VIDHARNESS: EVOLVING AGENT HARNESSES FOR COST-EFFICIENT LONG VIDEO UNDERSTANDING

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-language models (VLMs) can answer questions about hour-long videos, but processing every frame is prohibitively expensive, even though the evidence for a question usually spans only a few seconds. Video agents, i.e., harness programs wrapped around a frozen VLM, address this by observing the video selectively, yet existing harnesses are hand-crafted by experts through slow build-and-test cycles. We propose VidHarness, a framework that automates harness design for cost-efficient long video understanding, in which a harness proposer iteratively evolves harnesses based on execution feedback from an evolution environment. To escape the local optima of greedy refinement, we organize the evolution as Monte Carlo tree search (MCTS), and to reduce the evaluation cost, we integrate uncertainty-aware multi-fidelity validation, which screens new harnesses on a few questions and promotes only the promising ones. Since the best harness varies with the frame budget, we further introduce a mixture-of-harness that routes each question to a harness specialized for its budget. VidHarness sets new state-of-the-art results on LongVideoBench, Video-MME, and Video-Holmes, outperforms the strongest hand-crafted video agent by up to 11.2 points, and generalizes to the knowledge-intensive benchmarks Video-MMMU and MMVU with fewer than half of the frames of uniform sampling.

1 INTRODUCTION

Machine perception is shifting from clips to hours. Film and sports analysis must follow narratives across full-length footage (Wu et al., 2024; Wang et al., 2025a; Fu et al., 2025; Cheng et al., 2025), personal assistants must recall events from day-long egocentric recordings (Tian et al., 2026), and surveillance systems accumulate continuous streams that no human can watch in full (Yuan et al., 2024). In these applications, the answer to a question typically occupies seconds of content buried in hours of redundancy, so treating every frame equally wastes most computation on irrelevant content.

Vision-language models (VLMs) have responded by extending context windows and compressing visual tokens to fit more frames into a single forward pass (Team et al., 2024; Zhang et al., 2024; Chen et al., 2025b; Shu et al., 2025; Bai et al., 2025). This strategy scales poorly with duration: a one-hour video at one frame per second yields about 1.84 million visual tokens, exceeding most models' context windows, and the cost grows linearly with video length while the answer-relevant evidence does not. Cost-efficient long video question answering (QA) is therefore in high demand.

A promising response is the agentic *harness*: a program wrapped around a frozen VLM that decides what the model observes, in what order, and when it answers (Anthropic, 2025b; Guo et al., 2026). Under this view, existing video agents (Wang et al., 2024; Fan et al., 2024; Wang et al., 2025b; Ma et al., 2025; Yang et al., 2025; Pang & Wang, 2025; Zhang et al., 2026b; Li et al., 2026; Lin et al., 2026) are specific harnesses, and recent ones such as VideoSeek (Lin et al., 2026) and LensWalk (Li et al., 2026) achieve strong accuracy with few frames. Behind these results, however, lies extensive manual engineering: experts propose, implement, test, diagnose, and revise mechanisms over many trial-and-error cycles, which demands joint expertise in video understanding and agent design and must be repeated for every new application. We ask **whether this design loop can be automated**, and answer affirmatively with VidHarness, a framework that automatically evolves agent harnesses to boost a frozen VLM on cost-efficient long video QA.

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.

In VidHarness, a harness proposer operates inside an evolution environment: it writes candidate harnesses, executes them under a strictly enforced frame budget, analyzes the execution feedback, and revises them over iterations. Effective harness evolution and deployment, however, face three challenges. **(1) Local optima.** Vanilla harness evolution (Lee et al., 2026) revises harnesses along a single lineage and easily converges to a local optimum. We therefore organize the evolution as Monte Carlo tree search (MCTS) over a harness tree, which balances introducing new mechanisms (exploration) and refining promising harnesses (exploitation). **(2) Evaluation cost.** Fully validating every candidate on long videos takes hours and incurs high API cost, although most candidates are not competitive. We therefore propose uncertainty-aware multi-fidelity validation, which first validates new harnesses on a few questions, promotes promising ones to more questions, and penalizes accuracies measured on few questions during node selection. **(3) Diverse budgets.** Applications request different frame budgets, and the best harness under a small budget (e.g., 16 frames) often differs from the best one under a large budget (e.g., 128 frames). We therefore introduce a mixture-of-harness that routes each question to a harness specialized for its budget tier.

To evolve harnesses, we curate a validation set of $1{,}000$ QA pairs from existing video datasets (Chen et al., 2026b; Fu et al., 2025; Wu et al., 2024; Cheng et al., 2025; Chen et al., 2026b; Zhu et al., 2026), which we will release. VidHarness sets new state-of-the-art results on LongVideoBench (72.0%), Video-MME (73.1% without and 83.2% with subtitles), and Video-Holmes (60.8%), and outperforms VideoSeek on the same backbone by up to 11.2 points. On the out-of-domain benchmarks Video-MMMU (Hu et al., 2026) and MMVU (Zhao et al., 2025), even its lowest budget tier surpasses uniform frame sampling while using fewer than half of the frames.

In summary, our contributions are:

- We propose VidHarness, a framework that automates harness design for cost-efficient long video understanding through iterative harness evolution.
- We design an MCTS-based evolution algorithm with uncertainty-aware multi-fidelity validation, which escapes local optima and substantially reduces the evaluation cost.
- We introduce a mixture-of-harness that routes each question by its frame budget.
- We demonstrate state-of-the-art results on LongVideoBench, Video-MME, and Video-Holmes, surpassing the hand-crafted VideoSeek by up to 11.2 points.

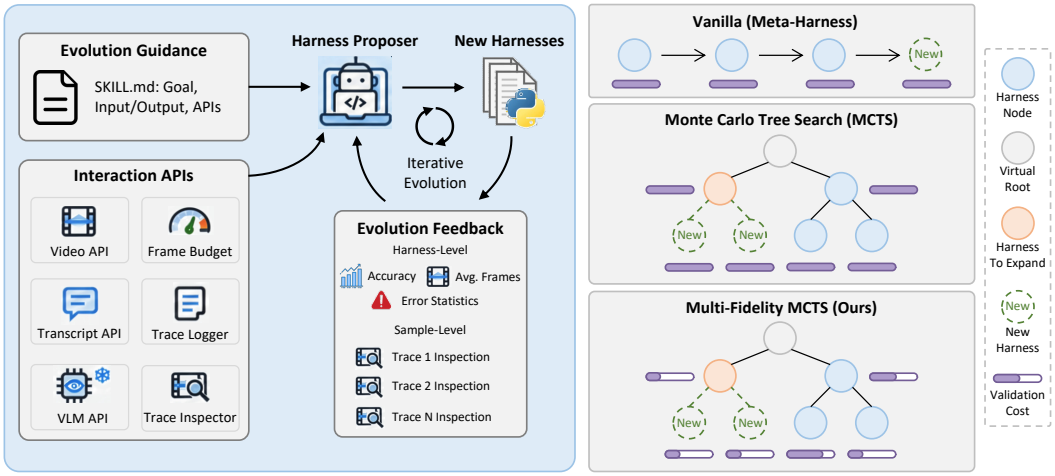
2 Related Work

Our work is closely related to long video understanding, agents and harnesses, and evolution algorithms. We summarize each line of work below and provide an extended discussion in Appendix A.

Long Video Understanding. Existing methods scale VLMs to ingest more frames (Team et al., 2024; Zhang et al., 2024; Shu et al., 2025), select query-relevant keyframes (Tang et al., 2025; Yu et al., 2025; Ye et al., 2025), or wrap the VLM in hand-crafted agentic harnesses that gather evidence (Wang et al., 2024; 2025b; Zhang et al., 2026b; Li et al., 2026; Lin et al., 2026), whereas VidHarness evolves the harness automatically.

Agents and Harnesses. Recent work recognizes that agent capability depends heavily on the surrounding harness and optimizes harnesses for text, code, game, and, concurrently, video tasks (Guo et al., 2026; Lee et al., 2026; Lou et al., 2026; Xu & Chen, 2026; Cui et al., 2026), but these methods largely follow a single improvement trajectory and deploy one agent, whereas VidHarness explores a harness tree with multi-fidelity MCTS and assembles a mixture-of-harness.

Evolution Algorithms. LLM-driven evolution and tree search have been used to discover programs, prompts, and agentic systems (Romera-Paredes et al., 2024; Novikov et al., 2025; Hu et al., 2025; Zhang et al., 2025b; Jiang et al., 2025), but most of them evaluate every candidate in full, whereas VidHarness integrates multi-fidelity evaluation (Li et al., 2018) into MCTS to reduce the cost of validating harnesses on long videos.



(a) Evolution Environment. (b) Evolution Algorithms.

Figure 1: Overview of harness evolution in VidHarness. Left: in the evolution environment, the harness proposer follows the evolution guidance and uses the interaction APIs to design new harnesses, whose evolution feedback is returned to the proposer for the next iteration. Right: comparison of evolution algorithms, where each circle is a harness and its bar shows the fraction of validation questions evaluated. Vanilla iterative refinement grows a single lineage and MCTS grows a tree, but both validate every harness in full; our multi-fidelity MCTS validates new harnesses on a small subset and re-validates only the promising ones.

3 VIDHARNESS

We propose VidHarness, a framework that automatically evolves agent harnesses to boost a frozen VLM for cost-efficient long video QA. We first formulate long video QA under a frame budget and the harness evolution problem in Section 3.1. Section 3.2 presents the evolution environment, in which candidate harnesses are written, executed, and diagnosed. Section 3.3 describes the validation set on which the candidate harnesses are evaluated. Section 3.4 then details the multi-fidelity MCTS that navigates the harness space at a low evaluation cost. Finally, Section 3.5 introduces the mixture-of-harness, which routes each question to an evolved harness specialized for its frame budget.

3.1 PROBLEM FORMULATION

Long Video QA under a Frame Budget. Given a video v, its audio transcription a, a question q, and a frame budget b, the task is to design an approach that uses a frozen VLM f to produce a prediction y matching the ground truth y^*, while decoding no more than b frames from v. As discussed in Section 1, many video agents (Wang et al., 2024; Fan et al., 2024; Wang et al., 2025b; Ma et al., 2025; Yang et al., 2025; Pang & Wang, 2025; Zhang et al., 2026b; Li et al., 2026; Lin et al., 2026) build agentic frameworks around frozen VLMs that perform question-driven frame selection, which enables cost-efficient long video QA under a limited frame budget. We refer to such an agentic workflow as a *harness* h: an executable program that orchestrates the multi-turn interaction between the video, the transcription, and the model. Running the harness yields

$$y = h(v, a, q;\, f, b). \tag{1}$$

Harness Evolution. Let $\mathcal{H}$ denote the space of executable harnesses. Given a validation set $\mathcal{D}_{\text{val}} = \{(v_i, a_i, q_i, y_i^*)\}_{i=1}^{|\mathcal{D}_{\text{val}}|}$, harness evolution seeks the harness that maximizes accuracy under the frame budget b:

$$h^* = \arg\max_{h \in \mathcal{H}} \frac{1}{|\mathcal{D}_{\text{val}}|} \sum_{i=1}^{|\mathcal{D}_{\text{val}}|} \mathbb{I}\big[h(v_i, a_i, q_i;\, f, b) = y_i^*\big], \tag{2}$$

where $\mathbb{I}[\cdot]$ is the indicator function. To solve Equation 2, we design an iterative evolution algorithm that identifies the top-performing harness h^* in the harness space $\mathcal{H}$.

3.2 Evolution Environment

The evolution environment (Figure 1, left) supports the evolution algorithm with three services, namely evolution guidance, interaction APIs, and evolution feedback, through which the harness proposer designs, executes, and diagnoses candidate harnesses.

Evolution Guidance. We provide the guidance to the harness proposer as a skill file (`SKILL.md`) (Anthropic, 2025c), which specifies the task goal, i.e., maximizing the accuracy of long video QA within the frame budget b, the input and output protocol of a harness, and the available APIs with their constraints. In particular, a harness is a self-contained program that realizes Equation 1 and returns its prediction together with a standardized execution trace.

Interaction APIs. The interaction APIs serve both the harnesses and the proposer. A harness uses the video API to sample frames within any time window at an adjustable resolution, the transcript API to retrieve the transcript by time or keyword at no frame cost, and the VLM API to converse with the frozen backbone f using interleaved video frames and text. Every decoded frame is deducted from the budget b, and requests beyond the remaining budget are rejected. For the proposer, a trace logger records every VLM call, frame sample, and transcript read, and a trace inspector re-renders the frames a harness observed for failure analysis.

Evolution Feedback. After validation, the environment returns harness-level feedback, including the accuracy, the average frame usage, and error statistics, together with the sample-level trace of every question. To understand failures, the proposer dispatches trace inspectors, vision-capable subagents that examine the observed frames, diagnose causes such as missed coverage, insufficient resolution, or flawed reasoning, and return concise text reports.

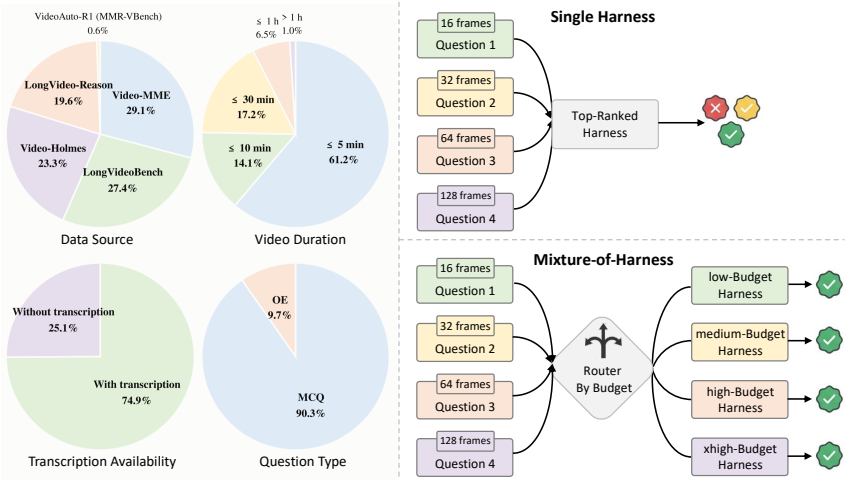
Harness Proposer. The harness proposer $\mathcal{M}$ is instantiated as a coding agent. At each iteration, it receives the guidance, the interaction APIs, and the feedback of the harnesses evaluated so far, diagnoses their failures, and designs new harnesses that aim to improve accuracy under the frame budget. The new harnesses are then validated, and their feedback is returned to the proposer for the next iteration. Section 3.4 describes how the evolution algorithm decides which harness to revise and how thoroughly to validate each new harness.

3.3 Validation Set Construction

To facilitate harness evolution, we curate a validation set $\mathcal{D}_{\text{val}}$ of $1{,}000$ QA pairs from multiple data sources (Fu et al., 2025; Wu et al., 2024; Cheng et al., 2025; Chen et al., 2026b; Zhu et al., 2026). Figure 2 (left) summarizes its statistics. The QA pairs mainly come from Video-MME (29.1%), LongVideoBench (27.4%), Video-Holmes (23.3%), and LongVideo-Reason (19.6%), with a small portion from VideoAuto-R1 (MMR-VBench) (0.6%). We ensure that the validation set does not overlap with any test benchmark in Section 4. The video durations range from under 5 minutes to over 1 hour, which exposes the evolution to inputs of diverse lengths and encourages harnesses that handle both short and long videos. In addition, 74.9% of the samples include an ASR transcription (Radford et al., 2023) and the remaining 25.1% do not, which guides the evolution to handle both cases. The questions cover both multiple-choice (90.3%) and open-ended (9.7%) formats. We will release the validation set to the community to support future research.

3.4 Multi-Fidelity MCTS

Vanilla iterative refinement (Meta-Harness; Lee et al., 2026) repeatedly revises an existing harness based on its evolution history and validates every candidate on the full validation set. This strategy has two limitations. First, it is greedy: since all proposals descend from the first design that works, the evolution is easily trapped in **local optima**. Second, it is slow: most of the evaluation budget is spent on candidates that turn out to be weak, resulting in a high **evaluation cost**. VidHarness instead organizes harnesses into a tree rather than a single lineage, and combines Monte Carlo tree search (MCTS), which balances exploration and exploitation, with uncertainty-aware multi-fidelity



(a) Validation Set Statistics. (b) Harness Deployment Comparison.

Figure 2: Validation set statistics and mixture-of-harness. Left: distributions of data sources, video durations, transcription availability, and question types (multiple-choice, MCQ, and open-ended, OE) in the validation set. Right: a single top-ranked harness serves questions of all frame budgets but fits only some of them well (top), whereas the mixture-of-harness routes each question to the harness specialized for its budget tier (bottom).

validation, which spends evaluation effort in proportion to how promising a harness is, as illustrated in Figure 1 (right) and summarized in Algorithm 1 (Appendix B).

Monte Carlo Tree Search. We organize the harnesses designed during evolution into a harness tree $\mathcal{T}$, in which each child revises its parent. The root is a virtual node that corresponds to no harness, and every other node is a harness h associated with its evolution feedback. Each iteration of MCTS consists of four stages: node selection, node expansion, validation, and backpropagation.

In node selection, the evolution algorithm descends from the root until it reaches a node that can be further expanded. At each node h, it moves to the child h' with the highest upper confidence bound for trees (UCT) score (Kocsis & Szepesvári, 2006):

$$\mathrm{UCT}(h') = \underbrace{Q(h')}_{\text{accuracy (exploitation)}} + \underbrace{\lambda_{\mathrm{x}} \sqrt{\frac{\ln\big(N(h)+1\big)}{N(h')+1}}}_{\text{visit count (exploration)}}, \tag{3}$$

where $Q(h')$ is the highest accuracy among the harnesses in the subtree rooted at h', $N(\cdot)$ is the visit count of a node, i.e., the number of harnesses evaluated in its subtree, and λ_{x} is the exploration weight. The first term favors branches that contain accurate harnesses, and the second term favors branches that have rarely been visited.

In node expansion, the harness proposer designs new harnesses as children of the selected node h, conditioned on the current harness tree $\mathcal{T}$, which contains all evaluated harnesses and their feedback. We denote this operation by $\mathcal{M}(\mathcal{T})$. If the selected node is the virtual root, the proposer designs harnesses whose core mechanisms differ from all harnesses in the tree, which encourages exploration. Otherwise, it revises the selected harness, which encourages exploitation.

In validation, each new harness is executed on the validation set $\mathcal{D}_{\mathrm{val}}$ under the frame budget b, which yields its accuracy $\mathrm{acc}(h)$ and its feedback.

In backpropagation, every ancestor of a new node increments its visit count $N(\cdot)$ and updates $Q(\cdot)$ if the new harness is more accurate than all harnesses in its subtree. We adopt this max-style backup rather than the usual average because evolution seeks the single best harness.

Uncertainty-Aware Multi-Fidelity Validation. To reduce the evaluation cost, we propose uncertainty-aware multi-fidelity validation, inspired by multi-fidelity hyperparameter optimiza-

tion (Jamieson & Talwalkar, 2016; Li et al., 2018; Falkner et al., 2018). We define L rungs $r_1 < \cdots < r_L = |\mathcal{D}_{\text{val}}|$, where rung r_ℓ covers the first r_ℓ questions of $\mathcal{D}_{\text{val}}$. Let $\text{acc}_r(h)$ denote the accuracy of h on the first r questions, and let $r(h)$ denote the number of questions on which harness h has been validated. Each new harness is first validated only at rung r_1, which substantially accelerates evolution and reduces its cost. However, an accuracy measured on a few questions is noisy: its standard error can reach $1/(2\sqrt{r})$ (Appendix C), so it is an unreliable estimate of the true accuracy of h. We address this uncertainty with two designs.

First, we add a promotion stage at the end of each iteration. Among the harnesses that have not yet been validated on the full validation set, we select the one with the highest accuracy,

$$h^\dagger = \underset{h:\, r(h)<|\mathcal{D}_{\text{val}}|}{\arg\max} \text{acc}_{r(h)}(h), \tag{4}$$

and re-validate it at the next rung. We promote the current leader because the reliability of the best harness's accuracy matters most: if its accuracy holds, the promotion confirms a promising harness; if it drops, the lead is attributed to sampling noise. Since the rungs are nested, a promotion only evaluates the questions that the harness has not yet answered.

Second, we add a low-fidelity penalty to node selection, which discounts a high accuracy that is measured on only a few questions. Specifically, we define the penalized accuracy of a harness as a lower confidence bound on its accuracy:

$$\underline{\text{acc}}(h) = \text{acc}_{r(h)}(h) - \lambda_{\text{p}} \cdot \frac{1}{2\sqrt{r(h)}}, \tag{5}$$

where $\frac{1}{2\sqrt{r(h)}}$ is the upper bound on the standard error of the accuracy measured on $r(h)$ questions (Appendix C), and λ_{p} controls the strength of the penalty. For example, with $\lambda_{\text{p}} = 1.5$, the penalty is 7.5 percentage points on 100 questions but only 2.4 points on $1{,}000$ questions, so an accuracy of 76% on 100 questions does not outrank an accuracy of 72% on $1{,}000$ questions. We then use the penalized accuracy in the exploitation term of Equation 3, i.e., $Q(h')$ becomes the highest penalized accuracy among the harnesses in the subtree rooted at h'. As a result, a branch whose best harness has only been validated on a few questions attracts less exploitation. After the evolution, all harnesses are ranked by their penalized accuracy.

3.5 Mixture-of-Harness

Multi-fidelity MCTS returns a tree of harnesses ranked under a single frame budget b. In practice, however, questions arrive with different budget requests: a latency- or cost-sensitive application may afford only a few frames per question, whereas offline analysis may afford over a hundred. Across this range, the relative performance of the evolved harnesses varies with the budget. Harnesses that commit to an answer after a brief survey of the video use a small budget efficiently but benefit little from additional frames, whereas harnesses that continue to gather evidence perform well under a generous budget but degrade substantially when the budget is reduced. As a result, the best harness under a low budget often differs from that under a high budget, so a single top-ranked harness is suboptimal for part of the budget range. We therefore propose a mixture-of-harness (MoH), which routes each question to a harness specialized for its budget (Figure 2, right).

Member Selection. We define a set of budget tiers $\mathcal{K} = \{\text{low}, \text{medium}, \text{high}, \text{xhigh}\}$, where each tier $k \in \mathcal{K}$ corresponds to a range of frame budgets (Appendix D). After evolution, we reuse the harness proposer $\mathcal{M}$ to analyze the evolved harnesses. For each harness h in $\mathcal{T}$, $\mathcal{M}$ inspects its code together with the feedback collected on the validation set during evolution, and characterizes its behavior pattern, including its frame allocation over the video, its stopping behavior (answering early or exhausting the budget), and its reliance on the transcript for evidence localization. Based on this analysis, $\mathcal{M}$ assigns to each tier the harness that best suits its budget range:

$$\{h_k\}_{k\in\mathcal{K}} = \mathcal{M}(\mathcal{T}, \mathcal{K}). \tag{6}$$

Member selection relies only on the harness code and the feedback collected during evolution, so it requires no additional harness executions. The selected harnesses may come from different lineages of the harness tree, and the same harness may serve several tiers.

Table 1: Performance on long video understanding benchmarks: LongVideoBench, VideoMME, and LVBench. For VideoMME and LVBench, accuracy is reported both with and without subtitles. For our approach, VidHarness, we report results under low, medium, high, and extra-high (xhigh) inference frame budgets. #Frames indicates the number of frames each method processes. **Bold** indicates the best result and underline the second best.

Method	Model	LongVideoBench		VideoMME (w/o sub)		VideoMME (w/ sub)		LVBench (w/o sub)		LVBench (w/ sub)	
		#Frames	Long (val)	#Frames	Long	#Frames	Long	#Frames	Test	#Frames	Test
Large Multimodal Models											
Qwen-2.5-VL-72B (Bai et al., 2025)		-	-	256	53.2	256	64.4	768	47.3	-	-
GPT-4o (Hurst et al., 2024)		256	60.9	384	65.3	384	72.1	384	30.8	-	-
Gemini-1.5-Pro (Team et al., 2024)		256	58.6	1,233	67.4	1,233	77.4	3,600	33.1	-	-
Gemini-2.0-Flash (Google DeepMind, 2024)		256	45.7	1,233	63.0	-	-	4,037	48.3	-	-
GPT-5 (OpenAI, 2025)		384	64.5	384	67.9	384	78.1	384	60.1	384	66.5
Qwen-3.6-27B (Qwen Team, 2026)		64	65.6	64	68.0	64	79.8	64	55.5	64	63.5
Video Agentic Models											
VideoAgent (Wang et al., 2024)	GPT-4+LaViLa	-	-	24.6	46.4	-	-	25.5	29.3	-	-
VideoTree (Wang et al., 2025b)	GPT-4+LaViLa	-	-	98.0	53.1	-	-	103.2	28.8	-	-
DrVideo (Ma et al., 2025)	GPT-4+LaViLa	-	-	493.2	51.7	493.2	71.7	-	-	-	-
VCA (Yang et al., 2025)	GPT-4o	-	-	18.1	54.2	-	-	20.0	41.3	-	-
MR. Video (Pang & Wang, 2025)	GPT-4o+Gemini-2.0-Flash	2,816	61.6	4,932	61.8	4,932	-	8,074	60.8	-	-
DVD (Zhang et al., 2026b)	GPT-o3+GPT-4.1	2,816	68.6	4,932	67.3	4,932	-	8,074	**74.2**	8,074	**76.0**
VideoSeek (Lin et al., 2026)	GPT-5	29.6	73.5	60.9	70.1	15.9	81.2	92.3	68.4	27.2	76.7
VideoSeek (Lin et al., 2026)	Qwen-3.6-27B	57.5	67.9	123.0	65.0	17.6	79.6	111.5	54.7	45.9	63.3
VidHarness (Ours)											
low	Qwen-3.6-27B	23	67.7	22	62.0	18	78.7	25	49.8	22	62.6
medium	Qwen-3.6-27B	44	69.1	43	65.9	36	80.3	50	54.7	43	63.5
high	Qwen-3.6-27B	68	71.1	67	67.4	66	82.8	70	56.0	69	66.1
xhigh	Qwen-3.6-27B	126	**72.0**	126	**73.1**	126	**83.2**	126	61.1	126	66.7

Routing by Budget. At inference, each question q arrives with a requested frame budget b, for example chosen to meet a latency or cost target. MoH identifies the tier $k \in \mathcal{K}$ whose range contains b and routes the question to the harness of that tier:

$$y = h_k(v, a, q;\ f, b). \tag{7}$$

Since routing depends only on the budget rather than the video or question, it requires no learned router and adds no runtime cost. Section 4.3 reports its accuracy at each budget tier.

4 Experiments

4.1 Experimental Details

Benchmarks. Following the evaluation protocol of VideoSeek (Lin et al., 2026), we evaluate on six long video benchmarks: LVBench (Wang et al., 2025a) (1,549 questions over 103 hour-long videos), the long subset of Video-MME (Fu et al., 2025) (900 questions, average duration 41 minutes), the long-duration validation split of LongVideoBench (Wu et al., 2024) (564 questions, 900 to 3,600 seconds), Video-Holmes (Cheng et al., 2025) (1,837 questions over 270 suspense films), Video-MMMU (Hu et al., 2026) (900 questions on professional lecture videos), and MMVU (Zhao et al., 2025) (1,000 expert-level questions). For LVBench and Video-MME we report both with- and without-subtitle settings where applicable.

Baselines. We compare against two groups of approaches: large multimodal models and video agentic models. Large multimodal models process uniformly sampled frames in a single pass: Qwen-2.5-VL-72B (Bai et al., 2025), GPT-4o (Hurst et al., 2024), Gemini 1.5 Pro (Team et al., 2024), Gemini 2.0 Flash (Google DeepMind, 2024), GPT-5 (OpenAI, 2025), Qwen-3.6-27B (Qwen Team, 2026). Video agentic models actively gather evidence: VideoAgent (Wang et al., 2024), VideoTree (Wang et al., 2025b), DrVideo (Ma et al., 2025), VCA (Yang et al., 2025), MR. Video (Pang & Wang, 2025), DVD (Zhang et al., 2026b), and VideoSeek (Lin et al., 2026) (current state-of-the-art approach). On Video-Holmes we additionally include some reasoning-enhanced video models (Chen et al., 2025a; Li et al., 2025; Feng et al., 2026). For a fair comparison, we reproduce VideoSeek's results with Qwen-3.6-27B to isolate the effect of backbone models.

Implementation Details. We provide the implementation details, including the harness proposer, the evolution hyperparameters, and the budget tiers of the mixture-of-harness, in Appendix D.

Table 2: Comparison on Video-Holmes. The seven reasoning categories are: SR is social reasoning, IMC is intention and motive chaining, TCI is temporal causal inference, TA is timeline analysis, MHR is multimodal hint reasoning, PAR is physical anomaly reasoning, and CTI is core theme inference. For our approach, VidHarness, we report results under low, medium, high, and extra-high (xhigh) inference frame budgets. #Frames denotes the frame usage. **Bold** marks the best performance, and underline marks the second-best.

Method	#Frames	SR	IMC	TCI	TA	MHR	PAR	CTI	Overall
Large Multimodal Models									
Qwen-2.5-VL-32B (Bai et al., 2025)	32	43.2	44.2	31.5	51.0	36.4	31.4	32.2	38.4
SEED-Bench-R1 (Chen et al., 2025a)	32	42.8	35.1	25.6	40.5	29.2	29.9	32.6	33.5
VideoChat-R1 (Li et al., 2025)	32	42.1	38.8	24.5	39.5	29.5	27.8	29.3	33.0
Video-R1 (Feng et al., 2026)	32	48.6	41.7	28.9	34.5	31.0	33.5	35.9	36.5
GPT-4o (Hurst et al., 2024)	32	50.0	49.6	38.8	30.0	44.0	39.2	37.0	42.0
Gemini 1.5 Pro (Team et al., 2024)	185.1	52.1	48.2	34.4	26.0	39.2	46.4	38.9	41.2
Gemini 2.5 Pro (Comanici et al., 2025)	185.1	46.6	49.3	46.9	53.0	40.1	44.3	37.4	45.0
Gemini 2.0 Flash (Google DeepMind, 2024)	185.1	41.8	33.7	23.1	20.5	30.1	26.8	33.7	30.6
Gemini 2.0 Flash Thinking (Google DeepMind, 2025)	185.1	43.4	46.9	43.1	51.0	37.9	43.6	39.3	43.1
GPT-5 (OpenAI, 2025) (Base)	384	47.2	43.4	40.6	53.5	46.3	38.1	39.6	44.1
Qwen-3.6-27B (Qwen Team, 2026)	63.8	68.5	63.0	58.6	59.0	57.5	52.6	50.7	58.9
Video Agentic Models									
VideoSeek (Lin et al., 2026) (GPT-5)	42.7	56.1	43.8	45.0	54.5	46.6	43.3	41.8	47.3
VideoSeek (Lin et al., 2026) (Qwen 3.6 27B)	68.7	60.3	53.3	44.3	49.0	46.1	45.4	47.4	49.6
VidHarness (Ours)									
low	21.4	67.5	60.1	51.3	48.5	55.4	52.6	49.3	55.5
medium	43.3	62.0	**66.7**	59.0	54.0	**57.5**	58.3	52.2	58.7
high	82.0	67.5	62.0	56.0	**62.5**	55.7	60.0	52.6	59.3
xhigh	126.0	**69.2**	65.6	**59.0**	60.0	56.0	**61.9**	**54.4**	**60.8**

4.2 Comparison with State-of-the-Art Methods

Table 1 reports the comparison on LongVideoBench, Video-MME, and LVBench (with and without subtitles). At its highest budget (xhigh, about 128 frames), VidHarness reaches 72.0% on LongVideoBench, 73.1% on Video-MME without subtitles, and 83.2% on Video-MME with subtitles, outperforming all existing baselines and setting a new state of the art on all three settings. On both Video-MME settings, accuracy with subtitles is consistently higher than without, showing that VidHarness makes effective use of the transcribed audio content to assist long video understanding. On LVBench, in contrast, VidHarness falls behind DVD in both the with-subtitle and without-subtitle settings, because DVD processes an order of magnitude more frames per question (8,074 against VidHarness's 126). VideoSeek is the state-of-the-art video agentic model and thus the most direct baseline for VidHarness; against VideoSeek with the same Qwen-3.6-27B backbone, VidHarness shows a clear advantage: on LVBench without subtitles, VideoSeek reaches 54.7% with 111.5 frames, while VidHarness reaches 61.1% with 126 frames and surpasses VideoSeek, 56.0%, with only 70 frames; on LVBench with subtitles, VidHarness reaches 63.5% with 43 frames against VideoSeek's 63.3% with 45.9 frames. VidHarness at its high tier also generally outperforms the Qwen-3.6-27B uniform-sampling baseline across the five settings.

Table 2 shows the results on Video-Holmes, which requires multi-step reasoning over suspense films. VidHarness obtains 60.8% overall at its xhigh tier, surpassing VideoSeek with the same backbone by 11.2 points and the uniform-sampling baseline by 1.3 points, with the largest gains over VideoSeek on the Physical Anomaly Reasoning (+16.5) and Temporal Causal Inference (+14.7) categories. VidHarnessalso outperforms Qwen-3.6-27B with similar frame budgets (59.3% against 58.9%).

4.3 Ablation Studies

We ablate the two central designs of VidHarness, the evolution strategy and the mixture-of-harness, on Video-Holmes, keeping the backbone and the validation set fixed.

Evolution Strategy. The top half of Table 3 compares three evolution strategies: vanilla iterative refinement following Meta-Harness (Lee et al., 2026), which revises harnesses along a single lineage and validates every candidate on the full validation set; MCTS with full validation; and MCTS with uncertainty-aware multi-fidelity validation (MCTS + MF). Confined to a single lineage, vanilla iterative refinement reaches only 57.3% after 75.5h of evolution, below the uniform-sampling baseline (58.9%). MCTS improves the accuracy to 58.4% by exploring multiple lineages, but it still validates every candidate on all 1,000 questions and requires 71.9h. MCTS + MF reduces the validation sam-

Table 3: Ablation studies on Video-Holmes. Top: evolution strategies, where Evolution Time is the total evolution time and Val. Samples is the total number of validation samples consumed during evolution. Bottom: mixture-of-harness (MoH) versus the top-1 harness from evolution at each of the four budget tiers (low, medium, high, and xhigh).

Method	Evolution Time (h)	Val. Samples (K)	#Frames	Video-Holmes
Qwen-3.6-27B (Qwen Team, 2026)	-	-	63.8	58.9
Vanilla (Lee et al., 2026)	75.5	40.0	64.0	57.3
MCTS	71.9	40.0	64.0	58.4
MCTS + MF	34.9	7.8	56.8	59.4
MCTS + MF + MoH (low)	-	-	21.4	55.5
MCTS + MF + Top1 (low)	-	-	21.2	55.2
MCTS + MF + MoH (medium)	-	-	43.3	58.7
MCTS + MF + Top1 (medium)	-	-	40.1	57.5
MCTS + MF + MoH (high)	-	-	56.8	59.4
MCTS + MF + Top1 (high)	-	-	56.8	59.4
MCTS + MF + MoH (xhigh)	-	-	126.0	60.8
MCTS + MF + Top1 (xhigh)	-	-	82.0	59.3

Table 4: Comparison on knowledge-intensive, out-of-domain video benchmarks, Video-MMMU and MMVU.

Method	Video-MMMU		MMVU	
	#Frames	Acc.	#Frames	Acc.
Qwen-3.6-27B (Qwen Team, 2026)	63	85.3	50	74.4
VideoSeek (Lin et al., 2026) (Qwen 3.6 27B)	16.8	79.1	76.0	69.0
VidHarness (Ours, low)	20	**85.8**	22	**76.8**

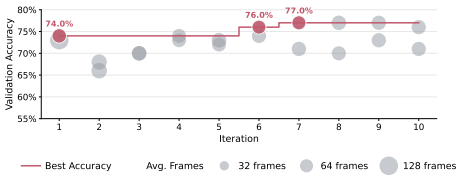


Figure 3: Validation accuracy over evolution iterations. Each circle is a harness designed at that iteration, with its size indicating the average number of frames used, and the red line tracks the best accuracy so far.

ples by $5.1\times$ and the evolution time by $2.1\times$ relative to MCTS, while achieving the highest accuracy of 59.4% with fewer frames (56.8 versus 64.0).

Mixture-of-Harness. The bottom half of Table 3 compares MoH with the top-1 harness from evolution, which is applied to every budget tier. MoH outperforms the top-1 harness at the low (55.5% versus 55.2%), medium (58.7% versus 57.5%), and xhigh (60.8% versus 59.3%) tiers. At the high tier, MoH selects the top-1 harness itself, so the two results are identical (59.4%). Notably, at the xhigh tier, the top-1 harness uses only 82.0 frames on average, whereas the MoH member uses 126.0, showing that a single harness may fail to exploit a larger budget.

4.4 Generalization to Out-of-Domain Benchmarks

The validation set of Section 3.3 and the benchmarks of Tables 1 and 2 all concern long video understanding. To test whether the evolved harnesses generalize beyond this domain, we report results on Video-MMMU and MMVU (Table 4), two knowledge-intensive benchmarks. Even the low tier of VidHarness, which uses 20 and 22 frames, respectively, matches or exceeds the uniform-sampling baseline ($+0.5$ and $+2.4$ points) while using less than a third and less than half of its frames, and outperforms the same-backbone VideoSeek by 6.7 and 7.8 points.

4.5 Evolution Dynamics

Figure 3 shows the validation accuracy of the harnesses designed over the first 10 iterations of evolution. The best accuracy improves from 74.0% at the first iteration to 76.0% at the sixth and 77.0% at the seventh iteration, indicating that the evolution algorithm progressively discovers better harnesses from execution feedback. From the third iteration on, every new harness reaches at least 70%, suggesting that the proposer accumulates useful design knowledge across iterations.

5 Conclusion

We presented VidHarness, a framework that automatically evolves agent harnesses for cost-efficient long video understanding. VidHarness organizes evolution as Monte Carlo tree search with uncertainty-aware multi-fidelity validation, which escapes local optima at a low evaluation cost, and deploys a mixture-of-harness that routes each question to a budget-specialized harness. Experiments show that VidHarness achieves state-of-the-art results on three long video benchmarks.

AI USE STATEMENT

In this work, we used generative AI tools (Claude Code) to implement methods: Claude Code serves as the harness proposer, i.e., the LLM backbone that proposes and edits agent harnesses within VidHarness, and is therefore a core component of our experimental method rather than only a writing aid. We have not used generative AI tools for the other tasks with required disclosure (generating synthetic datasets, developing theoretical models or conceptual frameworks, formulating mathematical claims, proving mathematical claims, assisting with translation, cleaning/reformatting datasets, supporting qualitative or thematic data analysis, or interpreting results); these are not applicable to this work. Additionally, we used generative AI tools for tasks with recommended disclosure: creating and editing scientific figures from our experimental results, and editing the paper to fix grammar and rephrase less-polished passages. We have reviewed all AI-assisted work: all LLM-proposed harnesses were evaluated through our experimental pipeline and validated against held-out data as described in the paper, and all AI-edited text and figures were checked by the authors for accuracy and consistency with the underlying results. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Anthropic. Claude sonnet 4.5. https://www.anthropic.com/news/claude-sonnet-4-5, 2025a.

Anthropic. Claude code. https://claude.com/claude-code, 2025b.

Anthropic. Agent skills. https://www.anthropic.com/news/skills, 2025c.

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

Lingjiao Chen, Matei Zaharia, and James Zou. Frugalgpt: How to use large language models while reducing cost and improving performance. *arXiv preprint arXiv:2305.05176*, 2023.

Tingyang Chen, Shuo Lu, Kang Zhao, Weicheng Meng, Hanlin Teng, Tianhao Li, Chao Li, Xule Liu, Jian Liang, Zhizhong Zhang, et al. Harnessx: A composable, adaptive, and evolvable agent harness foundry. *arXiv preprint arXiv:2606.14249*, 2026a.

Yi Chen, Yuying Ge, Rui Wang, Yixiao Ge, Lu Qiu, Ying Shan, and Xihui Liu. Exploring the effect of reinforcement learning on video understanding: Insights from seed-bench-r1. *arXiv preprint arXiv:2503.24376*, 2025a.

Yukang Chen, Fuzhao Xue, Dacheng Li, Qinghao Hu, Ligeng Zhu, Xiuyu Li, Yunhao Fang, Haotian Tang, Shang Yang, Zhijian Liu, et al. Longvila: Scaling long-context visual language models for long videos. In *International Conference on Learning Representations*, volume 2025, pp. 18227–18246, 2025b.

Yukang Chen, Wei Huang, Baifeng Shi, Qinghao Hu, Hanrong Ye, Ligeng Zhu, Zhijian Liu, Pavlo Molchanov, Jan Kautz, Xiaojuan Qi, et al. Scaling rl to long videos. *Advances in Neural Information Processing Systems*, 38:172842–172870, 2026b.

Junhao Cheng, Yuying Ge, Teng Wang, Yixiao Ge, Jing Liao, and Ying Shan. Video-holmes: Can mllm think like holmes for complex video reasoning? *arXiv preprint arXiv:2505.21374*, 2025.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

Benlei Cui, Ruize Wang, Junjie Li, Jinhao Chen, Longtao Huang, Yinghao Chen, Yuwen Zhai, Jingqun Tang, Ruijian Jia, Weiwei Wu, et al. Metavideoagent: Automated video-agent evolution for long-form video understanding. *arXiv preprint arXiv:2608.04587*, 2026.

Stefan Falkner, Aaron Klein, and Frank Hutter. Bohb: Robust and efficient hyperparameter optimization at scale. In *International conference on machine learning*, pp. 1437–1446. PMLR, 2018.

Yue Fan, Xiaojian Ma, Rujie Wu, Yuntao Du, Jiaqi Li, Zhi Gao, and Qing Li. Videoagent: A memory-augmented multimodal agent for video understanding. In *European Conference on Computer Vision*, pp. 75–92. Springer, 2024.

Kaituo Feng, Kaixiong Gong, Bohao Li, Zonghao Guo, Yibing Wang, Tianshuo Peng, Junfei Wu, Xiaoying Zhang, Benyou Wang, and Xiangyu Yue. Video-r1: Reinforcing video reasoning in mllms. *Advances in Neural Information Processing Systems*, 38:99114–99137, 2026.

Chrisantha Fernando, Dylan Banarse, Henryk Michalewski, Simon Osindero, and Tim Rocktäschel. Promptbreeder: Self-referential self-improvement via prompt evolution. *arXiv preprint arXiv:2309.16797*, 2023.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24108–24118. IEEE, 2025.

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

Google DeepMind. Gemini 2.0 flash thinking. `https://ai.google.dev/gemini-api/docs/thinking`, 2025.

Jianyuan Guo, Zhiwei Hao, Chengcheng Wang, Cheng Fan, Tingzhang Luo, Hongguang Li, Ying Gao, Hefei Mei, Jiankun Peng, Rongjian Xu, et al. From question answering to task completion: A survey on agent system and harness design. *arXiv preprint arXiv:2606.20683*, 2026.

Qingyan Guo, Rui Wang, Junliang Guo, Bei Li, Kaitao Song, Xu Tan, Guoqing Liu, Jiang Bian, and Yujiu Yang. Connecting large language models with evolutionary algorithms yields powerful prompt optimizers. In *International Conference on Learning Representations*, volume 2024, pp. 34133–34156, 2024.

Kairui Hu, Penghao Wu, Fanyi Pu, Wang Xiao, Xiang Yue, Bo Li, Yuanhan Zhang, and Ziwei Liu. Video-mmmu: Evaluating knowledge acquisition from multidisciplinary professional videos. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 27798–27828, 2026.

Shengran Hu, Cong Lu, and Jeff Clune. Automated design of agentic systems. In *International Conference on Learning Representations*, volume 2025, pp. 21344–21377, 2025.

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

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

Kevin Jamieson and Ameet Talwalkar. Non-stochastic best arm identification and hyperparameter optimization. In *Artificial intelligence and statistics*, pp. 240–248. PMLR, 2016.

Zhengyao Jiang, Dominik Schmidt, Dhruv Srikanth, Dixing Xu, Ian Kaplan, Deniss Jacenko, and Yuxiang Wu. Aide: Ai-driven exploration in the space of code. *arXiv preprint arXiv:2502.13138*, 2025.

Levente Kocsis and Csaba Szepesvári. Bandit based monte-carlo planning. In *European conference on machine learning*, pp. 282–293. Springer, 2006.

Yoonho Lee, Roshen Nair, Qizheng Zhang, Kangwook Lee, Omar Khattab, and Chelsea Finn. Meta-harness: End-to-end optimization of model harnesses. *arXiv preprint arXiv:2603.28052*, 2026.

Keliang Li, Yansong Li, Hongze Shen, Mengdi Liu, Hong Chang, and Shiguang Shan. Lenswalk: Agentic video understanding by planning how you see in videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19518–19528, June 2026.

Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. Hyperband: A novel bandit-based approach to hyperparameter optimization. *Journal of machine learning research*, 18(185):1–52, 2018.

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

Jingyang Lin, Jialian Wu, Jiang Liu, Ximeng Sun, Ze Wang, Xiaodong Yu, Jiebo Luo, Zicheng Liu, and Emad Barsoum. Videoseek: Long-horizon video agent with tool-guided seeking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5465–5475, June 2026.

Xinghua Lou, Miguel Lázaro-Gredilla, Antoine Dedieu, Carter Wendelken, Wolfgang Lehrach, and Kevin P Murphy. Autoharness: improving llm agents by automatically synthesizing a code harness. *arXiv preprint arXiv:2603.03329*, 2026.

Elias Lumer, Sahil Sen, Kevin Paul, and Vamse Kumar Subbiah. Recursive agent harnesses. *arXiv preprint arXiv:2606.13643*, 2026.

Ziyu Ma, Chenhui Gou, Hengcan Shi, Bin Sun, Shutao Li, Hamid Rezatofighi, and Jianfei Cai. Drvideo: Document retrieval based long video understanding. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18936–18946. IEEE, 2025.

Alexander Novikov, Ngân Vũ, Marvin Eisenberger, Emilien Dupont, Po-Sen Huang, Adam Zsolt Wagner, Sergey Shirobokov, Borislav Kozlovskii, Francisco JR Ruiz, Abbas Mehrabian, et al. Alphaevolve: A coding agent for scientific and algorithmic discovery. *arXiv preprint arXiv:2506.13131*, 2025.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E Gonzalez, Mohammed Kadous, and Ion Stoica. Routellm: Learning to route llms from preference data. In *International Conference on Learning Representations*, volume 2025, pp. 34433–34448, 2025.

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

Ziqi Pang and Yu-Xiong Wang. Mr. video:" mapreduce" is the principle for long video understanding. *arXiv preprint arXiv:2504.16082*, 2025.

Junjia Qi, Zichuan Fu, Jingtong Gao, Wenlin Zhang, Hanyu Yan, Xian Wu, and Xiangyu Zhao. Llm-as-code agentic programming for agent harness. *arXiv preprint arXiv:2606.15874*, 2026.

Qwen Team. Qwen3.6-27B: Flagship-level coding in a 27B dense model, April 2026. URL `https://qwen.ai/blog?id=qwen3.6-27b`.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *International conference on machine learning*, pp. 28492–28518. PMLR, 2023.

Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Matej Balog, M Pawan Kumar, Emilien Dupont, Francisco JR Ruiz, Jordan S Ellenberg, Pengming Wang, Omar Fawzi, et al. Mathematical discoveries from program search with large language models. *Nature*, 625(7995):468–475, 2024.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *Advances in neural information processing systems*, 36:68539–68551, 2023.

Yu Shang, Yu Li, Keyu Zhao, Likai Ma, Jiahe Liu, Fengli Xu, and Yong Li. Agentsquare: Automatic llm agent search in modular design space. In *International Conference on Learning Representations*, volume 2025, pp. 3841–3865, 2025.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in neural information processing systems*, 36:8634–8652, 2023.

Yan Shu, Zheng Liu, Peitian Zhang, Minghao Qin, Junjie Zhou, Zhengyang Liang, Tiejun Huang, and Bo Zhao. Video-xl: Extra-long vision language model for hour-scale video understanding. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26160–26169. IEEE, 2025.

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

Xi Tang, Jihao Qiu, Lingxi Xie, Yunjie Tian, Jianbin Jiao, and Qixiang Ye. Adaptive keyframe sampling for long video understanding. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 29118–29128. IEEE, 2025.

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

Shulin Tian, Ruiqi Wang, Hongming Guo, Penghao Wu, Yuhao Dong, Xiuying Wang, Jingkang Yang, Hao Zhang, Hongyuan Zhu, and Ziwei Liu. Ego-r1: Agentic chain-of-tool-thought for ultra-long egocentric video reasoning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2026.

Weihan Wang, Zehai He, Wenyi Hong, Yean Cheng, Xiaohan Zhang, Ji Qi, Ming Ding, Xiaotao Gu, Shiyu Huang, Bin Xu, et al. Lvbench: An extreme long video understanding benchmark. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22958–22967. IEEE, 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*, pp. 58–76. Springer, 2024.

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 *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3272–3282. IEEE, 2025b.

Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. Longvideobench: A benchmark for long-context interleaved video-language understanding. *Advances in Neural Information Processing Systems*, 37:28828–28857, 2024.

Guoyang Xu and Hao Chen. Videoharness-rsi: Recursive harness self-improvement for long-video understanding with frozen vision-language models. *arXiv preprint arXiv:2608.24302*, 2026.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V Le, Denny Zhou, and Xinyun Chen. Large language models as optimizers. In *International Conference on Learning Representations*, volume 2024, pp. 12028–12068, 2024a.

John Yang, Carlos Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. Swe-agent: Agent-computer interfaces enable automated software engineering. *Advances in Neural Information Processing Systems*, 37:50528–50652, 2024b.

Zeyuan Yang, Delin Chen, Xueyang Yu, Maohao Shen, and Chuang Gan. Vca: Video curious agent for long video understanding. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 20168–20179. IEEE, 2025.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

Jinhui Ye, Zihan Wang, Haosen Sun, Keshigeyan Chandrasegaran, Zane Durante, Cristobal Eyzaguirre, Yonatan Bisk, Juan Carlos Niebles, Ehsan Adeli, Li Fei-Fei, et al. Re-thinking temporal search for long-form video understanding. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8579–8591. IEEE, 2025.

Xunjian Yin, Xinyi Wang, Liangming Pan, Li Lin, Xiaojun Wan, and William Yang Wang. G\" odel agent: A self-referential agent framework for recursive self-improvement. *arXiv preprint arXiv:2410.04444*, 2024.

Sicheng Yu, Chengkai Jin, Huanyu Wang, Zhenghao Chen, Sheng Jin, Zhongrong Zuo, Xiaolei Xu, Zhenbang Sun, Bingni Zhang, Jiawei Wu, et al. Frame-voyager: Learning to query frames for video large language models. In *International Conference on Learning Representations*, volume 2025, pp. 84154–84179, 2025.

Tongtong Yuan, Xuange Zhang, Bo Liu, Kun Liu, Jian Jin, and Zhenzhen Jiao. Surveillance video-and-language understanding: from small to large multimodal models. *IEEE Transactions on Circuits and Systems for Video Technology*, 35(1):300–314, 2024.

Guibin Zhang, Luyang Niu, Junfeng Fang, Kun Wang, Lei Bai, and Xiang Wang. Multi-agent architecture search via agentic supernet. *arXiv preprint arXiv:2502.04180*, 2025a.

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*, volume 2026, pp. 104223–104294, 2026a.

Jiayi Zhang, Jinyu Xiang, Zhaoyang Yu, Fengwei Teng, Xionghui Chen, Jiaqi Chen, Mingchen Zhuge, Xin Cheng, Sirui Hong, Jinlin Wang, et al. Aflow: Automating agentic workflow generation. In *International Conference on Learning Representations*, volume 2025, pp. 34040–34077, 2025b.

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

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. *Advances in Neural Information Processing Systems*, 38:89863–89895, 2026b.

Yilun Zhao, Haowei Zhang, Lujing Xie, Tongyan Hu, Guo Gan, Yitao Long, Zhiyuan Hu, Weiyuan Chen, Chuhan Li, Zhijian Xu, et al. Mmvu: Measuring expert-level multi-discipline video understanding. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8475–8489. IEEE, 2025.

Kejian Zhu, Zhuoran Jin, Hongbang Yuan, Jiachun Li, Shangqing Tu, Pengfei Cao, Yubo Chen, Kang Liu, and Jun Zhao. Mmr-v: What's left unsaid? a benchmark for multimodal deep reasoning in videos. In *International Conference on Learning Representations*, volume 2026, pp. 67949–67991, 2026.

A Extended Related Work

Long Video Understanding. Existing methods either scale VLMs to the video by extending context windows and compressing visual tokens (Team et al., 2024; Zhang et al., 2024; Chen et al., 2025b; Shu et al., 2025; Bai et al., 2025; Song et al., 2024), or reduce the input by selecting query-relevant keyframes (Tang et al., 2025; Yu et al., 2025; Ye et al., 2025). Another line of work strengthens video reasoning through reinforcement fine-tuning (Feng et al., 2026; Li et al., 2025; Chen et al., 2025a). Closest to our setting, agentic approaches wrap the VLM in a reasoning loop that plans which evidence to gather (Wang et al., 2024; Fan et al., 2024; Wang et al., 2025b; Ma et al., 2025; Yang et al., 2025; Pang & Wang, 2025; Zhang et al., 2026b). Among them, VideoSeek (Lin et al., 2026) equips a reasoning model with a multi-granular seeking toolkit, and LensWalk (Li et al., 2026) lets the model plan its own observation scope and sampling density. Although these agents achieve strong accuracy with few frames, each is a fixed design hand-crafted by experts around a particular model. In contrast, VidHarness evolves the harness automatically and adapts it to the backbone and the frame budget.

Agents and Harnesses. Large language model agents interleave reasoning with actions and tool calls (Yao et al., 2022; Shinn et al., 2023; Schick et al., 2023), and much of their capability is determined by the surrounding *harness*, the runtime that shapes observations, context, control flow, and verification (Yang et al., 2024b; Anthropic, 2025b; Guo et al., 2026). Recent work engineers this layer explicitly (Chen et al., 2026a; Qi et al., 2026; Lumer et al., 2026) or optimizes it automatically (Lou et al., 2026). Among them, Meta-Harness (Lee et al., 2026) optimizes harnesses for text tasks with an outer-loop coding agent. Concurrent work brings harness optimization to video. VideoHarness-RSI (Xu & Chen, 2026) recursively improves context-construction programs for a frozen VLM. MetaVideoAgent (Cui et al., 2026) evolves a modular video agent for a target video distribution: it diagnoses recurring failures against ground-truth evidence, updates only the responsible module in each iteration, and tests modifications on minimal validation tasks derived from these failures. Both largely follow a single improvement trajectory and deploy one evolved agent. In contrast, VidHarness explores a harness tree with multi-fidelity MCTS and assembles harnesses specialized for different frame budgets into a mixture-of-harness.

Evolution Algorithms. With LLMs as mutation operators, evolution algorithms discover programs (Romera-Paredes et al., 2024; Novikov et al., 2025), optimize prompts (Yang et al., 2024a; Guo et al., 2024; Fernando et al., 2023), and design agentic systems. ADAS (Hu et al., 2025) programs new agents in code space with a meta agent, AFlow (Zhang et al., 2025b) applies MCTS to workflow graphs, and related efforts search modular agent spaces or pursue self-referential improvement (Shang et al., 2025; Zhang et al., 2025a; Yin et al., 2024; Zhang et al., 2026a). Tree search over code also drives automated machine-learning engineering (Jiang et al., 2025). However, most of these methods evaluate every candidate in full on tasks where evaluation is cheap. Since evaluation on long videos is expensive, VidHarness combines MCTS (Kocsis & Szepesvári, 2006) with multi-fidelity evaluation in the spirit of successive halving and Hyperband (Jamieson & Talwalkar, 2016; Li et al., 2018; Falkner et al., 2018), where every arm is a harness program. Finally, our mixture-of-harness is related to mixture-of-experts (Jacobs et al., 1991; Shazeer et al., 2017) and LLM routing (Chen et al., 2023; Ong et al., 2025), which select experts or models based on the input content. In contrast, the mixture-of-harness selects among harnesses built around one frozen VLM based on the requested frame budget.

B Algorithm of Uncertainty-Aware Multi-Fidelity MCTS

Algorithm 1 summarizes the evolution procedure of Section 3.4. The harness tree is initialized with a virtual root. In each iteration, node selection descends from the root by the UCT score in Equation 3 until it reaches a node that can be further expanded. The harness proposer then expands the selected node with new harnesses, each of which is validated on the first rung and scored by its penalized accuracy in Equation 5. Backpropagation updates the visit counts and the exploitation terms of the ancestors of the new harnesses. Finally, the promotion stage selects the most accurate harness that has not been validated on the full validation set (Equation 4), re-validates it at the next rung, and updates its penalized accuracy together with the exploitation terms of its ancestors. After all iterations, the harnesses in the tree are ranked by their penalized accuracy.

Algorithm 1 MCTS with uncertainty-aware multi-fidelity validation in VidHarness

Require: validation set $\mathcal{D}_{\text{val}}$; frozen VLM f; frame budget b; harness proposer $\mathcal{M}$; number of iterations I; rungs $r_1 < \cdots < r_L = |\mathcal{D}_{\text{val}}|$; exploration weight λ_{x}; penalty coefficient λ_{p}
Ensure: harnesses in the harness tree $\mathcal{T}$ ranked by penalized accuracy $\underline{\text{acc}}$

initialize $\mathcal{T}$ with a virtual root
for each of the I iterations **do**
  $h \leftarrow$ the virtual root
  while h cannot be further expanded **do** ▷ node selection
    $h \leftarrow$ the child of h with the highest UCT score ▷ Equation 3
  end while
  $\mathcal{H}_{\text{new}} \leftarrow \mathcal{M}(\mathcal{T})$ ▷ node expansion
  for each new harness in $\mathcal{H}_{\text{new}}$ **do** ▷ validation
    add it to $\mathcal{T}$ as a child of h
    validate it on the first r_1 questions of $\mathcal{D}_{\text{val}}$ under budget b
    compute its penalized accuracy ▷ Equation 5
  end for
  update $N(\cdot)$ and $Q(\cdot)$ of all ancestors of $\mathcal{H}_{\text{new}}$ ▷ backpropagation
  $h^\dagger \leftarrow$ the most accurate harness not yet validated on all of $\mathcal{D}_{\text{val}}$ ▷ Equation 4
  validate $h^\dagger$ on the unanswered questions of its next rung ▷ promotion
  update $\underline{\text{acc}}(h^\dagger)$ and $Q(\cdot)$ of its ancestors
end for
return harnesses in $\mathcal{T}$ ranked by $\underline{\text{acc}}$

C Standard Error of Low-Fidelity Accuracy

We derive the bound on the standard error of the low-fidelity accuracy used in Section 3.4. Consider a harness h with true accuracy $p \in [0, 1]$, and let $z_i \in \{0, 1\}$ indicate whether h answers the i-th validation question correctly. Treating the questions as independent draws from the data distribution, each z_i follows a Bernoulli distribution with mean p and variance $p(1 - p)$. The accuracy on the first r questions is the sample mean

$$\text{acc}_r(h) = \frac{1}{r} \sum_{i=1}^{r} z_i, \tag{8}$$

whose expectation is p and whose variance is

$$\text{Var}\big[\text{acc}_r(h)\big] = \frac{1}{r^2} \sum_{i=1}^{r} \text{Var}[z_i] = \frac{p(1-p)}{r}. \tag{9}$$

Since $p(1 - p) = \frac{1}{4} - \big(p - \frac{1}{2}\big)^2 \le \frac{1}{4}$, with equality at $p = \frac{1}{2}$, the standard error is bounded by

$$\text{SE}\big[\text{acc}_r(h)\big] = \sqrt{\frac{p(1-p)}{r}} \le \frac{1}{2\sqrt{r}}. \tag{10}$$

The bound holds for any harness regardless of its accuracy and depends only on the number of questions r. It equals 5.0, 3.2, 2.2, and 1.6 percentage points at $r = 100$, 250, 500, and 1,000, respectively. The low-fidelity penalty in Equation 5 is λ_{p} times this bound, so the penalized accuracy $\underline{\text{acc}}(h)$ is a lower confidence bound that lies λ_{p} worst-case standard errors below the measured accuracy.

D Implementation Details

Evolution runs on the validation set of Section 3.3 with a frame budget of 128 per question. The harness proposer is Claude Code (Anthropic, 2025b) with Claude Opus 5.0, and the trace inspectors run on Claude Sonnet (Anthropic, 2025a). We run for 20 evolution iterations with 2 children harness per iteration, producing 40 candidate harnesses in total. For MCTS, we use an exploration weight $\lambda_{\text{x}} = 0.5$. For multi-fidelity validation, we use rungs of 100, 250, 500, and 1,000 questions, a

penalty coefficient $\lambda_{\mathrm{p}} = 1.5$, and 1 promotion per iteration. We ensemble four harnesses in the mixture-of-harness to handle different budgets, with the low, medium, high, and xhigh tiers centered around 16, 32, 64, and 128 frames, respectively. VLM backbone decoding uses temperature 1.0, top-$p = 0.95$, and top-$k = 20$.