MoHA: Model-Conditioned Harness Adaptation for Long-Video Agents

Anonymous authors
Paper under double-blind review

Abstract

Long-video understanding increasingly relies on agentic systems that actively acquire evidence rather than process fixed video representations. Their execution harnesses must coordinate language-level planning with video-level observation and evidence acquisition, yet are typically fixed across model stacks. We therefore introduce MoHA, a framework for model-conditioned harness adaptation that adapts a common initial execution harness to the specific planner and observer models it hosts. MoHA treats the harness itself as a model-conditioned optimization object: under the same task distribution and initialization, different planner–observer stacks can benefit from different executable harnesses. Starting from a common initial harness, MoHA analyzes failed calibration traces to diagnose model–harness mismatches and adapts both planner-side reasoning support, such as overview, memory, and verification, and observer-side capability assignments, including OCR and ASR specialists. After structural adaptation, MoHA separately calibrates observer execution configurations for the hosted model stack. Across three frozen planner stacks, MoHA improves Video-MME Full accuracy by 6.2–27.1 percentage points and Video-Holmes accuracy by 3.1–20.1 points over the fixed initial harness. Cross-model transfer further shows that the adapted harnesses generalize only partially to other models, highlighting the harness as a model-conditioned optimization layer for long-video understanding. Code is available at `https://anonymous.4open.science/r/MoHA-3CEC/`

1 Introduction

Long-video understanding remains challenging not only because videos contain far more visual content than can be processed densely, but also because useful evidence is sparse and unevenly distributed over time. Recent methods therefore increasingly move beyond processing a fixed video representation and instead actively decide what evidence to acquire and how to acquire it. Some methods adaptively select frames or regions during reasoning (He et al., 2025; Zou et al., 2026; Zeng et al., 2026), while external video agents use retrieval, memory, navigation, and perceptual tools to inspect videos on demand (Wang et al., 2024; Li et al., 2026b; Yin et al., 2026; Lin et al., 2026; Liu et al., 2026). As a result, long-video performance increasingly depends not only on the underlying models, but also on the execution harness that coordinates reasoning, evidence acquisition, and perception.

Existing agentic long-video systems typically pair a model stack with a prescribed execution pipeline that specifies the planner's reasoning support, the routing of perceptual requests, and the processing of video evidence. Recent work has shown that the computation surrounding frozen models can itself be optimized (Zhang et al., 2026). However, improving a harness for one model stack does not establish whether the resulting design is equally well suited to others. A natural alternative is to expose all available capabilities to every stack, but this still leaves the harness model-agnostic and does not account for which forms of reasoning support or perception each stack can use effectively. Planners may differ in how effectively they search for, retain, integrate, and verify evidence, while observers may differ in which perceptual requests they handle reliably and which execution configurations they benefit from. These differences motivate adapting the harness not only to the task, but also to the reasoning and perceptual capabilities of the models it hosts (see Figure 1).

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.

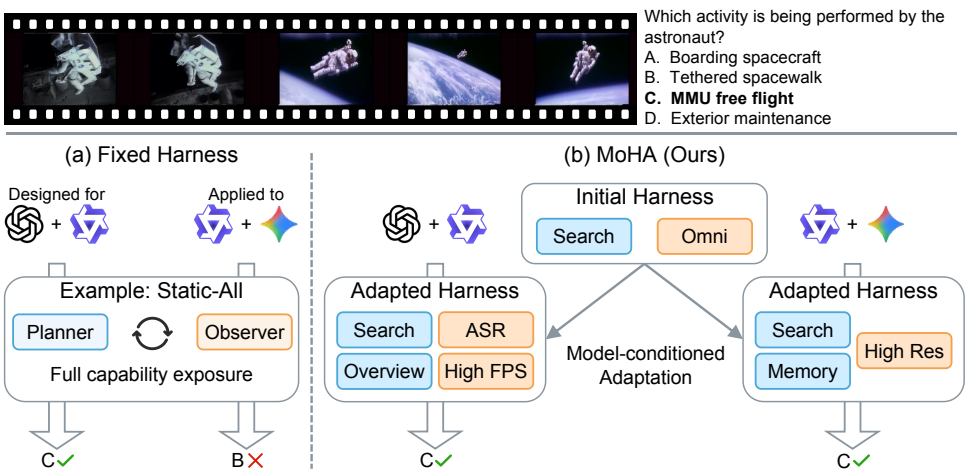


Figure 1: **Motivation for model-conditioned harness adaptation.** (a) A fixed harness is shared across planner–observer stacks despite their different failure patterns; exposing all available capabilities does not necessarily resolve this mismatch. (b) Starting from the same lightweight harness, MoHA adapts planner-facing modules, observer assignments, and execution configurations to each hosted stack.

We therefore introduce **MoHA**, a framework for *model-conditioned harness adaptation* in long-video agents. Starting from a common lightweight harness, MoHA adapts the harness to each planner–observer stack in two stages, while keeping model weights and the underlying runtime fixed. First, it uses failed calibration trajectories to diagnose recurrent model–harness mismatches and propose bounded, localized structural interventions. These interventions modify the reasoning support of the planner, such as overview, memory, or verification, or route suitable observation requests to OCR or ASR specialists. Each candidate intervention is retained only when it improves held-out validation performance under matched execution budgets. After structural adaptation, MoHA freezes the resulting harness structure and separately calibrates observer configurations, including temporal sampling rate and visual input scale. This separation distinguishes which capabilities are exposed to the model stack from how its perceptual capabilities are executed.

We evaluate MoHA across three frozen planner stacks on Video-MME and Video-Holmes. Starting from the same lightweight harness, MoHA consistently improves all three stacks, with gains of 6.2–27.1 percentage points on Video-MME Full and 3.1–20.1 points on Video-Holmes. Model-conditioned adaptation is particularly effective on long videos. Our analysis further reveals that the preferred harness changes meaningfully across models. For example, adding verification improves held-out performance for Qwen3.5-27B but hurts the other two planners, while harnesses adapted to one planner transfer only partially to others. Together, these results show that effective long-video harnesses are shaped not only by the task and available capabilities, but also by the models they host.

Our contributions are summarized as follows:

(1) **Model-conditioned harness adaptation for long-video agents.** We formulate the execution harness as a model-conditioned optimization layer around a frozen planner–observer stack, allowing different model stacks to derive different validated harnesses from the same initialization and adaptation distribution.

(2) **Trace-driven structural adaptation with separate observer calibration.** We introduce a bounded procedure that uses failed calibration trajectories to propose localized changes to planner-side reasoning support and observer capability assignments, retains them through held-out validation, and separately calibrates observer execution configurations after the harness structure is frozen.

(3) **Empirical gains and model-dependent harness behavior.** MoHA consistently improves the evaluated planner stacks over the common initial harness on both Video-MME and Video-Holmes. We further show that failure patterns, intervention effects, and cross-model transfer behavior differ across models, demonstrating that effective long-video harnesses depend on the models they host.

2 Related Work

Long-Video Reasoning. Long-video understanding has progressed from extending or compressing visual context (Zhang et al., 2024a; Chen et al., 2025b; Shen et al., 2025) to stronger temporal and spatiotemporal reasoning (Feng et al., 2025; Li et al., 2025; Chen et al., 2025a). Recent work further learns adaptive evidence acquisition through iterative frame selection or tool-conditioned reasoning (He et al., 2025; Zou et al., 2026; Zeng et al., 2026). Across these approaches, the adaptation target ranges from visual representations and context construction to learned reasoning and evidence acquisition policies. They primarily improve the video model or its learned inference behavior, whereas MoHA keeps the hosted planner–observer stack frozen and adapts the execution system surrounding it.

Agentic Video Understanding. A complementary line organizes long-video inference as an active interaction process, using retrieval and adaptive observation (Wang et al., 2024; 2025a; Yang et al., 2024b; Li et al., 2026b; Lin et al., 2026; Wang et al., 2025b), hierarchical or long-term memory (Yin et al., 2026; Zuo et al., 2025; Long et al., 2025; Yeo et al., 2026), and specialized tools or reasoning roles (Zhang et al., 2025; Li et al., 2026a; Liu et al., 2025; Yan et al., 2026; Qiu et al., 2026). These systems demonstrate that evidence acquisition and execution structure are critical for long-video reasoning, but typically instantiate a particular agent pipeline or interaction policy. More recently, the video-agent system itself has become an optimization target: MetaVideoAgent evolves modular video agents for a target video distribution (Cui et al., 2026), while VideoHarness-RSI searches executable context-construction programs around a frozen vision-language model (Xu & Chen, 2026). MoHA builds on this direction, but conditions harness adaptation on the *hosted model stack*, and explicitly studies adaptation and transfer across different planners and observers.

Agent Harness Optimization. Beyond video, earlier agent systems showed that interfaces, memory, and execution orchestration can substantially shape fixed-model behavior (Yao et al., 2023; Packer et al., 2023; Yang et al., 2024a; Wu et al., 2023). More recent work directly optimizes the computation surrounding language models (LMs), including modular programs built around LMs, agent functions, prompts, and compound AI systems (Khattab et al., 2024; Zhang et al., 2024b; Yuksekgonul et al., 2025; Agrawal et al., 2026). Other approaches externalize or search execution logic and harness structure (Pan et al., 2026; Lee et al., 2026), while Self-Harness uses execution traces to identify model-specific weaknesses and validate localized harness modifications (Zhang et al., 2026). These methods mainly optimize language-side execution components. MoHA extends this model-conditioned perspective to long-video agents, where adaptation additionally spans learned perceptual observers and their execution configurations.

3 Preliminaries

Long-video agents combine reasoning models with mechanisms for retrieval, perception, memory, and iterative evidence acquisition (Wang et al., 2024; Yin et al., 2026; Li et al., 2026b; Lin et al., 2026). As these systems actively interact with video content, their behavior depends not only on the underlying models but also on the execution harness that coordinates reasoning and perception. We therefore treat the harness as an explicit design object.

Let $\mathcal{M} = (M_p, \mathcal{O})$ denote a frozen planner–observer model stack, where M_p is the planner responsible for evidence acquisition and final answering, and $\mathcal{O}$ is the pool of available perceptual backends. The pool contains a general-purpose multimodal observer and may additionally contain specialist backends.

We write the adaptable harness configuration as $H = (S, c)$, where S denotes the *harness structure* and c denotes the observer execution configuration. The Harness Runtime $\mathcal{R}$ is fixed throughout adaptation and is therefore omitted from the variable harness notation. For a video question x, the resulting agent produces $\hat{y} = \mathrm{Run}(\mathcal{M}, H, \mathcal{R}; x)$.

System overview. At a high level, a long-video agent alternates between planning and evidence acquisition. The planner uses the harness to locate relevant video regions and request evidence from selected regions. The structure S determines the planner support that is available and which

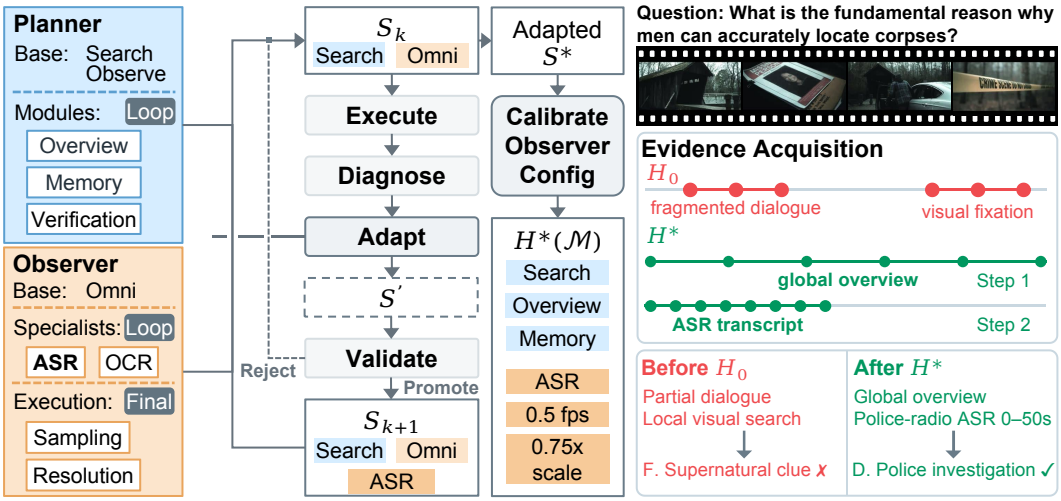


Figure 2: **Overview of MoHA.** Starting from a shared initial harness $H_0 = (S_0, c_0)$, MoHA collects calibration traces, diagnoses recurrent model–harness mismatches, and iteratively proposes and validates localized structural interventions. Planner-facing modules and observer capability assignments are adapted while c_0 remains fixed, producing the final adapted structure S^*. MoHA then freezes S^* and calibrates the observer execution configuration, yielding the final model-conditioned harness $H^*(\mathcal{M}) = (S^*, c^*)$. Right: an example in which global overview and ASR expose previously missing evidence and correct the initial answer.Loop denotes components adapted during structural adaptation; Final denotes post-structural execution calibration.

perceptual capabilities can handle observation requests, while the configuration c determines how a selected backend processes the video.

Planner-facing structure. The planner-facing portion of S specifies the harness-level support available to the planner during iterative evidence acquisition and reasoning. Such support may affect how the agent locates relevant content, organizes and retains acquired evidence, or re-examines evidence before producing a final answer. At minimum, a long-video harness must allow the planner to identify relevant video content and request targeted perceptual evidence; richer harnesses may provide additional support around these basic interactions.

We represent an observation request by a video region W_t and a typed observation goal g_t, specifying respectively *where* to inspect and *what evidence* is sought. The planner determines (W_t, g_t), but does not choose the perceptual backend or low-level execution settings.

Observer-facing structure and execution. The harness uses the observation goal to route each request to a perceptual backend available under S. The configuration c then determines how the selected backend processes the requested video region, including settings such as temporal sampling and visual input scale. This separates *which perceptual capabilities are exposed and assigned* from *how the selected backends are executed*.

Harness Runtime. The fixed Harness Runtime $\mathcal{R}$ executes planner operations, dispatches observation requests, maintains intermediate state and execution traces, and enforces inference budgets.

4 Model-Conditioned Harness Adaptation

Given a frozen model stack $\mathcal{M}$, MoHA adapts a common initial harness using a calibration set $\mathcal{D}_{\text{cal}}$ and a video-disjoint held-out validation set $\mathcal{D}_{\text{val}}$, neither of which contains benchmark test examples. All model stacks start from the same initial harness $H_0 = (S_0, c_0)$, where S_0 is the initial harness structure and c_0 is the common observer execution configuration.

MoHA separates adaptation into two stages. It first adapts the structure S, which determines the planner-facing support and observer-facing capabilities available to the hosted model stack, while keeping c_0 fixed. After the structure is frozen, MoHA separately calibrates the observer execution configuration c. Model weights and the Harness Runtime remain fixed throughout. Figure 2 summarizes the overall procedure. For readability, we omit the dependence on $\mathcal{M}$ from intermediate notation and make it explicit again for the final harness.

4.1 Structural Harness Adaptation

The structural stage determines which additional planner-facing support and observer-facing capabilities are useful for the hosted model stack. Starting from S_0, MoHA uses failed calibration trajectories to propose localized structural changes and retains a change only when it improves held-out validation performance.

Adaptation space. The initial structure S_0 instantiates the minimal planner–harness interaction described in Section 3 with two fixed operations. `search` identifies candidate temporal regions relevant to the current question, while `observe` requests targeted evidence from a selected region. No additional planner-facing support is initially active, and all observation requests are handled by a general-purpose multimodal observer.

MoHA draws structural harness updates from a small, fixed intervention catalog $\mathcal{I}$:

$$\mathcal{I} = \{\texttt{overview}, \texttt{memory}, \texttt{verification}, \texttt{OCR}, \texttt{ASR}\}. \tag{1}$$

Planner-facing interventions may activate `overview`, `memory`, or `verification`, while observer-facing interventions may route eligible text- or speech-oriented observation requests to OCR or ASR specialists. Each intervention modifies a single previously inactive component. The base `search` and `observe` operations remain unchanged, and structural updates are additive in our implementation.

These components address different recurring failure patterns. For example, a planner that repeatedly enters localized search without first establishing useful global context may benefit from `overview`, whereas one that acquires useful evidence but later loses access to it may instead benefit from `memory`. MoHA does not redesign these components themselves; it determines which of them should be exposed to a particular model stack.

At round k, let $\mathcal{I}_k \subseteq \mathcal{I}$ denote the interventions still available under the current harness state, excluding components that are already active or unavailable.

Trace-driven proposal. At round k, MoHA executes the current structure S_k with the fixed configuration c_0 on $\mathcal{D}_{\text{cal}}$ and collects failed execution trajectories. For each failed trajectory τ_i, a structured LLM judge analyzes the question, acquired evidence, interaction context, execution trajectory, and final prediction. It assigns a failure type from a bounded taxonomy (see Table 5) and identifies the trajectory steps supporting its diagnosis.

For non-observer failures, the judge may recommend at most one currently available intervention when the diagnosis and recorded trace support a clear structural change; otherwise, it makes no recommendation. Failure types do not map deterministically to interventions: the same failure category may lead to different recommendations depending on the trajectory.

Observer failures undergo an additional diagnostic probe before MoHA considers adding an OCR or ASR specialist. For example, if the planner localizes a relevant segment and requests speech evidence but the general-purpose observer fails to recover the dialogue, MoHA does not immediately add ASR. Instead, it repeats the same request on the same region with the same observer under alternative execution settings (e.g., a different frame rate or visual input scale). If any alternative recovers the requested evidence, the failure can be resolved without introducing a specialist, so no specialist recommendation is made. If none of the feasible alternatives succeeds, the corresponding specialist may then be recommended. These alternative settings are used only for diagnosis, while c_0 remains fixed during structural adaptation. Full details of the failure taxonomy and diagnostic probe are provided in Appendix A.

Aggregation, validation, and update. MoHA aggregates trajectory-level recommendations using equal votes. If round k contains M_k failed trajectories, the support for candidate a is

$$s_k(a) = \sum_{i=1}^{M_k} \mathbf{1}[a_i = a], \qquad a \in \mathcal{I}_k. \tag{2}$$

Only valid intervention recommendations are counted. Candidates with nonzero support are ranked by $s_k(a)$, with a fixed deterministic tie-breaking rule. The aggregated failure statistics are used only for analysis and do not affect the ranking.

The ranked candidates are then evaluated sequentially on $\mathcal{D}_{\text{val}}$. Let $U_{\text{val}}(S; c)$ denote validation accuracy for structure S under observer configuration c. For candidate a, $S_k \oplus a$ denotes the structure obtained by applying that localized intervention while leaving all other components unchanged. A candidate is promoted only if

$$U_{\text{val}}(S_k \oplus a; c_0) - U_{\text{val}}(S_k; c_0) > \epsilon. \tag{3}$$

Thus, diagnosis determines which modifications are worth testing, not which modification is accepted. For example, even if `overview` is repeatedly recommended across failed trajectories, it is rejected if its held-out gain does not exceed ϵ.

The first candidate satisfying the criterion becomes S_{k+1}, so at most one structural intervention is promoted per round. Rejected candidates are not reconsidered under the same structure. Structural adaptation stops when the round budget is exhausted, no admissible intervention receives support, or no supported candidate passes held-out validation. We denote the resulting structure by S^*.

4.2 Observer Configuration Calibration

Structural adaptation determines *which* capabilities the harness should contain; this second stage determines *how* the resulting observer system should be executed. With S^* fixed, MoHA evaluates a bounded candidate set $\mathcal{C}$ of observer execution configurations, varying settings such as temporal sampling and visual input scale under matched sensory budgets. Planner-side modules and observer capability assignments remain unchanged throughout this stage.

MoHA selects

$$c^* = \arg\max_{c \in \mathcal{C}} U_{\text{val}}(S^*; c), \tag{4}$$

and forms the final model-conditioned harness

$$H^*(\mathcal{M}) = (S^*, c^*). \tag{5}$$

Together with the fixed Harness Runtime, $H^*(\mathcal{M})$ is frozen before benchmark evaluation. Further adaptation details are provided in Appendix A.

5 Experiments

We evaluate MoHA across multiple planner–observer stacks on long-video understanding benchmarks. Beyond overall performance, we further analyze how the adapted harnesses behave across models and which components of the adaptation process contribute to the gains.

5.1 Experimental Setup

Datasets and evaluation protocol. We use the Video-Holmes training split (Cheng et al., 2025) as an independent adaptation source, with a fixed video-disjoint split of 64 calibration and 32 held-out validation examples shared across all model stacks. No benchmark test examples are used during adaptation. After adaptation, each harness is frozen and evaluated on Video-MME-v1 (Fu et al., 2025) and the Video-Holmes test set. Controlled transfer and ablation studies use Video-MME-v1 Long. Video-MME-v1 is evaluated without provided subtitles; audio remains available to the harness for speech evidence acquisition.

Table 1: **Overall performance under matched execution budgets.** Initial uses the common harness H_0; Shared adapts one harness from evidence pooled across model stacks; MoHA adapts a separate harness for each hosted stack. Short, Medium, Long, Full, and Video-Holmes report accuracy (%). *Frames/Q* is the average number of sampled video frames consumed per question on Video-MME-v1. Bold indicates the best result within each planner block.

Planner	Harness	Video-MME-v1					Video-Holmes
		Short	Medium	Long	Full	Frames/Q ↓	
GPT-5.5	Initial H_0	74.3	74.4	66.2	71.7	174.3	47.7
	Shared	**81.4**	**78.0**	72.1	77.2	164.9	**52.1**
	MoHA	78.8	77.7	**77.3**	**77.9**	**98.2**	51.9
Qwen3.5-27B	Initial H_0	72.4	65.6	53.9	64.0	106.1	17.9
	Shared	**78.9**	59.1	53.7	63.9	85.6	23.2
	MoHA	78.3	**72.2**	**65.2**	**71.9**	**60.6**	**38.0**
Qwen3.5-9B	Initial H_0	43.2	31.2	22.8	32.4	33.7	26.3
	Shared	67.2	**62.0**	42.1	57.1	53.8	**30.9**
	MoHA	**74.6**	52.7	**51.1**	**59.5**	**28.0**	29.4

Model stacks. We evaluate frozen planner–observer stacks $\mathcal{M} = (M_p, \mathcal{O})$, all starting from the same harness H_0. For planner-conditioned adaptation, we compare GPT-5.5 (OpenAI, 2026), Qwen3.5-27B, and Qwen3.5-9B (Qwen Team, 2026) with Qwen3-Omni-30B-A3B (Xu et al., 2025b) as the observer. For observer-conditioned transfer, we fix GPT-5.5 and compare Qwen3-Omni-30B-A3B with Qwen2.5-Omni-7B (Xu et al., 2025a). All non-varied components and execution budgets are held fixed.

Baselines. *Initial* uses the common harness H_0 without adaptation. For *Shared*, all model stacks jointly adapt a single harness. At each adaptation round, the stacks run under the same current harness, and their calibration evidence is pooled to propose harness updates, which are then evaluated using validation evidence pooled across the stacks. This produces one adapted harness that is shared by all model stacks. *MoHA* instead adapts a separate harness $H^*(\mathcal{M})$ for each hosted stack.

Implementation and budgets. All controlled comparisons use at most 16 planner calls and the same per-observation limit of 128 sampled frames. For Video-MME-v1, we additionally report the average number of sampled video frames consumed per question as a measure of realized visual cost. Additional adaptation settings, observer configuration candidates, and implementation details are provided in Appendix B.

5.2 Overall Performance

Table 1 compares the common initial harness, the jointly adapted Shared harness, and model-conditioned MoHA. On Video-MME-v1 Full, MoHA improves over H_0 by 6.2, 7.9, and 27.1 percentage points for GPT-5.5, Qwen3.5-27B, and Qwen3.5-9B, respectively. The gains are larger on the Long subset, reaching 11.1, 11.3, and 28.3 points.

Shared adaptation substantially improves some stacks, indicating that some harness modifications transfer across models. However, it does not consistently recover the benefit of stack-specific adaptation. On Video-MME-v1 Long, MoHA exceeds Shared by 5.2 points for GPT-5.5, 11.5 for Qwen3.5-27B, and 9.0 for Qwen3.5-9B. Shared can nevertheless match or outperform MoHA on individual shorter-video subsets, while the advantage of model-specific adaptation is most consistent on the Long subset.

MoHA also improves over H_0 on the Video-Holmes test set for all three stacks, by 4.2, 20.1, and 3.1 points for GPT-5.5, Qwen3.5-27B, and Qwen3.5-9B, respectively. Notably, these gains do not come from consuming more sampled video frames. On Video-MME-v1, MoHA reduces Frames/Q by 43.7%, 42.9%, and 16.9% relative to H_0 for the three planners, respectively, while improving Full accuracy in all cases. Compared with Shared, MoHA likewise achieves higher Full accuracy

with fewer sampled frames for every stack. Together, these results show that model-conditioned adaptation can improve task performance while reducing realized visual sampling cost.

5.3 Cross-Model Harness Transfer

The overall gains do not by themselves establish that the adapted harness depends on the hosted model: different adaptation runs could simply discover equivalent generally useful configurations. We therefore transfer each frozen H^* to another model without further adaptation. For planner-conditioned transfer, we hold the general-purpose observer and specialist pool fixed and replace only the planner. For observer-conditioned transfer, we fix GPT-5.5 and replace only the general-purpose Omni observer.

Table 2: **Cross-model harness transfer on Video-MME-v1 Long.** Rows denote the model used at evaluation time, while each $H^*_{\cdot}$ column denotes a harness adapted for the model indicated by its subscript and transferred without further adaptation. Diagonal entries therefore correspond to matched model–harness pairs, whereas off-diagonal entries measure cross-model transfer. Initial denotes the shared starting harness H_0. Bold indicates the matched adapted harness in each row.

(a) Planner-conditioned transfer

Planner	Initial	H^*_{GPT}	H^*_{Q27B}	H^*_{Q9B}
GPT-5.5	66.2	**77.3**	71.6	67.7
Qwen3.5-27B	53.9	50.4	**65.2**	57.0
Qwen3.5-9B	22.8	41.9	35.5	**51.1**

(b) Observer-conditioned transfer

Observer	Initial	$H^*_{\mathrm{Q2.5O}}$	H^*_{Q3O}
Qwen2.5-Omni	50.4	**57.8**	54.7
Qwen3-Omni	66.2	67.3	**77.3**

Table 2 shows that the matched harness performs best in every completed planner-transfer comparison. For example, Qwen3.5-27B reaches 65.2 with H^*_{Q27B}, compared with 50.4 under H^*_{GPT} and 57.0 under H^*_{Q9B}. Observer-conditioned transfer shows the same pattern: the matched harness performs best for both Omni observers, indicating that model dependence is not limited to the planner.

At the same time, transfer is only partially successful and can be asymmetric. For example, H^*_{GPT} raises Qwen3.5-9B from 22.8 under H_0 to 41.9, yet lowers Qwen3.5-27B from 53.9 to 50.4. Thus, an adapted harness is not simply better or worse in isolation: its effect depends on the model that hosts it.

5.4 Ablation of Adaptation Surfaces

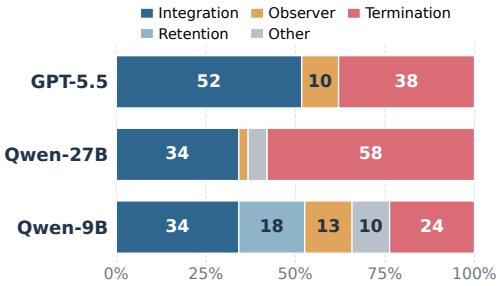
We next ask whether MoHA's gains can be reproduced by simply exposing more capabilities, and which adaptation surfaces contribute to the final result. *Static-All* enables all available planner-side modules and specialist observer assignments without model-conditioned structural selection, while retaining the same observer configuration calibration stage.
We further remove each adaptation surface individually under matched data, models, and budgets. Static-All remains 1.6 points below MoHA; removing planner-side adaptation causes the largest drop (-8.2 points), followed by specialist adaptation (-5.2) and observer configuration calibration (-1.4).

Table 3: **Ablation of MoHA adaptation surfaces for the GPT-5.5/Qwen3-Omni stack on Video-MME-v1 Long.**

Variant	Accuracy
Initial H_0	66.2
Static-All	75.7
w/o planner-side adaptation	69.1
w/o specialist adaptation	72.1
w/o observer config calibration	75.9
MoHA	**77.3**

5.5 Analysis of Model-Conditioned Adaptation

We finally examine how model dependence appears within the adaptation process. Figure 3 compares failure patterns under the shared initial harness with the validation effects and final outcomes of subsequent structural interventions.



(a) Failure profiles under H_0.

	GPT-5.5	Qwen-27B	Qwen-9B
Overview	$-6.2\times$	$+21.9^{(1)}$	$+6.2^{(2)}$
Memory	$+4.7^{(1)}$	$+1.6\times$	$+7.8^{(1)}$
Verify	$-21.9\times$	$+3.1^{(2)}$	$-26.6\times$
ASR	$+3.1^{(2)}$	–	$+3.1^{(3)}$
OCR	–	–	$-10.9\times$
FPS	0.5	0.5	2.0
Scale	0.50	0.75	0.75

(b) Intervention effects and selected harnesses.

Figure 3: **Analysis of model-conditioned harness adaptation.** (a) Failure profiles over unsuccessful calibration trajectories under the common initial harness H_0. (b) Validation effects of tested interventions and final harness choices. Values are accuracy changes in percentage points; superscripts indicate promotion order and $\times$ denotes rejected candidates. Final rows report the selected FPS and input scale.

Different models fail differently under the same initial harness. Figure 3a considers only unsuccessful trajectories produced under H_0, before any model-specific adaptation. For visualization, trajectories that do not produce a valid final answer due to step-budget termination, final-output parsing failure, a missing recognizable answer, or abstention are grouped as *Termination*; infrequent semantic diagnosis categories are merged into *Other*. Termination accounts for 38%, 58%, and 24% of unsuccessful trajectories for GPT-5.5, Qwen3.5-27B, and Qwen3.5-9B, respectively. The semantic composition of the remaining failures also differs across planners, showing that the same initial harness exposes different weaknesses for different hosted models.

The same intervention can reverse its effect across models. Figure 3b combines the held-out validation effect of each tested intervention with the resulting promotion path and final observer configuration. Each value reports the accuracy change relative to the current harness state at the round in which that candidate is evaluated. Overview, for example, changes validation accuracy by -6.2 points for GPT-5.5, but by $+21.9$ for Qwen3.5-27B and $+6.2$ for Qwen3.5-9B. Verification instead improves Qwen3.5-27B by $+3.1$ points while reducing GPT-5.5 and Qwen3.5-9B by 21.9 and 26.6 points, respectively. Model dependence therefore appears not only in effect magnitude but even in effect direction: a capability that benefits one hosted model can be harmful for another. This motivates held-out validation rather than unconditional capability addition. A positive change can still be rejected when it does not exceed the promotion margin, as with Memory for Qwen3.5-27B ($+1.6$ points).

The resulting harnesses consequently differ in both structure and perceptual execution. GPT-5.5 promotes Memory followed by ASR and selects 0.5 FPS with input scale 0.5. Qwen3.5-27B promotes Overview followed by Verification and selects 0.5 FPS with scale 0.75. Qwen3.5-9B promotes Memory, Overview, and ASR and selects 2.0 FPS with scale 0.75. Together with the cross-model transfer results, these differences show that model conditioning affects both which capabilities are exposed and how the resulting perceptual system is executed.

6 Conclusions and Future Work

We introduced MoHA, a framework for adapting the execution harness of a long-video agent to its frozen planner–observer stack. Across three planner stacks, MoHA improves over the common initial harness on both Video-MME and Video-Holmes, while cross-model transfer and intervention analyses show that effective harness configurations depend on the models they host. These results support treating the execution harness as a model-conditioned adaptation target rather than a fixed component of long-video agents. Future work could explore automatic harness-component discovery, joint model–harness co-adaptation, and broader evaluation across models, domains, and multimodal agent tasks.

AI USE STATEMENT

Generative AI tools were used as assistive tools during this work, primarily for improving the clarity and presentation of author-developed ideas, providing feedback on experimental design, assisting with code editing and debugging, summarizing relevant literature, and polishing the manuscript and scientific figures. They were also used to assist in organizing and interpreting experimental results, with all analyses and conclusions independently reviewed and verified by the authors. The central research question, core methodological ideas, and main scientific contributions of this work were developed by the authors rather than generated by AI tools. All experimental decisions, implementations, reported results, and final claims were reviewed and validated by the authors, who take full responsibility for the content of the paper.

REPRODUCIBILITY STATEMENT

We provide the information necessary to reproduce our experiments throughout the main paper and appendix. The main paper specifies the MoHA adaptation procedure, model stacks, datasets, evaluation protocol, and inference budgets. The appendix further documents the harness components, calibration and validation procedure, prompts, observer configuration calibration, implementation details, and additional experimental settings and analyses. We additionally provide anonymized source code and configuration files in the supplementary materials, including scripts for running calibration, harness adaptation, and evaluation.

REFERENCES

Lakshya A. Agrawal, Shangyin Tan, Dilara Soylu, Noah Ziems, Rishi Khare, Krista Opsahl-Ong, Arnav Singhvi, Herumb Shandilya, Michael J. Ryan, Meng Jiang, Christopher Potts, Koushik Sen, Alexandros G. Dimakis, Ion Stoica, Dan Klein, Matei Zaharia, and Omar Khattab. GEPA: Reflective prompt evolution can outperform reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/0e9e708b6f48e14fd0ac29e167413f76-Abstract-Conference.html.

Yukang Chen, Wei Huang, Baifeng Shi, Qinghao Hu, Hanrong Ye, Ligeng Zhu, Zhijian Liu, Pavlo Molchanov, Jan Kautz, Xiaojuan Qi, Sifei Liu, Hongxu Yin, Yao Lu, and Song Han. Scaling RL to long videos. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025a. URL https://arxiv.org/abs/2507.07966. Also arXiv:2507.07966.

Yukang Chen, Fuzhao Xue, Dacheng Li, Qinghao Hu, Ligeng Zhu, Xiuyu Li, Yunhao Fang, Haotian Tang, Shang Yang, Zhijian Liu, Ethan He, Hongxu Yin, Pavlo Molchanov, Jan Kautz, Linxi Fan, Yuke Zhu, Yao Lu, and Song Han. LongVILA: Scaling long-context visual language models for long videos. In *International Conference on Learning Representations*, 2025b.

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. URL https://arxiv.org/abs/2505.21374.

Benlei Cui, Ruize Wang, Junjie Li, Jinhao Chen, Longtao Huang, Yinghao Chen, Yuwen Zhai, Jingqun Tang, Ruijian Jia, Weiwei Wu, Pengfei Sun, and Haiwen Hong. MetaVideoAgent: Automated video-agent evolution for long-form video understanding. *arXiv preprint arXiv:2608.04587*, 2026. URL https://arxiv.org/abs/2608.04587.

Kaituo Feng, Kaixiong Gong, Bohao Li, Zonghao Guo, Yibing Wang, Tianshuo Peng, Junfei Wu, Xiaoying Zhang, Benyou Wang, and Xiangyu Yue. Video-R1: Reinforcing video reasoning in MLLMs. *arXiv preprint arXiv:2503.21776*, 2025. URL https://arxiv.org/abs/2503.21776. Accepted to NeurIPS 2025.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24108–24118, 2025. URL https://arxiv.org/abs/2405.21075.

Zefeng He, Xiaoye Qu, Yafu Li, Siyuan Huang, Daizong Liu, and Yu Cheng. FrameThinker: Learning to think with long videos via multi-turn frame spotlighting. *arXiv preprint arXiv:2509.24304*, 2025. URL https://arxiv.org/abs/2509.24304.

Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan A, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, Hanna Moazam, Heather Miller, Matei Zaharia, and Christopher Potts. DSPy: Compiling declarative language model calls into state-of-the-art pipelines. In *International Conference on Learning Representations*, 2024.

Yoonho Lee, Roshen Nair, Qizheng Zhang, Kangwook Lee, Omar Khattab, and Chelsea Finn. Meta-Harness: End-to-end optimization of model harnesses. In *Conference on Language Modeling (COLM)*, 2026. URL https://arxiv.org/abs/2603.28052.

Chenglin Li, Qianglong Chen, Feng Han, Yikun Wang, Xingxi Yin, Yan Gong, Ruilin Li, Yin Zhang, and Jiaqi Wang. VideoThinker: Building agentic videollms with llm-guided tool reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Findings*, pp. 8226–8236, 2026a. URL https://openaccess.thecvf.com/content/CVPR2026F/html/Li_VideoThinker_Building_Agentic_VideoLLMs_with_LLM-Guided_Tool_Reasoning_CVPRF_2026_paper.html.

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, 2026b. URL https://arxiv.org/abs/2603.24558. Also arXiv:2603.24558.

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. URL https://arxiv.org/abs/2504.06958.

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)*, 2026. URL https://arxiv.org/abs/2603.20185. Also arXiv:2603.20185.

Xiao Liu, Nayu Liu, Junnan Zhu, Ruirui Chen, Guohui Xiang, Changjian Wang, Kaiwen Wei, Rongzhen Li, and Jiang Zhong. ReTool-Video: Recursive tool-using video agents with meta-augmented tool grounding. *arXiv preprint arXiv:2605.13228*, 2026. URL https://arxiv.org/abs/2605.13228.

Ye Liu, Kevin Qinghong Lin, Chang Wen Chen, and Mike Zheng Shou. VideoMind: A chain-of-lora agent for temporal-grounded video reasoning. *arXiv preprint arXiv:2503.13444*, 2025. URL https://arxiv.org/abs/2503.13444.

Lin Long, Yichen He, Wentao Ye, Yiyuan Pan, Yuan Lin, Hang Li, Junbo Zhao, and Wei Li. Seeing, listening, remembering, and reasoning: A multimodal agent with long-term memory. *arXiv preprint arXiv:2508.09736*, 2025. URL https://arxiv.org/abs/2508.09736. M3-Agent.

OpenAI. GPT-5.5 System Card. OpenAI System Card, 2026. URL https://openai.com/index/gpt-5-5-system-card/.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. MemGPT: Towards LLMs as operating systems. *arXiv preprint arXiv:2310.08560*, 2023. URL https://arxiv.org/abs/2310.08560.

Linyue Pan, Lexiao Zou, Shuo Guo, Jingchen Ni, and Hai-Tao Zheng. Natural-language agent harnesses. *arXiv preprint arXiv:2603.25723*, 2026. URL https://arxiv.org/abs/2603.25723.

Chenhao Qiu, Yechao Zhang, Xin Luo, Shien Song, and Xusheng Liu. VideoSEAL: Mitigating evidence misalignment in agentic long video understanding by decoupling answer authority. *arXiv preprint arXiv:2605.12571*, 2026. URL https://arxiv.org/abs/2605.12571. Accepted to ICML 2026.

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

Xiaoqian Shen, Yunyang Xiong, Changsheng Zhao, Lemeng Wu, Jun Chen, Chenchen Zhu, Zechun Liu, Fanyi Xiao, Balakrishnan Varadarajan, Florian Bordes, Zhuang Liu, Hu Xu, Hyunwoo J. Kim, Bilge Soran, Raghuraman Krishnamoorthi, Mohamed Elhoseiny, and Vikas Chandra. LongVU: Spatiotemporal adaptive compression for long video-language understanding. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 54582–54599, 2025. URL https://proceedings.mlr.press/v267/shen25j.html.

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 (ECCV)*, pp. 58–76, 2024. doi: 10.1007/978-3-031-72989-8_4. URL https://arxiv.org/abs/2403.10517. Also arXiv:2403.10517.

Ziyang Wang, Shoubin Yu, Elias Stengel-Eskin, Jaehong Yoon, Feng Cheng, Gedas Bertasius, and Mohit Bansal. VideoTree: Adaptive tree-based video representation for llm reasoning on long videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3272–3283, 2025a. URL https://arxiv.org/abs/2405.19209.

Ziyang Wang, Honglu Zhou, Shijie Wang, Junnan Li, Caiming Xiong, Silvio Savarese, Mohit Bansal, Michael S. Ryoo, and Juan Carlos Niebles. Active video perception: Iterative evidence seeking for agentic long video understanding. *arXiv preprint arXiv:2512.05774*, 2025b. URL https://arxiv.org/abs/2512.05774.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryen W. White, Doug Burger, and Chi Wang. AutoGen: Enabling next-gen LLM applications via multi-agent conversation. *arXiv preprint arXiv:2308.08155*, 2023. URL https://arxiv.org/abs/2308.08155.

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. URL https://arxiv.org/abs/2608.24302.

Jin Xu, Zhifang Guo, Jinzheng He, Hangrui Hu, Ting He, Shuai Bai, Keqin Chen, Jialin Wang, Yang Fan, Kai Dang, Bin Zhang, Xiong Wang, Yunfei Chu, and Junyang Lin. Qwen2.5-Omni Technical Report. *arXiv preprint arXiv:2503.20215*, 2025a. URL https://arxiv.org/abs/2503.20215.

Jin Xu, Zhifang Guo, Hangrui Hu, Yunfei Chu, Xiong Wang, Jinzheng He, Yuxuan Wang, et al. Qwen3-Omni Technical Report. *arXiv preprint arXiv:2509.17765*, 2025b. URL https://arxiv.org/abs/2509.17765.

Haiyang Yan, Hongyun Zhou, Peng Xu, Xiaoxue Feng, and Mengyi Liu. Symphony: A cognitively-inspired multi-agent system for long-video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026. URL https://arxiv.org/abs/2603.17307. Also arXiv:2603.17307.

John Yang, Carlos E. Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. SWE-agent: Agent-computer interfaces enable automated software engineering. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024a. URL https://arxiv.org/abs/2405.15793.

Zeyuan Yang, Delin Chen, Xueyang Yu, Maohao Shen, and Chuang Gan. VCA: Video curious agent for long video understanding. *arXiv preprint arXiv:2412.10471*, 2024b. URL https://arxiv.org/abs/2412.10471.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023. URL https://arxiv.org/abs/2210.03629.

Woongyeong Yeo, Kangsan Kim, Jaehong Yoon, and Sung Ju Hwang. WorldMM: Dynamic multimodal memory agent for long video reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026. URL https://arxiv.org/abs/2512.02425. Highlight; also arXiv:2512.02425.

Yufei Yin, Qianke Meng, Minghao Chen, Jiajun Ding, Zhenwei Shao, and Zhou Yu. VideoARM: Agentic reasoning over hierarchical memory for long-form video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24042–24051, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Yin_VideoARM_Agentic_Reasoning_over_Hierarchical_Memory_for_Long-Form_Video_Understanding_CVPR_2026_paper.html.

Mert Yuksekgonul, Federico Bianchi, Joseph Boen, Sheng Liu, Pan Lu, Zhi Huang, Carlos Guestrin, and James Zou. Optimizing generative ai by backpropagating language model feedback. *Nature*, 639(8055):609–616, 2025. doi: 10.1038/s41586-025-08661-4.

Xiangyu Zeng, Zhiqiu Zhang, Yuhan Zhu, Xinhao Li, Zikang Wang, Changlian Ma, Qingyu Zhang, Zizheng Huang, Kun Ouyang, Tianxiang Jiang, Ziang Yan, Yi Wang, Hongjie Zhang, Yali Wang, and Limin Wang. Video-o3: Native interleaved clue seeking for long video multi-hop reasoning. *arXiv preprint arXiv:2601.23224*, 2026. URL https://arxiv.org/abs/2601.23224.

Hangfan Zhang, Shao Zhang, Kangcong Li, Chen Zhang, Yang Chen, Yiqun Zhang, Lei Bai, and Shuyue Hu. Self-harness: Harnesses that improve themselves. *arXiv preprint arXiv:2606.09498*, 2026. URL https://arxiv.org/abs/2606.09498.

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*, 2024a. URL https://arxiv.org/abs/2406.16852.

Shaokun Zhang, Jieyu Zhang, Jiale Liu, Linxin Song, Chi Wang, Ranjay Krishna, and Qingyun Wu. Offline training of language model agents with functions as learnable weights. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 60315–60335. PMLR, 2024b.

Xiaoyi Zhang, Zhaoyang Jia, Zongyu Guo, Jiahao Li, Bin Li, Houqiang Li, and Yan Lu. Deep video discovery: Agentic search with tool use for long-form video understanding. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. URL https://arxiv.org/abs/2505.18079. Also arXiv:2505.18079.

Junbo Zou, Ziheng Huang, Shengjie Zhang, Liwen Zhang, and Weining Shen. VideoBrain: Learning adaptive frame sampling for long video understanding. *arXiv preprint arXiv:2602.04094*, 2026. URL https://arxiv.org/abs/2602.04094.

Jialong Zuo, Yongtai Deng, Lingdong Kong, Jingkang Yang, Rui Jin, Yiwei Zhang, Nong Sang, Liang Pan, Ziwei Liu, and Changxin Gao. VideoLucy: Deep memory backtracking for long video understanding. *arXiv preprint arXiv:2510.12422*, 2025. URL https://arxiv.org/abs/2510.12422.

A Harness and Adaptation Details

We provide implementation details omitted from the main text. Following Section 3, the adaptable harness is $H = (S, c)$, where S denotes the executable harness structure and c the observer execution configuration. The Harness Runtime $\mathcal{R}$ is fixed throughout adaptation and is therefore omitted from the variable harness notation.

A.1 Harness Components and Observer Execution

MoHA operates over a small, fixed component library. The base `search` and `observe` operations are always available; structural adaptation only activates optional planner support or changes observer assignments.

Table 4: Executable components in the MoHA harness. Base operations remain available throughout adaptation, while optional planner modules and specialist assignments may be activated.

Component	Role	Behavior	Adaptable effect
`search`	Base planner operation	Retrieves candidate temporal regions	Always available
`observe`	Base planner operation	Acquires evidence from region W_t for typed goal g_t	Always available
`overview`	Planner support	Provides coarse global video orientation	May be activated
`memory`	Planner support	Preserves bounded information across reasoning steps	May be activated
`verification`	Planner support	Supports explicit re-examination of accumulated evidence	May be activated
Omni observer	Default observer	Handles general observation requests	Remains the fallback backend
OCR	Specialist observer	Handles typed text-evidence requests	Eligible requests may be reassigned from Omni
ASR	Specialist observer	Handles typed speech-evidence requests	Eligible requests may be reassigned from Omni

For an observation step, the planner selects a video support W_t and a typed semantic goal g_t. The active structure S determines the observer o_t: text or speech requests are routed to an active OCR or ASR specialist, respectively, while all remaining requests use the general-purpose observer. The selected backend is executed under configuration c:

$$z_t = \mathrm{Observe}\left(W_t, g_t, o_t; c\right). \tag{6}$$

Thus, the planner determines *where* to inspect and *what* evidence is needed, while the harness determines which available backend executes the request and how it is executed.

MoHA separates structural and execution-level adaptation. During structural adaptation, $c = c_0$ is fixed and only S may change. After the final structure S^* is frozen, observer configuration calibration selects c^* from a bounded candidate set. Temporary execution changes used by the diagnostic probe below do not update c_0.

A.2 Failure Diagnosis

At structural adaptation round k, MoHA executes the current structure S_k with fixed configuration c_0 on the calibration set. For each failed trajectory, a structured LLM judge attributes the failure using the question, acquired evidence, execution trace, and final prediction. Deterministic checks validate output structure, referenced trajectory steps, capability–request consistency, and candidate legality. Runtime or infrastructure failures are handled separately.

The judge uses the bounded taxonomy in Table 5. The taxonomy is diagnostic rather than prescriptive: a failure category does not deterministically map to an intervention.

Non-`unresolved` diagnoses must reference at least one recorded trajectory step. OCR or ASR capability attribution is valid only for an `observer` failure containing a corresponding typed text or speech request.

A.3 Observer Specialist Gating Probe

An `observer` diagnosis does not immediately justify adding OCR or ASR. Instead, MoHA first tests whether the cited general-purpose-observer failure can be rescued by changing only its execution configuration.

The probe re-executes the cited observation while keeping the video asset, source window, selected region, typed observation goal, and general-purpose observer fixed. Planner decisions, search trajectories, and final answer generation are not replayed, and the observer call does not receive the

Table 5: Failure taxonomy used for calibration-trace diagnosis.

Failure	Interpretation
Orientation	The planner fails to establish a useful region of the video before detailed evidence acquisition.
Retrieval	Relevant temporal candidates are not retrieved despite a plausible evidence need.
Candidate selection	A useful candidate is retrieved but not selected or followed up.
Goal specification	The selected region is plausible, but the observation request asks for insufficient or mismatched evidence.
Observer	The region and request are plausible, but the assigned observer does not recover evidence supporting the requested goal.
Evidence retention	Useful acquired evidence is no longer available to later reasoning steps.
Evidence integration	Relevant evidence remains available but is combined or used incorrectly.
Verification	The trajectory contains evidence motivating re-examination, but the planner commits without resolving the remaining conflict.
Unresolved	The trace does not support a reliable attribution to the categories above.

answer label. The probe varies only execution fields, using candidate values from

$$\{0.5, 1.0, 2.0\}\ \text{FPS}, \qquad \{0.5, 0.75, 1.0\}\ \text{visual scale},$$

and $\{\text{temporal}, \text{spatial}, \text{balanced}\}$ execution priority, excluding no-op, infeasible, or media-equivalent configurations. These temporary executions do not modify c_0.

A structured probe judge evaluates whether the baseline and alternative observations support the same requested goal. An *execution rescue* occurs when the baseline fails but at least one alternative succeeds; in this case the trace is not treated as sufficient evidence for introducing a specialist. A *no-rescue* result requires all feasible tested alternatives to fail with valid judgments. Other cases are treated as inconclusive.

Only a no-rescue result makes a matching specialist admissible: OCR additionally requires a typed text request and OCR attribution, while ASR requires a typed speech request and ASR attribution. The specialist must still be recommended at the trajectory level and pass the held-out validation gate described below.

A.4 Structural Adaptation and Promotion

Structural adaptation uses the fixed intervention catalog

$$\mathcal{I} = \{\texttt{overview}, \texttt{memory}, \texttt{verification}, \texttt{OCR}, \texttt{ASR}\}.$$

Updates are additive: once activated, a module or specialist assignment remains active in subsequent rounds.

Each failed calibration trajectory contributes at most one currently admissible intervention recommendation, or makes no recommendation. Invalid, unresolved, null, or illegal proposals contribute no vote. Valid proposals receive equal weight and are ranked by support count, using the candidate identifier as a deterministic tie-break. There is no learned global selector after these trajectory-level recommendations.

For candidate a, let $S_k \oplus a$ denote the structure obtained by applying only that intervention to S_k. The current and candidate structures are evaluated on the same held-out validation examples using the same frozen model stack, fixed configuration c_0, and matched execution and sensory budgets. A candidate is promoted only when

$$U_{\text{val}}(S_k \oplus a; c_0) - U_{\text{val}}(S_k; c_0) > \epsilon. \tag{7}$$

Candidates are tested in ranked order up to the configured per-round limit, and the first candidate passing the gate becomes S_{k+1}. Structural adaptation stops when the round budget is exhausted, no admissible candidate receives support, or none of the tested candidates passes validation.

Table 6: MoHA adaptation hyperparameters.

Hyperparameter	Value
Maximum structural rounds K	6
Candidates tested per round	2
Promotion margin ϵ	0.02
Calibration examples	64
Validation examples	32
Validation repeats	2
Structural-stage target FPS	1.0
Structural-stage visual scale	1.0
Structural-stage priority	balanced
Frame cap	128
Observer sampling candidates	$\{0.5, 1.0, 2.0\}$ FPS
Observer scale candidates	$\{0.5, 0.75, 1.0\}$

A.5 Observer Configuration Calibration

After structural adaptation terminates, the resulting structure S^* is frozen. MoHA then selects the observer execution configuration on the held-out validation set:

$$c^* = \arg\max_{c \in \mathcal{C}} U_{\text{val}}(S^*; c). \tag{8}$$

No planner module or observer assignment changes during this stage. The final harness is

$$H^*(\mathcal{M}) = (S^*, c^*), \tag{9}$$

which is frozen before benchmark evaluation.

In our experiments, $\mathcal{C}$ is the Cartesian product of $\{0.5, 1.0, 2.0\}$ FPS and $\{0.5, 0.75, 1.0\}$ visual input scales with balanced execution priority, under the same frame and visual-token ceilings used by the structural comparisons. Unlike the specialist gating probe, which asks whether one failed observation can be rescued, this stage selects a single execution configuration for the already frozen structure.

A.6 Full Adaptation Algorithm

Algorithm 1 summarizes the complete procedure.

B Experimental and Implementation Details

B.1 Calibration and Validation Split

Adaptation data are partitioned by video identity before harness optimization. We use 64 calibration examples for trace collection, diagnosis, probing, and intervention recommendation, and 32 held-out validation examples for structural promotion and post-structural observer configuration selection. No benchmark test example is used during adaptation.

B.2 Adaptation Hyperparameters

Table 6 summarizes the fixed hyperparameters used across adaptation runs. Structural adaptation uses a common observer configuration and a bounded number of rounds and candidate evaluations; after the structure is frozen, observer calibration searches the listed sampling-rate and visual-scale configurations.

B.3 Model and Runtime Configuration

All model weights remain frozen throughout adaptation. The released configuration specifies the exact planner and observer snapshots, specialist backends, maximum planner steps, observation

Algorithm 1 MODEL-CONDITIONED HARNESS ADAPTATION

Require: frozen model stack $\mathcal{M}$; shared initial harness $H_0 = (S_0, c_0)$; calibration set $\mathcal{D}_{\text{cal}}$; validation set $\mathcal{D}_{\text{val}}$; intervention catalog $\mathcal{I}$; configuration set $\mathcal{C}$; execution budget B; promotion margin ϵ; maximum structural rounds K
Ensure: frozen model-conditioned harness $H^*(\mathcal{M})$

$S \leftarrow S_0$
for $k = 0, \ldots, K-1$ **do**
  $\mathcal{T} \leftarrow \text{COLLECTTRACES}(\mathcal{M}, S, c_0, \mathcal{D}_{\text{cal}}, B)$
  $\mathcal{I}_k \leftarrow \text{ADMISSIBLE}(\mathcal{I}, S)$
  if $\mathcal{I}_k = \emptyset$ **then**
    break
  end if
  $\mathcal{P} \leftarrow \emptyset$
  for all failed trajectories $\tau_i \in \mathcal{T}$ **do**
    $d_i \leftarrow \text{DIAGNOSE}(\tau_i, \mathcal{I}_k)$
    if d_i.failure $=$ `observer` **then**
      $p_i \leftarrow \text{PROBEOBSERVER}(\tau_i, d_i, c_0)$
      $a_i \leftarrow \text{RECOMMEND}(\tau_i, d_i, p_i, \mathcal{I}_k)$
    else
      $a_i \leftarrow \text{RECOMMENDATION}(d_i)$
    end if
    if $a_i \neq \varnothing$ **then**
      $\mathcal{P} \leftarrow \mathcal{P} \cup \{a_i\}$
    end if
  end for
  $\mathcal{Q} \leftarrow \text{RANKBYSUPPORT}(\mathcal{P})$
  if $\mathcal{Q} = \emptyset$ **then**
    break
  end if
  promoted $\leftarrow$ false
  for all $a \in \mathcal{Q}$ in ranked order **do**
    if $\text{VALACC}(S \oplus a, c_0) - \text{VALACC}(S, c_0) > \epsilon$ **then**
      $S \leftarrow S \oplus a$
      promoted $\leftarrow$ true
      break
    end if
  end for
  if $\neg$promoted **then**
    break
  end if
end for
$S^* \leftarrow S$
$c^* \leftarrow \text{CALIBRATEOBSERVERCONFIG}(\mathcal{M}, S^*, \mathcal{C}, \mathcal{D}_{\text{val}}, B)$
$H^*(\mathcal{M}) \leftarrow (S^*, c^*)$
return $\text{FREEZE}(H^*(\mathcal{M}))$

and sensory budgets, video decoding procedure, diagnosis and probe judge models, and API or local inference settings used for each reported condition. Consequently, differences among Initial, Shared, MoHA, transfer, and ablation conditions arise from the execution harness or its controlled removal rather than model fine-tuning.

B.4 DIAGNOSTIC INTERFACE AND PROMPTS

The structured diagnosis interface records the failure category, supporting trajectory-step identifiers, optional OCR/ASR capability attribution, and at most one currently available intervention recommendation. The implementation additionally records confidence and short textual explanations for attribution, residual uncertainty, and recommendation or abstention.

Deterministic validation checks the required field set, legal enum values, confidence range, evidence-step references, capability–request consistency, and candidate legality. Non-`unresolved` diagnoses require recorded supporting steps; OCR attribution requires a typed text request and ASR

Table 7: Comparison with representative work that optimizes the execution system surrounding a frozen or hosted model.

Method	Primary conditioning focus	Adapted object	Distinction relative to MoHA
MetaVideoAgent (Cui et al., 2026)	Target video / task distribution	Modular video-agent design	Searches for an effective agent design for a target distribution; MoHA instead keeps the task distribution and initialization shared and asks how the harness changes with the hosted model stack.
VideoHarness-RSI (Xu & Chen, 2026)	Frozen video-model answering interface	Executable context-construction program	Optimizes the inference program around a fixed interface; MoHA separately exposes planner support, observer assignment, and observer execution configuration and studies how these change across model stacks.
Self-Harness (Zhang et al., 2026)	Model-specific execution traces	Localized harness modifications	Shares the model-conditioned, trace-driven perspective; MoHA extends it to long-video agents with learned perceptual backends and explicit observer-side adaptation.
Natural-Language Agent Harnesses / MetaHarness (Pan et al., 2026; Lee et al., 2026)	General agent execution logic	Harness / execution program	Treats harness logic as an optimization object more broadly; MoHA uses a bounded component space to enable interpretable model-conditioned adaptation and controlled cross-model transfer analysis.
MoHA	**Hosted planner–observer stack**	**Planner modules, observer assignment, and observer execution**	**Uses shared initialization and adaptation data across model stacks and explicitly evaluates whether adapted harnesses transfer across planners and observers.**

attribution requires a typed speech request. For the initial `observer` diagnosis, the intervention recommendation must be null until the specialist gating probe is resolved. Invalid structured outputs receive at most one format-repair attempt without forcing a semantic attribution.

The supplementary implementation provides the exact diagnosis and probe prompts, structured schemas, and validation logic. Neither diagnosis nor probe components receive held-out validation labels or benchmark test examples.

C Relation to Prior Harness and Agent Adaptation

Table 7 summarizes the closest lines of work along the dimensions most relevant to MoHA. The comparison is intended to clarify adaptation targets rather than rank methods.

D Representative Long-Video Agent Results

For literature context, Table 8 reports published Video-MME v1 results from representative long-video agents and tool-using systems. These systems differ in backbone models, audiovisual access, training data, retrieval procedures, and inference budgets, so the numbers should not be interpreted as controlled comparisons. Our primary controlled evidence is the matched Initial–Shared–MoHA evaluation in the main text.

Table 8: Representative reported Video-MME v1 results. Values are taken from the corresponding papers and are provided as literature context rather than a controlled leaderboard.

Method	Long	Full
VideoAgent (Wang et al., 2024)	46.4	–
VideoTree (Wang et al., 2025a)	54.2	60.6
VideoLucy (Zuo et al., 2025)	66.8	72.5
AVP (Gemini 2.5 Pro) (Wang et al., 2025b)	81.9	85.3
LensWalk (o3/GPT-4.1) (Li et al., 2026b)	70.0	–
VideoARM (o3/GPT-4.1) (Yin et al., 2026)	75.3	80.1
VideoSeek (GPT-5) (Lin et al., 2026)	70.1	–
ReTool-Video-9B (Liu et al., 2026)	–	76.6
MoHA (GPT-5.5)	**77.3**	**77.9**
MoHA (Qwen3.5-27B)	**65.2**	**71.9**
MoHA (Qwen3.5-9B)	**51.1**	**59.5**

E LIMITATIONS

MoHA is evaluated on a limited set of planner–observer stacks and long-video benchmarks. Although different stacks benefit from different adapted harnesses in our experiments, broader evaluation across model families, video domains, and multimodal tasks is needed to establish generality. MoHA also searches within a predefined space of planner-facing modules and observer capabilities rather than arbitrary harness programs or dynamically created capabilities.

Adaptation also depends on calibration-trace diagnoses, which may be imperfect for ambiguous failures, and incurs additional inference cost before deployment. Some experiments rely on externally served models that may exhibit run-to-run stochasticity. Finally, MoHA keeps planner and observer parameters frozen and adapts only the surrounding execution harness; joint model–harness co-adaptation remains a promising direction for future work.