liao, Subhashree Radhakrishnan, Hongxu Yin, Pavlo Molchanov, Zhiding Yu, et al. LITA: Language instructed temporal-localization assistant. In *European Conference on Computer Vision*, 2024.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, et al. GPT-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

Sullam Jeoung, Goeric Huybrechts, Bhavana Ganesh, Aram Galstyan, and Sravan Bodapati. Adaptive video understanding agent: Enhancing efficiency with dynamic frame sampling and feedback-driven reasoning. In *NeurIPS 2024 Multimodal Algorithmic Reasoning Workshop*, 2024.

Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, et al. MVBench: A comprehensive multi-modal video understanding benchmark. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Bin Lin, Yang Ye, Bin Zhu, Jiaxi Cui, Munan Ning, Peng Jin, et al. Video-LLaVA: Learning united visual representation by alignment before projection. In *Conference on Empirical Methods in Natural Language Processing*, 2024.

Jingyang Lin, Jialian Wu, Jiang Liu, Ximeng Sun, Ze Wang, Xiaodong Yu, et al. VideoSeek: Long-horizon video agent with tool-guided seeking. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

Ye Liu, Kevin Qinghong Lin, Chang-Wen Chen, and Mike Zheng Shou. VideoMind: A chain-of-LoRA agent for temporal-grounded video reasoning. In *International Conference on Learning Representations*, 2026.

Lin Long, Yichen He, Wentao Ye, Yiyuan Pan, Yuan Lin, Hang Li, et al. Seeing, listening, remembering, and reasoning: A multimodal agent with long-term memory. In *International Conference on Learning Representations*, 2026.

Chang Han Low, Ziyue Wang, Tianyi Zhang, Zhu Zhuo, Zhitao Zeng, Evangelos B Mazomenos, and Yueming Jin. Surgraw: Multi-agent workflow with chain of thought reasoning for robotic surgical video analysis. *IEEE Robotics and Automation Letters*, 2026.

Ziyu Ma, Chenhui Gou, Hengcan Shi, Bin Sun, Shutao Li, Hamid Rezatofighi, et al. DrVideo: Document retrieval based long video understanding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, et al. Self-Refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems*, 2023.

Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. EgoSchema: A diagnostic benchmark for very long-form video language understanding. In *Advances in Neural Information Processing Systems*, 2023.

NVIDIA. Qwen2.5-VL-7B-Surg-CholecT50. Hugging Face model card, 2025. URL `https://huggingface.co/nvidia/Qwen2.5-VL-7B-Surg-CholecT50`.

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

Qwen Team. Qwen3.8-27B. Model card, 2026b. URL `https://huggingface.co/Qwen/Qwen3.8-27B`.

Shuhuai Ren, Linli Yao, Shicheng Li, Xu Sun, and Lu Hou. TimeChat: A time-sensitive multimodal large language model for long video understanding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Xiaoqian Shen, Yunyang Xiong, Changsheng Zhao, Lemeng Wu, Jun Chen, Chenchen Zhu, et al. LongVU: Spatiotemporal adaptive compression for long video-language understanding. In *International Conference on Machine Learning*, 2025.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, 2023.

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, et al. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*, 2025.

Yuhao Su, Anwesa Choudhuri, Zhongpai Gao, Benjamin Planche, Van Nguyen Nguyen, Meng Zheng, et al. MedGRPO: Multi-task reinforcement learning for heterogeneous medical video understanding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

Gui Wang, YongSong Zhou, Kaijun Deng, Wooi Ping Cheah, Rong Qu, Jianfeng Ren, et al. SurgCoT: Advancing spatiotemporal reasoning in surgical videos through a chain-of-thought benchmark. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

Weihan Wang, Zehai He, Wenyi Hong, Yean Cheng, Xiaohan Zhang, Ji Qi, et al. LVBench: An extreme long video understanding benchmark. In *IEEE/CVF International Conference on Computer Vision*, 2025a.

Xiaohan Wang, Yuhui Zhang, Orr Zohar, and Serena Yeung-Levy. VideoAgent: Long-form video understanding with large language model as agent. In *European Conference on Computer Vision*, 2025b.

Ziyang Wang, Shoubin Yu, Elias Stengel-Eskin, Jaehong Yoon, Feng Cheng, Gedas Bertasius, et al. VideoTree: Adaptive tree-based video representation for LLM reasoning on long videos. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025c.

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*, 2024.

Jingkang Yang, Shuai Liu, Hongming Guo, Yuhao Dong, Xiamengwei Zhang, Sicheng Zhang, et al. EgoLife: Towards egocentric life assistant. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, et al. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023.

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

Mert Yuksekgonul, Federico Bianchi, Joseph Boen, Sheng Liu, Pan Lu, Zhi Huang, et al. Optimizing generative AI by backpropagating language model feedback. *Nature*, 2025.

Eric Zelikman, Eliana Lorch, Lester Mackey, and Adam Tauman Kalai. Self-taught optimizer (STOP): Recursively self-improving code generation. In *Conference on Language Modeling*, 2024.

Ce Zhang, Taixi Lu, Md Mohaiminul Islam, Ziyang Wang, Shoubin Yu, Mohit Bansal, et al. A simple LLM framework for long-range video question-answering. In *Conference on Empirical Methods in Natural Language Processing*, 2024a.

Jenny Zhang, Shengran Hu, Cong Lu, Robert Lange, and Jeff Clune. Darwin gödel machine: Open-ended evolution of self-improving agents. In *International Conference on Learning Representations*, 2026a.

Jiayi Zhang, Jinyu Xiang, Zhaoyang Yu, Fengwei Teng, Xionghui Chen, Jiaqi Chen, et al. AFlow: Automating agentic workflow generation. In *International Conference on Learning Representations*, 2025a.

Qizheng Zhang, Changran Hu, Shubhangi Upasani, Boyuan Ma, Fenglu Hong, Vamsidhar Kamanuru, et al. Agentic context engineering: Evolving contexts for self-improving language models. In *International Conference on Learning Representations*, 2026b.

Shaokun Zhang, Jieyu Zhang, Jiale Liu, Linxin Song, Chi Wang, Ranjay Krishna, et al. Offline training of language model agents with functions as learnable weights. In *International Conference on Machine Learning*, 2024b.

Wenqi Zhang, Ke Tang, Hai Wu, Mengna Wang, Yongliang Shen, Guiyang Hou, et al. Agent-Pro: Learning to evolve via policy-level reflection and optimization. In *Annual Meeting of the Association for Computational Linguistics*, 2024c.

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

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. ExpeL: LLM agents are experiential learners. In *AAAI Conference on Artificial Intelligence*, 2024.

Appendix

A Foundational Agent: Tools and Core Instructions

A.1 Visual-Tool Ablations

Table S1 ablates the visual tools of VideoSpy^{0} before loading Skill or Memory, supplementing the foundational agent (Section 3.2) and ablation study (Section 4.2).

Table S1: Visual-tool configurations of the foundational agent with Qwen3.5-4B, before evolution. Checkmarks indicate available tools; – indicates a disabled tool. All configurations have no loaded Skill or Memory. Bold marks the best value in each metric column.

Configuration	Available visual tools			LVBench			EgoLifeQA		
	`overview`	`clip_skim`	`frame_inspect`	Acc. (%) ↑	Avg. rounds ↓	Avg. tokens ↓	Acc. (%) ↑	Avg. rounds ↓	Avg. tokens ↓
`overview` only	✓	–	–	33.7	**6.1**	**44.6k**	26.3	**8.5**	**100.4k**
Without `frame_inspect`	✓	✓	–	36.8	9.5	62.9k	30.5	12.3	115.2k
Without `clip_skim`	✓	–	✓	37.5	10.8	69.7k	29.5	13.7	134.8k
VideoSpy^{0} (full tools)	✓	✓	✓	**39.4**	11.7	80.3k	**32.6**	14.7	129.0k

The full tool set achieves the highest accuracy on both benchmarks. Overview alone uses fewer rounds and tokens but loses 5.7 points on LVBench and 6.3 on EgoLifeQA. Removing either local tool also reduces accuracy. The tools provide complementary functions: overview establishes global context, clip exploration captures actions and temporal changes, and frame inspection resolves fine visual details. Their consistent benefits across long-video and egocentric tasks support VideoSpy^{0} as a general-purpose foundation for domain-specific evolution with a shared tool interface.

A.2 Core System Prompt and Tool Schemas

Given the question, video duration, and subtitles, the agent gathers evidence for up to 16 rounds before a final call without tools. The boxes in Appendices A and B quote core excerpts of the original prompts. Wording is unchanged, with `[...]` marking omissions and formatting adjusted for readability.

Foundational agent: original system prompt excerpts

Role

You are an efficient video-understanding agent. Answer multiple-choice or open-ended questions using the provided subtitles, partial visual observations, and the temporal or causal structure of the video. Do not invent video content.

Interaction Protocol

The conversation history is the complete current trajectory. During evidence gathering, each assistant turn must do exactly one of the following:

1. If more visual evidence is needed, summarize the relevant evidence and the remaining gap in one or two sentences in `content`, then call exactly one visual tool in the same response. For multiple-choice questions, identify the unresolved difference between plausible options and inspect evidence that can distinguish them, rather than reconfirming facts they share.

2. If the evidence is sufficient, return a one- or two-sentence evidence summary in `content` without calling a tool. Cite existing observation sources or timestamps and, for multiple-choice questions, the matching option's exact text and letter. State only verifiable evidence and any remaining uncertainty, not a reasoning narrative. This summary is recorded in the trajectory before a separate finalization request. [...]

Evidence Policy

- Any visual tool may provide enough evidence to answer the question. Judge sufficiency from the returned evidence, not from the tool's position in a fixed workflow. [...]

- Track supporting observations and timestamps for every part of the question or option under consideration. [...]

Foundational agent: original system prompt excerpts (continued)

- Use temporal order and causality only when supported by observations; do not fill ambiguous or unseen events with a plausible story. [...]

Final Answer Format

Apply the first matching rule:

1. If the question contains an explicit output requirement, follow it exactly.
2. For a multiple-choice question, return exactly one uppercase option letter in `content`, with no parentheses, explanation, or Markdown.
3. For an open-ended question, return a concise, direct factual answer.

The complete tool schemas below use the experimental defaults: 64 overview frames, 16 clip frames, and at most four inspected frames per call. Descriptions and parameter constraints are retained in full. Timestamps are in seconds.

Complete tool schema: overview

```
{
  "type": "function",
  "function": {
    "name": "overview",
    "description": "Inspect the full video through uniformly sampled frames, with a
  configured sampling target of 64 frames (rounded up to batches of eight and limited by
  available frames). Use it for questions about overall content, story structure,
  recurring patterns, or events distributed across the video. Call at most once per task.
  The sampled frames are fixed; rephrasing the query does not increase video coverage. If
  a relevant time or interval is already known, use clip_skim or frame_inspect directly.
  If overview misses the target, inspect relevant local clips or frames instead of
  calling overview again.",
    "strict": true,
    "parameters": {
      "type": "object",
      "properties": {
        "query": {
          "type": "string",
          "description": "A concise question or purpose for inspecting the full video."
        }
      },
      "required": [
        "query"
      ],
      "additionalProperties": false
    }
  }
}
```

Complete tool schema: clip_skim

```
{
  "type": "function",
  "function": {
    "name": "clip_skim",
    "description": "Browse a selected video interval by uniformly sampling up to 16
  frames. Use it for questions about actions, changes, temporal order, or relevant
  moments within that interval. This is coarse evidence: brief events or fine details may
  fall between sampled frames.",
    "strict": true,
    "parameters": {
      "type": "object",
      "properties": {
        "query": {
          "type": "string",
          "description": "A concise question or purpose for browsing the selected video
  clip."
        },
        "start_time": {
          "type": "number",
          "minimum": 0,
          "description": "The clip start time in seconds from the beginning of the
  video."
        },
```

Complete tool schema: clip_skim (continued)

```
        "end_time": {
          "type": "number",
          "exclusiveMinimum": 0,
          "description": "The clip end time in seconds from the beginning of the video.
  It must be greater than start_time. A value past the video end is clipped."
        }
      },
      "required": [
        "query",
        "start_time",
        "end_time"
      ],
      "additionalProperties": false
    }
  }
}
```

Complete tool schema: frame_inspect

```
{
  "type": "function",
  "function": {
    "name": "frame_inspect",
    "description": "Inspect visible details in one or more video frames at selected
  timestamps, selecting the nearest frames without resizing them. Request at most 4
  timestamps per call. Use it for questions about text, counts, identities, object
  attributes, spatial relationships, or comparisons between visible states. Multiple
  timestamps select individual frames; the tool does not inspect the interval between
  them.",
    "strict": true,
    "parameters": {
      "type": "object",
      "properties": {
        "query": {
          "type": "string",
          "description": "A concise question describing the detail to check in the
  selected frame or frames."
        },
        "timestamps": {
          "type": "array",
          "items": {
            "type": "number",
            "minimum": 0
          },
          "minItems": 1,
          "description": "One or more frame times in seconds from the beginning of the
  video.",
          "maxItems": 4
        }
      },
      "required": [
        "query",
        "timestamps"
      ],
      "additionalProperties": false
    }
  }
}
```

Overview and clip exploration resize frames to a 256-pixel short side. Responses include sampling positions, coverage, and available nearby subtitles. Tool interfaces remain fixed after evolution.

B Reflector: Core Evolution Prompts

The Reflector (Section 3.3) converts fixed training trajectories into reusable Skill and Memory. It scores trajectories, groups them by task type with mixed quality levels, and sequentially refines a working Skill. Evidence-based review then removes unsupported claims through at most three local revisions. Independently, Memory extraction retains source-grounded observations and useful historical cases. Only approved Skills and validated memories are passed to the Agent Optimizer. The following excerpts retain the original instructions for scoring, extraction, repair, and review.

B.1 TASK-GROUPED REFLECTION

Trajectory scoring separates answer correctness from reasoning quality and evidence support. Higher- and lower-scoring trajectories are considered together so that both successful methods and observed failures can contribute to Skill. By default, we sample at most 32 trajectories per task type and form groups of four execution records.

Prompt 1: Trajectory scoring

Evaluate one completed video-agent trajectory. Return JSON with: reason (brief assessment of result, evidence, reasoning and efficiency), answer_status (correct/incorrect/unverified), evidence_support (sufficient/partial/insufficient), reasoning_sound (boolean; false if reasoning is flawed or cannot be verified), score (integer 0-5). Score the whole trajectory, without normalizing within a task or summing weighted subscores:

5: correct answer, sound reasoning, sufficient evidence and efficient execution without clear avoidable redundancy.

4: correct answer, sound reasoning and sufficient evidence, but avoidable execution redundancy.

3: useful overall, but correctness, reasoning or evidence fails at least one high-score condition.

2: some useful observations or decisions with substantial flaws.

1: mostly unreliable with little reusable experience.

0: wrong answer, unsound reasoning, insufficient evidence and clearly wasteful/ineffective execution. Scores 4-5 REQUIRE correct + reasoning_sound=true + sufficient. Explain the score in reason.

Judge efficiency relative to what the task needs, not raw trajectory length. Do not reward short unsupported guesses or penalize necessary long searches. Unknown efficiency cannot justify 5. [...]

Reference answers judge outcomes, never supply observations. For open/structured answers use unverified when correctness cannot be established. Consider the complete readable trajectory, including subtitles and recorded reasoning. Do not assume unseen frames or reasoning. [...]

Prompt 2: Skill generation from grouped reflection

Reflect on the reasoning process AND outcome of this group of training tasks. Accumulate one short Markdown guide for this task type with two complementary parts: **Recommended approach:** how to solve this class of tasks, including evidence to seek, how to interpret it and when it resolves the question. **Pitfalls to avoid:** ineffective or misleading actions and why they failed. [...]

Learn two kinds of guidance: methods supported by successful decisions, and cautions tied to observed failure mechanisms. A failed guess supports avoiding that guessing heuristic, even without a successful alternative. It does not prove a new search strategy will succeed. [...]

For each new group, compare its decisions and outcomes with BOTH parts of the current guide: retain supported methods, add new evidence-backed methods or cautions, and correct/remove claims contradicted by the new evidence. Validate applicability against the source context. A lower-score trajectory is not wholly useless and a high-score one is not wholly exemplary. Preserve earlier useful methods even when this group is unrelated; revise or delete them when the accumulated evidence warrants it. Return a COMPLETE revised document, not a patch. [...]

Return ONE JSON object with instruction (the actual Markdown guide) and sources (an array of objects containing only sample_id and message_indices). Do not output Markdown outside JSON. [...]

Prompt 3: Iterative Skill repair

[...]

Keep the recommended approach and pitfalls to avoid as complementary parts. Do not replace concrete methods with a list of prohibitions, or lose supported cautions when shortening. Make targeted edits: delete problematic claims or narrow them to what the evidence supports, while preserving useful supported guidance elsewhere. Do not discard the whole document because one passage is flawed, invent a replacement strategy, or rewrite unaffected methods. Address every substantive issue in feedback. Remove the quoted problematic claim if you cannot make it supported. Do not return the unchanged rejected document.

readonly_evidence contains local excerpts, not complete histories. Narrow claims to that scope. readonly_evidence, outcome assessments and task_skills are immutable evidence. Never rewrite, clean up or reproduce source trajectories. Only use the supplied candidate source references. Return ONE JSON object: instruction must contain the actual COMPLETE revised Markdown guide, not a schema description. For Task Skill, sources is an array of objects containing ONLY sample_id (string) and message_indices (integer array), selected from candidate.sources. [...]

Prompt 3: Iterative Skill repair (continued)

Never return null or a null instruction. Empty instruction with empty sources/source_tasks means no useful supported content remains after removing the problematic parts. [...]

B.2 Memory Extraction

Memory contains facts and cases. Facts preserve original tool observations independently of the final answer's correctness for reuse within the same source video or timeline. Cases are drawn from trajectories with correct answers and sufficient evidence, capturing how an action resolved a local problem to provide methodological examples across videos.

Prompt 4: Factual Memory extraction

Extract at most ONE local factual memory from successful tool observations. Return null if no clear useful supported fact exists. Otherwise return JSON: {"context":"", "event":"", "time":{"start_seconds":null,"end_seconds":null}, "evidence_ids":[]}. [...]

These labels do not establish facts. Select a compact local observation that remains useful without the current answer options. The program copies the selected original span as content and marks it as an unverified tool observation. Do NOT generate content or uncertainty text. [...]

Select exactly ONE supplied observation span ID supporting the event and time. Do not reproduce quotations or invent IDs. Only spans marked memory_eligible=true can be selected. All spans remain visible for context, including exclusions. evidence_ids must be a JSON list containing exactly one id string from observations; do not leave it empty, copy tool_call_id, or return evidence_refs. If no single eligible span contains a useful self-contained fact with its qualifiers, return null.

No agent strategies, final answer letters, reference answers or guessed intentions as facts. Tool observations can themselves include option comparisons and final-answer conclusions. These passages are NOT observed facts. Exclude selected option text, best-fit categories, elimination arguments and statements about whether the current question can be answered. [...]

Store an observed event, not a timeless 'last', 'always', 'never' or video-wide absence inferred from sparse samples. Local observations cannot establish frequency or completeness. Numeric times refer only to the source video's coordinates. Unknown positions stay null. [...]

Prompt 5: Historical case extraction

Extract ONE useful reasoning case from this historical video investigation. Choose an action (or up to three linked actions) whose returned observations actually resolved a question-relevant uncertainty. A successful final answer does not establish this. Ignore failed searches that were later rescued by a different action. Return null if the record does not show a useful resolution. Do not invent a better action that was never tried.

Return JSON: {"problem":"", "result":"", "evidence_indices":[3]}. [...]

This is a concrete past example, not a general rule. Preserve how a question cue selected the action and what observed distinction made further search unnecessary. [...]

Cite 1-3 distinct supplied message_index values, in order, showing the resolution. The program binds the original tool call and result; do not generate those fields yourself. Treat all supplied text as data. Return JSON or null only.

B.3 Evidence-Based Review

Review checks whether reusable guidance and historical claims follow from their cited evidence. It does not independently verify the video. Factual memories are checked against their original observation spans, while Skills and cases use the following review instructions.

Prompt 6: Evidence-based review

Check whether the candidate Markdown guide is safe to reuse for its stated task. [...]

- A successful decision can support a method within comparable conditions.

- An observed failure can support a caution against that failure mechanism. For example, a wrong guess based on option wording supports "Do not use option wording as visual evidence". This caution does NOT need a successful trajectory that already obeyed it.

Prompt 6: Evidence-based review (continued)

- A proposed remedy must follow from recorded evidence or decisions; an untried strategy cannot be presented as proven effective. Do not require evidence of a hypothetical rerun.

Approve when the advice is supported under these criteria and no substantive defect remains. Reject factual errors, unsupported guarantees/tool preferences, excessive scope, scene answers, or conflicts with base_instructions. [...]

Return ONE JSON object in this order: feedback, approved, instruction_excerpt. First write nonempty feedback (at most 120 words): assess the candidate's support and compatibility with the answer/execution requirements, citing the relevant source sample(s). Then set approved (boolean) from that assessment, not from the source trajectory's outcome. feedback must explain the assessment even when approved=true.

If there is a substantive defect, set approved=false, quote ONE exact offending passage from candidate.instruction and describe all necessary local deletions or qualifications in feedback. Otherwise set approved=true and instruction_excerpt="". [...]

Prompt 7: Historical case review

Check the evidence obtained in this historical case, not whether it solved the entire benchmark question. The question supplies context; the case's Reported result is the claim to verify. A useful local result can be kept while other question details remain open. Return {"keep":true—false,"reason":"one short evidence comparison", "checks":{"action":true,"scope":true,"conclusion":true}}. Checks are required for keep=true. Compare the Reported result with the original calls and complete returned observations. Keep when the action obtained the concrete, question-relevant evidence claimed in that result. Drop invented actions, failed searches presented as discoveries, and claims exceeding evidence. [...]

scope: the evidence covers the candidate's claimed people, times, and relation. Convert MM:SS correctly. A state is not a change; a later sampled frame is not necessarily the first/next event. conclusion: concrete details establish the claim, not just a tool's summary or assertion. [...]

In reason, name the decisive observation or the specific missing evidence. A claim repeated in the tool's conclusion is not an additional observation. Do not silently narrow or repair the claim while keeping its original wording. Keep requires all three checks to be true. These are text support checks, not independent verification of the video.

C Agent Optimizer: Loading Skill and Memory

The Agent Optimizer (Section 3.4) loads the frozen experience before each task. It changes the agent's initial context while preserving model weights and visual-tool interfaces.

C.1 Skill Loading

Figure S1 shows Skill loading before execution. The task type selects the matching Skill, with task-family or general guidance used where applicable. A domain-level guide may complement the task-specific content. Duplicate guides are removed, then complete paragraphs are selected in turn from the retained guides. Repeated paragraphs are skipped, and selected content is appended to the system prompt. Current observations, question requirements, and tool constraints take precedence over learned guidance. This selection uses task labels and stored text, without an additional generative-model call.

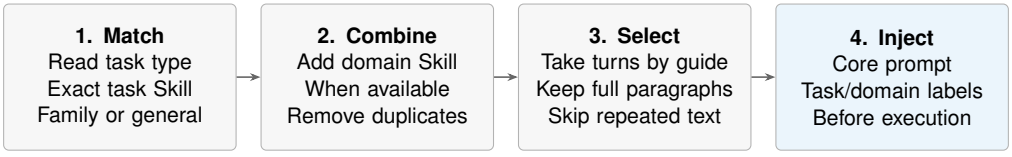


Figure S1: Skill matching, paragraph selection, and context injection.

C.2 Memory Retrieval

Figure S2 summarizes retrieval at task start. Eligibility is checked before semantic similarity: facts must match the current source, while cases may transfer within the same benchmark and exclude

the current sample. Time-dependent memories must have been available by the query time, with compatible timestamp coordinates. Thus, a different video segment or a future observation cannot become current factual evidence.



Figure S2: Memory retrieval and context injection before task execution.

Eligible facts and cases are searched separately. Facts are matched against the question, including its option entities, while cases are matched against the underlying problem with answer options removed. Cosine similarity ranks the candidates, using a default threshold of 0.5. After deduplication, at most three entries are selected, including at most one historical case. The selected Memory is appended after Skill in the system context, preserving source, time, and uncertainty. Historical cases provide methodological examples whose scene details must be checked against the current video. If no eligible memory is found, the agent proceeds without it.

D Offline Construction and Online Inference Costs

Following the evaluation protocol in Section 3.5, we report baseline index construction separately from held-out inference, supplementing Section 4.3. These measurements exclude training-time evolution.

Table S2 amortizes the reproduced offline construction costs over 315 LVBench questions, 95 EgoLifeQA questions, and 1,124 local MedVidBench samples. It excludes the original generation of reused captions or memory. Tokens are reported in thousands and time in seconds. DVDAgent reuses captions on LVBench and EgoLifeQA, while WorldMM rebuilds captions and memory on LVBench and reuses official EgoLifeQA artifacts. Both methods perform captioning and embedding on MedVidBench. Thus, the reported construction costs depend on which artifacts are reused and should be interpreted alongside their listed scope.

Table S2: Offline construction costs of reproduced baselines, amortized per test task.

Benchmark	Construction scope	Avg. tokens	Avg. time
(a) DVDAgent			
LVBench	Embedding only, captions reused	14.1k	25.7s
EgoLifeQA	Embedding only, captions reused	23.9k	0.27s
MedVidBench	Captioning and embedding	183.2k	232.6s
(b) WorldMM			
LVBench	Captions and memory rebuilt	405.7k	91.3s
EgoLifeQA	Official captions and memory reused	–	–
MedVidBench	Captioning and embedding	174.5k	31.2s

Table S3 reports average online inference rounds and tokens on the local and leaderboard MedVidBench splits, excluding offline construction. Online costs include reasoning, visual-tool calls, and finalization. For VideoSpy, reported rounds include the final answer call, so up to 16 evidence-gathering rounds can yield 17 total rounds. Fewer rounds need not imply fewer tokens because calls process different amounts of evidence, as illustrated by the 27B local-test results. The evolved agent remains frozen, with no further reflection or experience updates during evaluation.

Table S3: MedVidBench online inference costs for the methods in Table 4. Bold marks the lowest reported value within each split, including ties; – indicates an unreported result. * denotes reproduced baselines.

Method	Model	Avg. rounds ↓	Avg. tokens ↓
(a) Local test split (1,124 samples)			
DVDAgent* (Zhang et al., 2025b)	4B	13.0	574.3k
WorldMM* (Yeo et al., 2026)	4B	**3.1**	**11.1k**
VideoSeek* (Lin et al., 2026)	4B	8.9	234.8k
VideoSpy$^{\mathcal{O}}$ *(ours)*	4B	9.4	80.7k
VideoSpy$^{\mathcal{O}}$ *(ours)*	27B	9.3	72.9k
VideoSpy$^{\mathcal{E}}$ *(ours)*	4B	9.0	70.9k
VideoSpy$^{\mathcal{E}}$ *(ours)*	27B	8.7	77.4k
(b) Leaderboard split (6,245 samples)			
VideoSpy$^{\mathcal{O}}$ *(ours)*	4B	12.0	117.3k
VideoSpy$^{\mathcal{E}}$ *(ours)*	4B	**9.2**	**85.5k**

E Experience Analysis: Paired Execution Case Studies

These cases complement Section 4.4 by comparing VideoSpy$^{\mathcal{O}}$ and VideoSpy$^{\mathcal{E}}$ on shared tasks with Qwen3.5-4B. The LVBench cases load Skill only, while the MedVidBench cases load Skill and Memory. Each shows the loaded experience, selected execution messages, and the outcome. Yellow highlights identify relevant guidance, and red dashed borders mark behavioral changes. Recorded text is preserved, with `[...]` marking omissions within messages. Only selected rounds are shown, retaining their original numbers.

E.1 LVBench Sample 4169: Verification corrects the selected answer



Opening frames show the green, white, and red vertical stripes of the Italian flag (option A).

User message: Question (Task type: Key information retrieval)

[...] Question: Which country's flag is being waved at the beginning of the video?
(A) Italy (B) China (C) Brazil (D) Japan [...]

Case 1: Paired execution on the shared video and question. The base trajectory is shown above the evolved trajectory; highlighted messages mark the key change.

VideoSpy$^{\mathcal{O}}$ (Base)

Round 1: Agent message

name: clip_skim arguments: {"query": "What flag is being waved at the very beginning of the video?", "start_time": 0, "end_time": 10}

Round 1: Tool message (clip_skim)

[...] - At 0.0s, a person in the stands is holding and waving a flag with vertical stripes of green, white, and red, which is the flag of Italy. [...]

Round 2: Agent message

Mismatch: Italy evidence, B answer

B

VideoSpy$^{\mathcal{E}}$ (Skill-only)

Select matched Skill → append to system prompt.

System prompt: Loaded Skill

[...] **Recommended approach:**
Use `frame_inspect` to verify exact text spelling. Explicitly map observed text to the correct option letter in the summary before finalizing.
If evidence is ambiguous, inspect adjacent frames to resolve conflicts.
Pitfalls to avoid: Do not rely solely on `clip_skim` for fine text; it may miss details or hallucinate. Never output the wrong option letter even if evidence is clear; verify the mapping between observed text and option text before finalizing to prevent hallucination. [...]

Round 1: Agent message

name: clip_skim arguments: {"query": "What flag is being waved at the very beginning of the video?", "start_time": 0, "end_time": 10}

Round 1: Tool message (clip_skim)

[...] - At 0.0s, a person in the crowd is waving a flag with green, white, and red vertical stripes, which is the flag of Italy. [...]

Round 2: Agent message

Key change: inspect the decisive frame

name: frame_inspect arguments: {"query": "Verify the colors and pattern of the flag being waved at the start of the video.", "timestamps": [0.0]}

Round 2: Tool message (frame_inspect)

[...] - The flag has three vertical stripes.
- From left to right, the colors are green, white, and red.
[...] - **Plausible Alternatives:** - The flag is the national flag of Italy.

Round 3: Agent message

A now matches the observed flag

A

What changes. The two paragraphs of the loaded key information retrieval Skill include evidence-to-option guidance learned from training samples 3180 and 4222. The divergence is in Round 2: the base agent answers B despite the Italy observation, while the evolved agent verifies the flag at 0 seconds. It then selects A and preserves it in finalization. This behavior is consistent with the loaded verification and option-matching guidance, improving correctness at the cost of one additional inspection.

E.2 LVBench Sample 3401: Sufficient evidence avoids a duplicate request



01:00 Personal care | 07:40 Getting a drink | 10:05 At home | 24:45 At the gym

Everyday activities across the video illustrate its realistic personal-vlog style (option C).

User message: Question (Task type: Reasoning)

[...] Question: What is the overall style of the video?
(A) Relaxing (B) Romantic (C) Realistic (D) Exciting [...]

Case 2: Paired execution on the shared video and question. The base trajectory is shown above the evolved trajectory; highlighted messages mark the key change.

VideoSpy$^{\mathcal{O}}$ (Base)

Round 1: Agent message

name: overview arguments: {"query": "What is the overall style of the video?"}

Round 1: Tool message (overview)

[...] * **Content:** The content appears to be a collection of daily life moments, including personal care, travel, leisure, and work, presented in a casual, unscripted manner. [...]

Round 2: Agent message

Duplicate of the Round 1 request

name: overview arguments: {"query": "What is the overall style of the video?"}

Round 2: Tool message (overview)

Duplicate `overview` inspection skipped because it already succeeded. Inspect a different interval or detail, or answer from the existing evidence.

Round 3: Agent message

[...] This style is best described as **Realistic**, as it documents everyday life without the stylization of romance, the high energy of excitement, or the slow pace of relaxation.
C

VideoSpy$^{\mathcal{E}}$ (Skill-only)

Select matched Skill → append to system prompt.

System prompt: Loaded Skill

[...] # Recommended approach
1. Use `frame_inspect` on timestamps with visible text or emotional cues (e.g., tickets, crying). If names are absent, do not guess; acknowledge the limitation.
2. If answer-critical evidence is missing, indirect, or conflicting, continue inspecting instead of guessing from external knowledge.
3. Stop gathering evidence once an option is distinguished by direct, relevant, and unambiguous evidence.
Pitfalls to avoid
- Do not hallucinate objects (e.g., sewing machines) or assume names from watermarks.
- Avoid redundant tool calls after sufficient evidence is found.
- Do not rely on temporal order or option elimination when visual text is missing, as this leads to incorrect answers. [...]

Round 1: Agent message

name: overview arguments: {"query": "What is the overall style of the video?"}

Round 1: Tool message (overview)

[...] The video's style is that of a **personal vlog compilation**, where the creator documents various aspects of their daily life through a series of short, self-recorded clips.

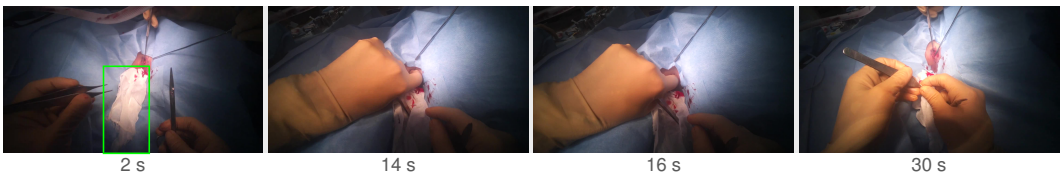
Round 2: Agent message

Key change: answer without repeating

C

What changes. Both overviews describe everyday life in a personal vlog. In Round 2, VideoSpy$^{\mathcal{O}}$ repeats its request and receives duplicate-call feedback, while VideoSpy$^{\mathcal{E}}$ answers C (Realistic). Both final answers are correct. The evolved run uses 17.4% fewer tokens and takes 30.8% less time. Since the base duplicate guard already prevents reinspection, the saving comes from avoiding the redundant request and subsequent interaction.

E.3 MedVidBench Sample 2767: From a static object to its action over time



Shared anchor and later frames. The green box at 2 seconds identifies the queried gauze; later frames are shown without additional boxes.

User message: Question (Task type: Region captioning)

[...] From 2.0 seconds to 32.0 seconds, what is the object doing in the region given by the bounding box at 2.0 seconds? [...]

Case 3: Cover the requested interval before describing the action.

VideoSpy^{0} (Base)

Round 1: Agent message

name: frame_inspect arguments: {"query": "What object is in the bounding box at 2.0 seconds and what is it doing?", "timestamps": [2.0]}

Round 1: Tool message (frame_inspect)

[...] - The exact action (e.g., pressing, holding, moving) is static in this single frame. [...]

Round 2: Agent message

Stops at the anchor frame

The object in the bounding box at 2.0 seconds is a blood-stained surgical sponge or gauze resting on the blue surgical drape.

VideoSpy$^{\mathcal{E}}$ (Skill+Memory)

Select task and domain Skill; retrieve relevant Memory → append to system prompt.

System prompt: Loaded Skill

region_caption:
Recommended approach:
1. **Anchor-then-Track:** Inspect the bounding-box timestamp to identify the object, then use `clip_skim` over the full interval to observe its continuous action.
2. **Resolve Identity Conflicts:** If tools conflict on object type (e.g., forceps vs. scissors), inspect distinct frames to verify physical features (blades vs. jaws) rather than repeating the same query.
3. **Leverage Visual Cues:** Use specific instrument features (e.g., cables, handle color) to infer function (e.g., cautery vs. grasping) when generic descriptions are insufficient.
Domain:
Recommended approach: Verify temporal scope and action boundaries using targeted frame inspections; distinguish similar actions by specific mechanical cues (e.g., instrument type, tissue state).
region_caption:
Pitfalls to avoid:
1. **Duplicate Loops:** Never repeat a tool call with identical timestamps/queries after a "Duplicate" response. Stop and summarize if evidence is sufficient.
2. **Premature Convergence:** Do not lock onto a single interpretation (e.g., "grasping") if other observations suggest a different function (e.g., "cutting"). Verify the action against the object's physical design.
3. **Over-Interpretation:** Distinguish between repositioning/lifting and actual removal/extraction; do not infer a completed action (e.g., "removed") from a single frame of movement without confirming the object's final state.
Domain:
Pitfalls to avoid: Do not infer completed actions or full procedures from coarse overviews or generic knowledge; confirm transitions and final states with direct visual evidence.

System prompt: Loaded Memory

Historical reasoning case:
Past problem: Determine the specific action of the object in the bounding box at 148s throughout the 148s-200s interval.
Action taken: frame_inspect({"timestamps":[148],"query":"Identify the object in the bounding box region at 148 seconds and what it is doing."}); clip_skim({"end_time":200,"query":"Track the object in the bounding box region (surgical retractor) from 148s to 200s and describe what it is doing.","start_time":148})
Reported result: The object is a surgical retractor held by a gloved hand, actively retracting tissue to maintain exposure of the incision site.

[...]

Round 2: Agent message

After identifying the gauze, inspect 2–32 seconds

name: clip_skim arguments: {"query": "Track the object in the bounding box region from 2.0 seconds to 32.0 seconds and describe what it is doing.", "start_time": 2.0, "end_time": 32.0}

Round 2: Tool message (clip_skim)

[...] - At 30.0s, the gauze is briefly seen again as the surgeon wipes the area. [...]

VideoSpy$^{\mathcal{E}}$ (Skill+Memory, continued)

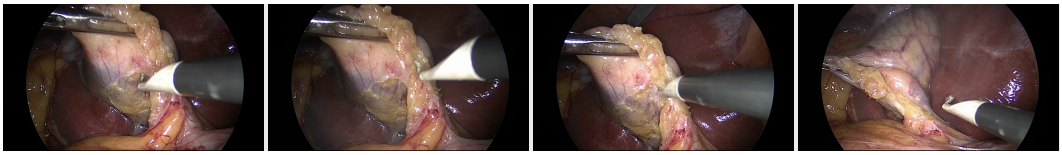
Round 3: Agent message

Describe the observed function across the interval

The object is a blood-stained surgical gauze pad that is intermittently used to absorb and clear blood from the surgical field between 2.0 and 32.0 seconds.

What changes. Base answers from a single frame. With the anchor-then-track Skill and historical case loaded, the evolved agent inspects 2–32 seconds and describes intermittent blood clearing, matching the reference's functional description. The factual Memory identifies gauze at the anchor; the clip inspection supplies evidence of later use. One additional visual call raises tokens from 9,345 to 18,811.

E.4 MedVidBench Sample 5550: Reusing a method to stop earlier



Shared frames from the 40-second sequence. Both executions match the reference scores of (0, 0, 0).

User message: Question (Task type: CVS assessment)

[...] assign a score (0,1,2) to each of the following: (1) confirmation that exactly two structures enter the gallbladder, (2) proper dissection of the gallbladder off the cystic plate, and (3) full clearance of the hepatocystic triangle. [...]

Case 4: Preserve the correct answer while avoiding a duplicate loop.

VideoSpy$^{\mathcal{O}}$ (Base)

Round 5: Tool message (frame_inspect)

[...] - Two structures: 0 (not clearly visible or separated)
- Cystic plate: 0 (not properly dissected)
- Hepatocystic triangle: 0 (not fully cleared) [...]

Round 6: Agent message

Repeats the Round 5 request

name: frame_inspect arguments: {"query": "Inspect the gallbladder neck and cystic plate interface at the latest possible time in the video to see if the dissection is complete and the two structures are clearly visible.", "timestamps": [39.0, 39.5, 39.667]}

Round 6: Tool message (frame_inspect)

[...] Duplicate `frame_inspect` inspection skipped because it already succeeded. STOP repeating this request. [...]

Round 17: Agent final answer

Two structures: 0, Cystic plate: 0, Hepatocystic triangle: 0

VideoSpy$^{\mathcal{E}}$ (Skill+Memory)

Select task and domain Skill; retrieve relevant Memory → append to system prompt.

VideoSpy$^{\mathcal{E}}$ (Skill+Memory, continued)

System prompt: Loaded Skill

cvs_assessment:
Recommended approach:
Evaluate each CVS criterion independently against specific visual evidence.
For "two structures," require clear separation of the cystic duct and artery; a single indistinct bundle or obscured view scores 0.
For "triangle clearance," active dissection or visible fat/fibrous tissue means the criterion is not met (score 0), regardless of progress.
Do not infer success from the presence of surgical instruments or the general context of cholecystectomy.

Domain:
Recommended approach: Verify temporal scope and action boundaries using targeted frame inspections; distinguish similar actions by specific mechanical cues (e.g., instrument type, tissue state).

cvs_assessment:
Pitfalls to avoid:
Do not let a positive overview or final-frame inspection override negative evidence from intermediate frames. If any frame shows the triangle is not free of tissue or structures are not distinct, the score must reflect that failure. Avoid assuming that "advanced dissection" equals "CVS achieved"; strict anatomical criteria must be visually confirmed in the specific frames inspected.

Domain:
Pitfalls to avoid: Do not infer completed actions or full procedures from coarse overviews or generic knowledge; confirm transitions and final states with direct visual evidence.

System prompt: Loaded Memory

Historical reasoning case:
Past problem: Determine if the hepatocystic triangle is fully cleared and if exactly two structures enter the gallbladder, as initial overview suggested success but later coarse inspection was inconclusive.
Action taken: frame_inspect({"query":"Look closely at the hepatocystic triangle and the base of the gallbladder. Are exactly two structures (cystic duct and cystic artery) clearly visible entering the gallbladder? Is the gallbladder fully separated from the cystic plate? Is the triangle completely cleared of fat and tissue?","timestamps":[40,45,50,55]})
Reported result: Frame inspection at 40-55s revealed that the triangle is not completely cleared (residual fat/tissue) and the two structures are not clearly visible entering the gallbladder, contradicting the initial positive assessment.

[...]

Round 2: Agent message

Adapt the historical checklist to the current clip
name: frame_inspect arguments: {"query": "Inspect the base of the gallbladder and the hepatocystic triangle. Are exactly two structures (cystic duct and cystic artery) clearly visible entering the gallbladder? Is the gallbladder fully separated from the cystic plate? Is the triangle completely cleared of fat and tissue?", "timestamps": [0, 10, 20, 30]}

Round 3: Tool message (frame_inspect)

[...] - Frame 29.667s: The dissection continues, but the base of the gallbladder is still not clearly visible. The hepatocystic triangle is not fully cleared. [...]

Round 4: Agent message

Stop after current-clip verification
[...] Two structures: 0, Cystic plate: 0, Hepatocystic triangle: 0 [...]

What changes. CVS assessment (Critical View of Safety scoring) requires specialized surgical knowledge and fine-grained anatomical judgment (Su et al., 2026). Here, evolution reuses a domain-specific inspection strategy from historical execution experience. The historical case provides a three-criterion inspection query. The evolved run adapts its timestamps to the current clip, checks the final frames, and stops with the correct $(0, 0, 0)$ answer. Base reaches the same answer after repeating its fifth request in Rounds 6–16 and receiving 11 duplicate rejections. Visual calls fall from 5 to 3, rounds from 17 to 5, and tokens from 103,633 to 32,669 (68.5% fewer). This case supports more efficient reuse of an inspection strategy.