MORPH-V: AUTONOMOUS GROWTH OF REUSABLE OBSERVATION PROGRAMS FOR LONG-VIDEO REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-video reasoners adapt their observations to individual questions, yet repeatedly plan observation from scratch when similar requirements recur across videos. This prevents successful observation procedures from being retained and reused, limiting the accumulation of observation capability. To this end, we propose MORPH-V, a two-phase framework that couples coverage-guided program acquisition with applicability-guided reuse through a persistent program library. During coverage-guided acquisition, MORPH-V identifies recurring observation requirements unsupported by the current library, composes candidate programs from typed primitives, verifies them structurally without answer outcomes, and stores the verified programs for reuse. Each newly stored program expands library coverage, redirecting subsequent acquisition toward the remaining unsupported requirements. During applicability-guided reuse, MORPH-V selects an applicable program, grounds its observation requirement in the current video, and executes it to construct evidence for the reasoner. Across five video evaluation populations and two frozen reasoners, MORPH-V improves over the corresponding observation controls. Further analysis reveals rising macro-averaged accuracy across reloaded checkpoints and that fixed deployment programs benefit both reasoners.

1 INTRODUCTION

Long-video reasoners adapt how they gather evidence to the current question (Tang et al., 2025; Sun et al., 2025; Wang et al., 2025a;b). Across videos, similar observation requirements recur in questions involving quoted speech, multi-state comparisons, and action sequences (Wu et al., 2024; Fu et al., 2025; Zhou et al., 2025). Although the relevant evidence changes from video to video, the procedure for locating, selecting, and organizing that evidence can often be reused. Yet current adaptive methods construct such observation procedures for individual inputs and discard them afterward, so each new video effectively requires observation planning from scratch. The challenge is therefore to adapt observation to the current video while retaining successful observation procedures and identifying missing capabilities, allowing a frozen reasoner to improve without parameter updates.

Existing methods expose a gap between adaptation and accumulation. Adaptive selectors optimize observation for the current input, whereas persistent-skill systems retain reusable procedures across inputs (Grand et al., 2024; Wang et al., 2023; Huang et al., 2026; Jiang et al., 2026). What remains missing is a closed loop in which what has already been acquired determines what should be learned next. Our key insight is that a persistent program library can serve both as reusable memory and as an explicit record of covered observation capability. Recurring requirements unsupported by the current library become acquisition targets. Once a verified program is stored, matching inputs become covered, redirecting subsequent acquisition toward the remaining gaps. In this way, reuse and capability growth arise from the same persistent state.

Building on this insight, we propose MORPH-V, a two-phase framework that couples coverage-guided program acquisition with applicability-guided program reuse. Each library entry pairs an observation program with a specification. The specification records when the program applies, how its observation requirement is grounded, and how execution is checked. During acquisition, MORPH-V targets an unsupported recurring observation requirement, composes and verifies a candidate program

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.

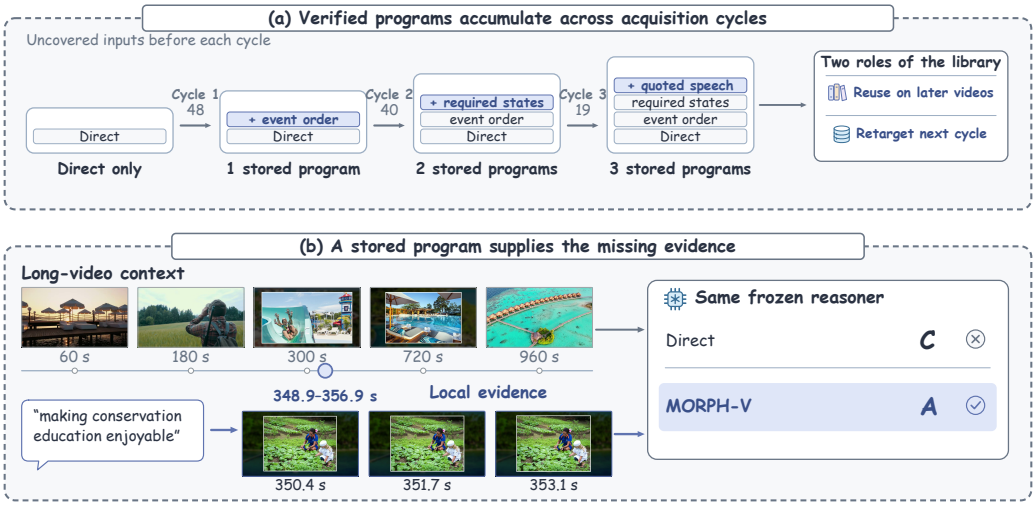

Figure 1: A persistent program library grows through coverage-guided acquisition and supports program reuse. **(a)** Across three held-out recovery cycles, verified programs accumulate in the library, while stored coverage removes supported inputs and redirects acquisition to the next unsupported requirement. The replay uses structural traces only, without answer calls or human edits. **(b)** For a question containing a quoted phrase, Direct misses the relevant event and predicts C, whereas MORPH-V uses a stored program to bind the phrase to 348.9–356.9 seconds, construct local evidence, and predict A with the same frozen reasoner.

without using answer outcomes, and stores the resulting entry to expand library coverage. During reuse, it selects an applicable entry, grounds its observation requirement in the current video, and executes the stored program to construct evidence for the frozen reasoner. Figure 1**(a)** shows how acquiring programs for event order, required states, and quoted speech retains earlier capabilities while coverage redirects later acquisition. Figure 1**(b)** shows how a stored quoted-speech program localizes the source interval and supplies evidence that Direct misses.

We evaluate MORPH-V on program acquisition, library growth, and program reuse. MORPH-V recovers 3/3 held-out observation programs, while accuracy improves progressively as verified programs accumulate in the library, yielding a total gain of 7.78 percentage points from the Direct-only initial state. On LongVideoBench, MORPH-V improves over the underlying frozen reasoners by up to 4.34 percentage points and shows that fixed deployment programs benefit both reasoners. Our contributions are summarized as follows:

- We introduce MORPH-V, a closed-loop framework for acquiring, retaining, and reusing observation programs in long-video reasoning without parameter updates, enabling observation capability to accumulate across tasks.
- We develop a persistent program library that couples coverage-guided acquisition with applicability-guided reuse, using stored coverage to expose missing capabilities and redirect subsequent acquisition.
- We show that MORPH-V acquires missing observation programs, yields progressive accuracy gains as the library grows, and supports reuse across reasoners and evaluation settings.

2 Related Work

2.1 Adaptive Video Reasoning

Long-video VLMs move beyond static sampling by adapting visual evidence to the current query. Some methods select or allocate informative frames using query relevance, temporal coverage, or adaptive exploration (Liu et al., 2025; Tang et al., 2025; Sun et al., 2025; Zhu et al., 2026; Zhang

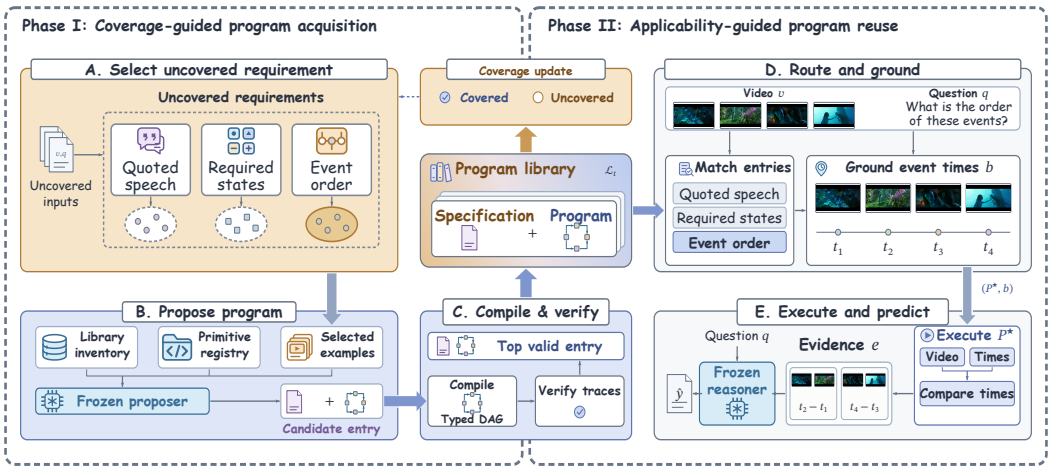

Figure 2: Overview of MORPH-V. **Phase I** (left): MORPH-V selects a recurring requirement not covered by the current library through **(A) target selection**, **(B) specification and graph proposal**, and **(C) compilation and structural verification**. The selected entry is stored, updating coverage for the next acquisition cycle. **Phase II** (right): MORPH-V **(D) routes and grounds** an applicable entry for a new input, then **(E) executes** its program for the frozen reasoner.

et al., 2025b; Peng et al., 2026; Li et al., 2026), while others learn adaptive selectors from supervision or downstream task signals (Buch et al., 2025; Hu et al., 2025a; Qin et al., 2026). Video agents further adapt observation in response to intermediate evidence (Wang et al., 2025a; Zou et al., 2026; Wang et al., 2026; Ma et al., 2026), and hierarchical approaches organize query-relevant information from coarse to fine (Wang et al., 2025b). Despite these advances, most observation policies remain episodic, and evidence-construction procedures are not retained for reuse when similar requirements recur. Model updating can internalize such procedures (Tan et al., 2026; Singh et al., 2026), but requires further updates as data or observation requirements change. This motivates an external mechanism for accumulating observation capability across inputs.

2.2 Learning Executable Programs and Persistent Skills

Program-based approaches construct executable procedures by composing visual modules (Gupta & Kembhavi, 2023) or using executable code that can be revised through interaction feedback (Wang et al., 2024). Persistent-skill systems extend such procedures across tasks. LILO compresses synthesized programs into reusable abstractions, Voyager maintains an executable skill library, and more recent systems distill interaction trajectories into reusable macro-tools, skills, or experiences (Grand et al., 2024; Wang et al., 2023; Huang et al., 2026; Jiang et al., 2026). Another line of work uses closed-loop search to generate, evaluate, and retain workflows or agent code (Hu et al., 2025b; Zhang et al., 2025a; Yin et al., 2025; Zhang et al., 2026a). In multimodal settings, Video-Zero organizes feedback around co-evolving roles (Zhang et al., 2026b). However, these approaches do not use the coverage of an executable program library to select which missing observation procedure should be acquired next. MORPH-V stores each acquired program with a reuse specification; updated coverage directs acquisition toward remaining unsupported requirements.

3 Methodology

Within a fixed typed observation space and primitive registry, MORPH-V grows a persistent library of reusable observation programs. Each entry pairs a **specification** for applicability, grounding, and verification with an **observation program** that constructs evidence. Coverage monitoring, compilation, execution, and routing are fixed; only the frozen proposer generates new compositions.

As shown in Figure 2, Phase I turns a recurring requirement unsupported by the current library into a verified program entry. Storing that entry both makes the program reusable and updates coverage,

redirecting the next acquisition cycle toward the remaining unsupported inputs. Phase II then routes a new video-question input to an applicable entry, grounds its requirement in the video, and executes the stored program for a frozen reasoner. Thus, the same library state links what MORPH-V acquires with what it can later reuse.

3.1 Problem Formulation

We define a video-question input as $x = (v, q)$, where v is a video and q is the corresponding question. The acquisition set $\mathcal{D}_{\text{acq}}$ contains inputs with answer labels withheld. A fixed, answer-independent monitor maps observable question structure to an observation requirement, defined as the evidence needed by a recurring class of questions. The fixed taxonomy maps quoted speech to intervals, required states to slot sets, and event order to event timelines. Inputs sharing a requirement can reuse one observation program, while grounding instantiates that requirement for each video. We denote the resulting interval, semantic-slot set, or event timeline by b. For example, quoted-speech questions require the interval containing the referenced utterance. Grounding locates that interval, after which the program samples frames from it.

The fixed primitive registry Ω contains typed executable primitives, namely atomic operations with declared input and output types. These primitives operate on intervals, semantic-slot sets describing required states or events, and event timelines. For example, they sample frames within a grounded interval or organize event times into pairwise temporal relations. An observation program connects type-compatible primitives into a directed acyclic graph that constructs evidence from v and b.

Before compilation, a specification σ records applicability and grounding rules, a symbolic graph over selected primitives, and the checks, evidence budget, and objective used for verification and selection. The compiler resolves that symbolic graph into an executable observation program P. A stored library entry $\ell = (\sigma, P)$ retains both the reuse contract and its executable program. We denote the persistent library before acquisition cycle t by $\mathcal{L}_t$. The initial library $\mathcal{L}_0$ contains only Direct, the fallback entry excluded from coverage monitoring.

3.2 Phase I: Coverage-Guided Program Acquisition

3.2.1 Coverage Monitoring and Target Selection

At cycle t, an input $x \in \mathcal{D}_{\text{acq}}$ is uncovered if no acquired entry in $\mathcal{L}_t$ applies to it. The monitor groups uncovered inputs first by evidence type and then by their shared observation requirement, retaining groups that meet a fixed recurrence threshold m. The controller selects the largest eligible group and uses its inputs as acquisition examples.

3.2.2 Program Proposal

The frozen proposer receives examples from the selected group, the typed signatures of entries in $\mathcal{L}_t$, and the primitive registry Ω. It returns candidate specifications in a fixed schema whose primitive choices define symbolic program graphs. Each specification also selects a type-compatible structural objective and declares the checks required for verification. The resulting specification-graph pairs are passed to the compiler.

3.2.3 Compilation and Structural Verification

For each proposal, the deterministic compiler resolves every symbolic call in its program graph against Ω. Compilation succeeds only if all calls resolve, connected inputs and outputs have compatible types, the dependency graph is acyclic, and all required structural checks are present. Successful compilation turns the specification-graph pair into an executable candidate entry.

Compilation guarantees a type-correct graph, but not that it can ground the selected requirement and construct the intended evidence. We therefore execute compiled candidates on inputs from the selected group, still without answer labels. Their traces record grounding success, executed primitives, output structure, and evidence cost.

A candidate is valid only if every trace satisfies the declared structural checks and evidence budget. Its registered objective maps each trace Z to a normalized score $J(Z) \in [0, 1]$, based on grounding

consistency, slot coverage and diversity, or event-order completeness and agreement. All registered objectives share this output contract, making their mean scores comparable. We rank valid candidates by their mean score, break ties deterministically, and denote the selected entry by $\ell_t^\star$. The ranking uses only structural traces, without answer predictions or outcome-based signals. A compiler error or failed invariant gives the proposer local feedback for another attempt within the fixed budget.

3.2.4 Program Library Update

After successful structural selection, we store the selected entry in the persistent library:

$$\mathcal{L}_{t+1} = \mathcal{L}_t \cup \{\ell_t^\star\}. \tag{1}$$

The stored specification marks matching inputs as covered, while its program becomes available for reuse. Monitoring then recomputes the uncovered set against $\mathcal{L}_{t+1}$, redirecting the next cycle to the remaining eligible groups. Acquisition ends when no eligible group remains or when a cycle exhausts its attempt budget without a valid candidate. A failed cycle leaves the library unchanged.

3.3 Phase II: Applicability-Guided Program Reuse

Rather than synthesizing a new program for each input, Phase II instantiates a stored entry through routing, grounding, and execution.

3.3.1 Applicability-Guided Routing and Grounding

The router first evaluates each acquired entry's applicability predicate, $\mathrm{Applicable}(\ell, x)$. For every match, $\mathrm{Bind}(\ell, x)$ returns an input-specific grounding b, which specifies where or what the program should observe in video v. An entry survives only if it is applicable and binding succeeds. When several entries survive, the predeclared deployment priority ρ_ℓ selects the entry with the lowest value. If none survives, MORPH-V falls back to Direct. Otherwise, routing returns $P^\star$ and b.

3.3.2 Program Execution and Prediction

When an acquired entry is selected, its program $P^\star$ uses grounding b to construct evidence e from video v. The frozen reasoner f_θ then predicts $\hat{y}$ from question q and evidence e:

$$e = P^\star(v, b), \qquad \hat{y} = f_\theta(q, e). \tag{2}$$

The reasoner's parameters remain fixed, with adaptation confined to evidence construction.

4 Experiments

4.1 Experimental Setup

Datasets. We evaluate MORPH-V on five video question-answering populations spanning three observation requirements. Interval localization uses a predeclared Video-MME-v2 slice ($N = 80$) (Fu et al., 2026) and LongVideoBench ($N = 335$) (Wu et al., 2024). Slot-set coverage uses HERBench Temporal Shot Ordering questions (Ben Ami et al., 2026), with a 60-question cohort included in the 100-question TSO100 aggregate. Event-timeline reasoning uses prospective HERBench Action Sequence Integrity and Identification (ASII; $N = 240$) questions. LongVideoBench also includes 96 slot-set questions, and we report accuracy over its full 1,337-question validation set. Acquisition inputs are disjoint from evaluation and have their answer labels withheld.

Observation Programs. QuoteBind grounds an interval and samples local detail. EvidenceSetBind uses SlotCover to distribute 64 frames across required slots, while OrderContrastBind organizes pairwise temporal relations without changing the input frames. Direct handles unsupported inputs. Program names are assigned after acquisition and are not supplied as choices to the proposer.

Baselines. Our matched-budget controls test localization (Static Parent, Random Child), slot coverage (SingleQuery, Direct/Uniform, Random), and temporal organization (Uniform64, RawAnchor, shuffled relations). Direct64 is an additional localization reference. MDP3 (Sun et al., 2025) provides a 64-frame adaptive-sampling comparison on HERBench and TSO100 with Qwen3-VL.

Table 1: Main-benchmark accuracy (%). Public rows provide context; local comparisons use $N = 80$ for Video-MME-v2 and $N = 431$ for LongVideoBench. Non-Lin is the official non-linear group score. Bold marks the better local result for each reasoner.

(a) Video-MME-v2

Model	Frames	Non-Lin	Acc.
Gemini-3-Pro	1 fps	38.20	56.84
Doubao-Seed-2.0-Pro	1 fps	35.16	53.12
Gemini-3-Flash	1 fps	32.87	52.36
GPT-5	50	26.35	44.69
Qwen3.5-397B-A17B-Think	64	20.70	39.30
Qwen3-VL-235B-A22B-Think	512	19.00	36.80
Qwen3-VL-235B-A22B-Instruct	64	16.50	33.80
InternVL3.5-241B-A28B-Instruct	64	15.80	32.90
Qwen3.6-27B	64	17.81	33.75
MORPH-V (Qwen3.6-27B)	≤ 64	**27.81**	**41.25**
Qwen3-VL-32B	64	13.54	30.00
MORPH-V (Qwen3-VL-32B)	≤ 64	**22.96**	**33.75**

(b) LongVideoBench

Model	Frames	Acc.
GPT-4o (0513)	256	66.7
Aria	256	64.2
Gemini-1.5-Pro (0514)	256	64.0
LLaVA-Video-72B-Qwen2	128	63.9
Gemini-1.5-Flash (0514)	256	61.6
LLaVA-OneVision-Qwen2-72B-OV	32	61.3
LLaVA-Video-7B-Qwen2	128	61.1
InternVL2-40B	16	59.3
GPT-4-Turbo (0409)	256	59.1
Qwen3.6-27B	64	64.97
MORPH-V (Qwen3.6-27B)	≤ 64	**77.03**
Qwen3-VL-32B	64	57.08
MORPH-V (Qwen3-VL-32B)	≤ 64	**70.53**

Metrics. We report question accuracy and gains for reuse, and structurally valid proposals and stored entries for acquisition. Video-MME-v2 also uses its official non-linear group score (Non-Lin). Growth uses an unweighted macro-average over three evidence types. LongVideoBench results distinguish routed and full-population performance; uncertainty is estimated per video.

Implementation Details. Frozen Qwen3.6 proposes programs; frozen Qwen3.6 and Qwen3-VL answer questions. Acquisition starts from $\mathcal{L}_0 = \{\text{Direct}\}$ at temperature 0 with at most three anonymized examples and no complete reference graphs in the library or examples. Checkpoints reload stored graphs through the versioned registry; the separate LongVideoBench deployment uses a fixed mixed-resolution QuoteBind variant.

4.2 Main Results

Performance on Video Benchmarks. Table 1 compares methods on matched questions and reasoners. On Video-MME-v2, Qwen3.6 accuracy increases from 33.75% to 41.25%, and Non-Lin rises from 17.81 to 27.81; Qwen3-VL accuracy improves from 30.00% to 33.75%. On the 431 routed LongVideoBench questions, accuracy rises from 64.97% to 77.03% with Qwen3.6 and from 57.08% to 70.53% with Qwen3-VL. Retaining Direct on the remaining questions gives full-validation gains of 3.89 and 4.34 points over 1,337 questions with unchanged answer models.

Performance Across Evidence Types. The improvements extend to slot-set and event-timeline reasoning (Table 2). On HERBench and TSO100, EvidenceSetBind exceeds the matched controls with

Table 2: Accuracy (%) on additional populations with fixed programs reused across reasoners. Matched control denotes Direct on HERBench and TSO100 and Uniform64 on ASII. The HERBench 60 questions are a subset of TSO100. Gains use unrounded answer counts. Bold marks the best measured result; dashes denote unmeasured comparisons.

Reasoner	Method	HERBench ($N = 60$)	TSO100 ($N = 100$)	ASII ($N = 240$)
Qwen3.6-27B	Matched control	48.33	44.00	42.50
	MORPH-V	**75.00**	**73.00**	**51.25**
Qwen3-VL-32B	Matched control	21.67	19.00	45.83
	MDP3-64	40.00	42.00	–
	MORPH-V	**56.67**	**58.00**	**55.42**

Table 3: Qwen3.6-27B accuracy (%) after reloading successive open-composition states on fixed interval ($N = 335$), slot-set ($N = 60$), and event-timeline ($N = 240$) populations. Macro is the unweighted mean of three populations; Δ is the percentage-point change from Direct-only $\mathcal{L}_0$.

State	Added entry	Accuracy (%) Interval	Slot set	Event timeline	Macro	Δ (pp)
$\mathcal{L}_0$	Direct initialization	63.88	55.00	37.08	51.99	–
$\mathcal{L}_1$	Interval entry	70.15	55.00	37.08	54.08	+2.09
$\mathcal{L}_2$	Slot-set entry	70.15	70.00	37.08	59.08	+7.09
$\mathcal{L}_3$	Event-timeline entry	70.15	70.00	39.17	**59.77**	**+7.78**

both reasoners, gaining 26.67 and 29.00 points with Qwen3.6. With Qwen3-VL, it also outperforms MDP3-64 on both populations. On ASII, OrderContrastBind improves over Uniform64 by 8.75 points with Qwen3.6 and 9.58 points with Qwen3-VL. Program reuse therefore benefits all three evidence types under both reasoners.

4.3 Program Acquisition

Acquisition Results. We assess program acquisition through held-out recovery and open composition. Held-out recovery starts without three reference programs and selects the largest uncovered cluster at each cycle. All three proposals compile, pass structural verification, and enter the library. Open composition instead follows a predeclared interval, slot-set, and event-timeline order. Of its 18 proposals, 10 are structurally valid and three distinct graphs are stored, none exactly matching a predeclared reference. Acquisition is therefore not limited to reference compositions.

Coverage Guides Program Acquisition. We next examine how stored coverage affects subsequent acquisition. Figure 1**(a)** tracks three uninterrupted held-out recovery cycles. Each stored entry removes matching inputs from the unsupported set, redirecting acquisition from event order to required states and then quoted speech. The library consequently grows from Direct alone to three specialized entries plus Direct, without answer calls or human edits between cycles.

Impact of Library Growth. To measure accumulated utility, we reload successive composition checkpoints and evaluate them on the same fixed answer populations. Table 3 shows that macro-averaged accuracy increases after each stored entry, rising from 51.99 at $\mathcal{L}_0$ to 59.77 at $\mathcal{L}_3$. The total gain is 7.78 points (95% CI, $[2.55, 13.20]$). Outcomes are withheld during acquisition and used only for checkpoint evaluation, so these gains measure the utility of structurally selected programs.

4.4 Ablation Study

We examine each evidence operation using the controls in Table 4. Pixel-matched localization controls and fixed-frame slot-set and temporal controls isolate the corresponding observation choices.

Table 4: Mechanism controls. Gain is the selected program minus each control, in percentage points. Blocks state the reasoner and evaluation size.

Setting	Acc. (%)	Gain (pp)
QuoteBind (Qwen3.6; $N = 335$)		
QuoteBind	**77.91**	–
Direct64	64.18	+13.73
Static Parent	67.16	+10.75
Random Child	59.70	+18.21
EvidenceSetBind (Qwen3.6; $N = 60$; 64 frames)		
EvidenceSetBind (SlotCover)	**75.00**	–
SingleQuery	63.33	+11.67
Direct / Uniform	48.33	+26.67
Random	33.33	+41.67
OrderContrastBind (Qwen3-VL; $N = 240$; identical 64 frames)		
OrderContrastBind	**55.42**	–
Uniform64	45.83	+9.58
RawAnchor	47.50	+7.92
Shuffled OrderContrast	32.08	+23.33

Localization Improves Video Reasoning. On LongVideoBench with Qwen3.6, QuoteBind gains 13.73 points over the lower-pixel Direct64 reference. Static Parent retains the parent interval, while Random Child moves local observations away from the bound event. Both match QuoteBind's pixels and transcript context but trail by 10.75 and 18.21 points, respectively, isolating the benefit of event-specific detail.

Slot Coverage Improves Multi-State Reasoning. We next assess multi-slot observation on HER-Bench with Qwen3.6. With 64 frames in every condition, EvidenceSetBind obtains 45/60 correct answers, compared with 38/60 for SingleQuery and 29/60 for Direct/Uniform. SingleQuery retains semantic retrieval but collapses the required states into one query; Direct/Uniform removes targeted allocation. The ordering indicates that covering distinct required states is more effective than relying on a global query or uniform sampling under the same frame budget.

Temporal Relations Improve Sequence Reasoning. Finally, we test temporal organization on ASII with Qwen3-VL, keeping the same 64 chronological frames across conditions. OrderContrastBind exceeds Uniform64 by 9.58 points. Removing pairwise organization (RawAnchor) lowers accuracy by 7.92 points, while shuffling the relations causes a larger drop of 23.33 points. Since these variants differ only in their relation payload, the results support accurate pairwise organization over timestamps alone or an arbitrary ordering.

4.5 Analysis

Cross-Model Transferability. To evaluate transfer across answer models, we keep programs, routes, and questions fixed and change the frozen reasoner. Figure 3 shows gains with both Qwen3.6 and Qwen3-VL. On LongVideoBench, QuoteBind contributes 3.44 points to the Qwen3.6 full-population gain, compared with 0.45 from EvidenceSetBind; both routes also improve Qwen3-VL. On ASII, OrderContrastBind likewise improves over Uniform64 under both reasoners. The stored programs therefore remain useful across these reasoners without re-acquisition.

Error and Reliability Analysis. We further examine when shared programs improve the evidence supplied to the reasoner. On ASII, OrderContrastBind repairs 35 Uniform64 errors with Qwen3.6 and 37 with Qwen3-VL, while introducing 14 errors under either model (Figure 4). In the predefined reliability strata, gains are larger when an independently phrased retrieval agrees with the selected event-order candidate. Both reasoners gain more than 21 points when the top-two relation margin is at least 0.40. These post-outcome diagnostics associate more reliable bound evidence with greater program utility across models.

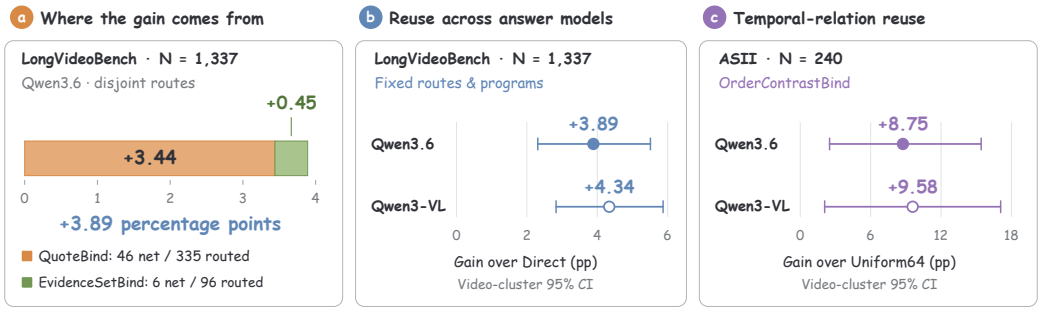

Figure 3: Fixed-library utility across **(a)** LongVideoBench route contributions, **(b)** cross-reasoner reuse, and **(c)** ASII reuse. Bars show gains over Direct or Uniform64 with video-cluster 95% CIs.

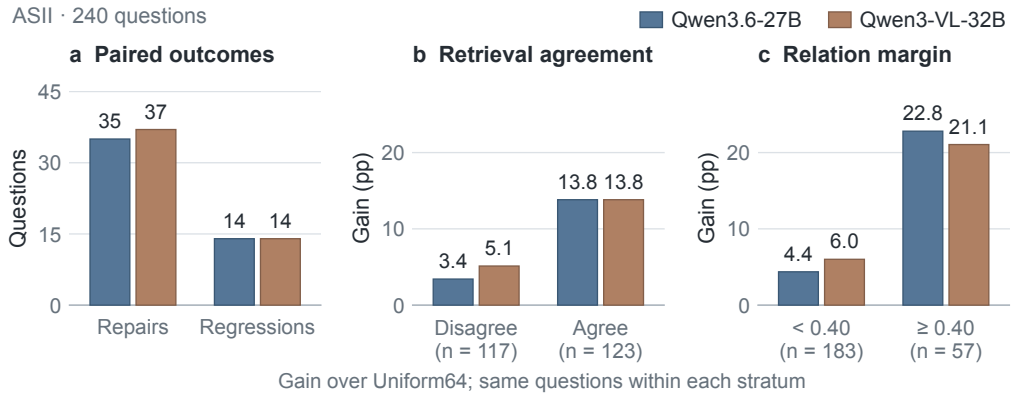

Figure 4: Error correction and evidence reliability on ASII (240 questions). The program and 64-frame inputs are fixed across reasoners. (a) Paired repairs and regressions relative to Uniform64. (b) Agreement of an independently phrased retrieval with the top candidate. (c) Score margin between the top two candidates. Reliability strata are analyzed after outcome evaluation.

Beyond Adaptive Frame Selection. On TSO100, EvidenceSetBind exceeds MDP3-64 by 16.00 points with Qwen3-VL and the same 64-frame budget (Table 2). On the 335-question LongVideoBench quote route, QuoteBind exceeds QCA-Allocated (Peng et al., 2026) by 12.54 points with Qwen3.6, holding transcript context, prompt, and the 20.87M-pixel budget fixed. These gains support explicit grounding and slot coverage beyond adaptive frame selection.

Capability Accumulation. In Table 3, the slot-set entry raises accuracy from 55.00% to 70.00% while interval accuracy stays at 70.15%. Adding the event-timeline entry then raises its accuracy from 37.08% to 39.17%, with both earlier scores unchanged. With questions and reasoner fixed, this run shows utility accumulating across requirements while earlier gains persist.

5 Conclusion

Our work addresses the lack of persistent observation capability in long-video reasoning. To this end, we propose MORPH-V, which couples coverage-guided acquisition with applicability-guided reuse of observation programs. Structural feedback supports program acquisition without answer outcomes, while stored coverage directs subsequent acquisition toward unsupported requirements. Experiments across five populations with two frozen reasoners show gains over matched controls and increasing utility as the library grows. Further analysis identifies complementary benefits from localization, multi-state coverage, and temporal organization, alongside cross-reasoner reuse of fixed programs. MORPH-V enables observation capability to accumulate without parameter updates.

AI USE STATEMENT

Generative AI tools were used to assist with language editing and improving the readability of the manuscript. All AI-assisted content was reviewed and verified by the authors. The authors take full responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

We provide a minimal implementation of the core algorithm and scripts for acquisition replay and library-growth evaluation from frozen records. The package does not rerun model inference or video preprocessing; the appendix details the method and evaluation protocols. Code and instructions are available at `https://anonymous.4open.science/r/morph-v-review-artifact-C3A6/for_review/README.md`.

REFERENCES

Dan Ben Ami, Gabriele Serussi, Kobi Cohen, and Chaim Baskin. Herbench: A benchmark for multi-evidence integration in video question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4505–4514, June 2026.

Shyamal Buch, Arsha Nagrani, Anurag Arnab, and Cordelia Schmid. Flexible frame selection for efficient video reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 29071–29082, June 2025.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, Peixian Chen, Yanwei Li, Shaohui Lin, Sirui Zhao, Ke Li, Tong Xu, Xiawu Zheng, Enhong Chen, Caifeng Shan, Ran He, and Xing Sun. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24108–24118, June 2025.

Chaoyou Fu, Haozhi Yuan, Yuhao Dong, Yi-Fan Zhang, Yunhang Shen, Xiaoxing Hu, Xueying Li, Jinsen Su, Chengwu Long, Xiaoyao Xie, Yongkang Xie, Xiawu Zheng, Xue Yang, Haoyu Cao, Yunsheng Wu, Ziwei Liu, Xing Sun, Caifeng Shan, and Ran He. Video-mme-v2: Towards the next stage in benchmarks for comprehensive video understanding, 2026. URL `https://arxiv.org/abs/2604.05015`.

Gabriel Grand, Lio Wong, Maddy Bowers, Theo X. Olausson, Muxin Liu, Joshua B Tenenbaum, and Jacob Andreas. Lilo: Learning interpretable libraries by compressing and documenting code. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 30399–30446, 2024. URL `https://proceedings.iclr.cc/paper_files/paper/2024/file/819cebb05f993840e8a52d7564c5c282-Paper-Conference.pdf`.

Tanmay Gupta and Aniruddha Kembhavi. Visual programming: Compositional visual reasoning without training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14953–14962, June 2023.

Kai Hu, Feng Gao, Xiaohan Nie, Peng Zhou, Son Tran, Tal Neiman, Lingyun Wang, Mubarak Shah, Raffay Hamid, Bing Yin, and Trishul Chilimbi. M-llm based video frame selection for efficient video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13702–13712, June 2025a.

Shengran Hu, Cong Lu, and Jeff Clune. Automated design of agentic systems. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 21344–21377, 2025b. URL `https://proceedings.iclr.cc/paper_files/paper/2025/file/36b7acf6f6010652b3f2a433774a66fe-Paper-Conference.pdf`.

Jing Huang, Luyuan Chen, Zhijie Xu, Yadong Li, Xingzhong Xu, Siye Chen, Jie Liu, Ming Kong, and Qiang Zhu. Meta: Meta evolution of tool trajectory adaptation for long-video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9837–9846, June 2026.

Guanyu Jiang, Zhaochen Su, Xiaoye Qu, and Yi R Fung. Xskill: Continual learning from experience and skills in multimodal agents. *arXiv preprint arXiv:2603.12056*, 2026.

Jialuo Li, Bin Li, Jiahao Li, and Yan Lu. Divide, then ground: Adapting frame selection to query types for long-form video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11369–11380, June 2026.

Shuming Liu, Chen Zhao, Tianqi Xu, and Bernard Ghanem. Bolt: Boost large vision-language model without training for long-form video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3318–3327, June 2025.

Martin Q. Ma, Yuxiao Qu, Aditya Agrawal, Willis Guo, Paul Pu Liang, Ruslan Salakhutdinov, and Louis-Philippe Morency. Act2see: Emergent active visual perception for video reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5455–5464, June 2026.

Jun Peng, Baiyang Song, Jie Li, Hui Li, Yiyi Zhou, Rongrong Ji, and Yonghong Tian. Qca: Query- and content-aware keyframe selection for long video understanding. In Paolo Favaro, Zuzana Kukelova, Atsuto Maki, Anna Rohrbach, Konrad Schindler, and Federico Tombari (eds.), *Computer Vision – ECCV 2026*, pp. 18–35, Cham, 2026. Springer Nature Switzerland. ISBN 978-3-032-37321-2.

Yaxuan Qin, Hefei Li, Wenqi Mu, and Yancheng He. Efficient frame selection for long video understanding via reinforcement learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16944–16953, June 2026.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine Mcleavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 28492–28518. PMLR, 23–29 Jul 2023. URL `https://proceedings.mlr.press/v202/radford23a.html`.

Akshit Singh, Shyam Marjit, Wei Lin, Paul Gavrikov, Serena Yeung-Levy, Hilde Kuehne, Rogerio Feris, Sivan Doveh, James Glass, and M. Jehanzeb Mirza. Ttrv: Test-time reinforcement learning for vision language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 33153–33163, June 2026.

Hui Sun, Shiyin Lu, Huanyu Wang, Qing-Guo Chen, Zhao Xu, Weihua Luo, Kaifu Zhang, and Ming Li. Mdp3: A training-free approach for list-wise frame selection in video-llms. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 24090–24101, October 2025.

Wenhui Tan, Xiaoyi Yu, Jiaze Li, Yijing Chen, Jianzhong Ju, Zhenbo Luo, Ruihua Song, and Jian Luan. Msjoe: Jointly evolving mllm and sampler for efficient long-form video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19487–19496, June 2026.

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

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023.

Xiaohan Wang, Yuhui Zhang, Orr Zohar, and Serena Yeung-Levy. Videoagent: Long-form video understanding with large language model as agent. In Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol (eds.), *Computer Vision – ECCV 2024*, pp. 58–76, Cham, 2025a. Springer Nature Switzerland. ISBN 978-3-031-72989-8.

Xingyao Wang, Yangyi Chen, Lifan Yuan, Yizhe Zhang, Yunzhu Li, Hao Peng, and Heng Ji. Executable code actions elicit better LLM agents. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 50208–50232. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/wang24h.html.

Ziyang Wang, Shoubin Yu, Elias Stengel-Eskin, Jaehong Yoon, Feng Cheng, Gedas Bertasius, and Mohit Bansal. Videotree: Adaptive tree-based video representation for llm reasoning on long videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3272–3283, June 2025b.

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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Findings*, pp. 9088–9099, June 2026.

Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. Longvideobench: A benchmark for long-context interleaved video-language understanding. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 28828–28857. Curran Associates, Inc., 2024. doi: 10.52202/079017-0907. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/329ad516cf7a6ac306f29882e9c77558-Paper-Datasets_and_Benchmarks_Track.pdf.

Xunjian Yin, Xinyi Wang, Liangming Pan, Li Lin, Xiaojun Wan, and William Yang Wang. Gödel agent: A self-referential agent framework for recursively self-improvement. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 27890–27913, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.1354. URL https://aclanthology.org/2025.acl-long.1354/.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11975–11986, October 2023.

Jenny Zhang, Shengran Hu, Cong Lu, Robert Lange, and Jeff Clune. Darwin gödel machine: Open-ended evolution of self-improving agents. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 104223–104294, 2026a. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/aa5f5e6eb6f613ec412f1d948dfa21a5-Paper-Conference.pdf.

Jiayi Zhang, Jinyu Xiang, Zhaoyang Yu, Fengwei Teng, XiongHui Chen, Jiaqi Chen, Mingchen Zhuge, Xin Cheng, Sirui Hong, Jinlin Wang, Bingnan Zheng, Bang Liu, Yuyu Luo, and Chenglin Wu. Aflow: Automating agentic workflow generation. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 34040–34077, 2025a. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/5492ecbce4439401798dcd2c90be94cd-Paper-Conference.pdf.

Ruixu Zhang, Deyi Ji, Lanyun Zhu, Xuanyi Liu, Yuxin Meng, Ruihang Chu, and Yujiu Yang. Videozero: Self-evolution video understanding, 2026b. URL https://arxiv.org/abs/2605.14733.

Shaojie Zhang, Jiahui Yang, Jianqin Yin, Zhenbo Luo, and Jian Luan. Q-frame: Query-aware frame selection and multi-resolution adaptation for video-llms. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22056–22065, October 2025b.

Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Zhengyang Liang, Shitao Xiao, Minghao Qin, Xi Yang, Yongping Xiong, Bo Zhang, Tiejun Huang, and Zheng Liu. Mlvu: Benchmarking multi-task long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13691–13701, June 2025.

Zirui Zhu, Hailun Xu, Yang Luo, LIU YONG, Kanchan Sarkar, Zhenheng Yang, and Yang You. Focus: Efficient keyframe selection for long video understanding. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 98463–98486, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/a010577f93a8301dada61f0b97cda923-Paper-Conference.pdf.

Yuanhao Zou, Shengji Jin, Andong Deng, Youpeng Zhao, Jun Wang, and Chen Chen. A.i.r.: Enabling adaptive, iterative, and reasoning-based frame selection for video question answering. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 11302–11329, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/1332893b662f655660c9abdf793230cf-Paper-Conference.pdf.

A Program Acquisition and Library Growth

A.1 Autonomous Growth Replay

The acquisition implementation initializes the library state with Direct and monitors three streams of answer-independent observation requirements. Inputs are grouped by evidence type, after which the largest recurrent cluster with at least $m = 10$ inputs and not covered by a stored applicability predicate becomes the next acquisition target. The proposal packet contains anonymized examples, typed signatures already represented by stored entries, and the primitive registry. Source identifiers, task identifiers, and every outcome-bearing field are removed before proposal. The speech-anchor packet additionally excludes answer-choice text. After storage, the updated signature set is supplied to the next cycle.

Given this packet, frozen Qwen3.6 generates a specification in strict JSON using non-thinking mode and temperature 0. The proposer receives at most three anonymized examples, with limits of six attempts, four candidates, and 64 structural executions per cluster. The DSL exposes the applicability predicate, observation-gap diagnosis, binding operator, evidence transformation, invariants, and structural objective separately, without supplying a program name or preassembled target signature. Evidence typing connects compatible roles through an interval, slot set, or event timeline. The expanded registry provides three binders, three evidence transformations, and three structural objectives per type. Its Cartesian product contains 81 type-compatible signatures, including the three held-out evaluation targets, but no complete target entry appears in the initial library state or proposal demonstrations.

The resulting specification enters compilation before structural verification. Applicability must match the cluster, evidence types must agree, dependencies must be acyclic, every primitive must resolve through the executable primitive registry, and the required answer-call, budget, outcome-exclusion, replay, and type-specific invariants must be declared. Passing these checks emits an executable graph. The generic executor runs the graph on answer-independent verification inputs, records its trace, and applies the selected structural objective to candidates satisfying all invariants. A failed check returns the local error to the proposer within the fixed attempt budget. Exhausting that budget records a failed cycle.

Table 5: Proposal-to-entry records for the deterministic uninterrupted replay. All three first proposals compile and pass structural verification. Canonical program names are assigned by signature comparison only after storage.

Cycle	Inputs	Proposed name	Selected program behavior	Canonical name	Attempts
1	48	`ChronoSequence RelationalTrace`	Event timeline; ordered pairwise relations	OrderContrast Bind	1
2	40	`semantic_slot coverage_optimizer`	Slot set; balanced slot coverage	EvidenceSet Bind	1
3	19	`speech-anchored concurrent-state`	Interval; semantic transcript binding	QuoteBind	1

A.2 Expanded Typed DSL Inventory

Serialized Representation. Each program specification is serialized as

$$\sigma = (a, \delta, b, \phi, \mathcal{I}, J), \tag{3}$$

where a is the applicability predicate, δ is the observation-gap diagnosis, b is the binder, ϕ is the evidence transformation, $\mathcal{I}$ is the set of execution invariants, and J is the structural objective. The main-text operators expose this representation as $\mathrm{Applicable}(\ell, x) = a(x)$, $\mathrm{Bind}(\ell, x) = b(x)$, and $\mathrm{Score}(P, Z) = J(Z)$. The condition $\mathrm{Valid}(P, Z) = 1$ holds exactly when Z satisfies $\mathcal{I}$ and stays within its evidence budget.

The typed DSL is $\Omega = (\mathcal{U}, \Gamma, \{\mathcal{B}_u, \Phi_u, \mathcal{J}_u\}_{u \in \mathcal{U}})$, where $\mathcal{U}$ contains the three evidence types, Γ is the versioned primitive registry, and $\mathcal{B}_u$, Φ_u, and $\mathcal{J}_u$ are the compatible binders, transformations, and

Table 6: Expanded DSL inventory. Semicolons separate the three alternatives exposed for each role.

Evidence type	Binder primitives	Evidence transformations	Structural objectives
Interval	lexical-span; semantic-segment; hybrid span-segment	local reallocation; multiscale local context; boundary contrast	anchor capture; binding specificity; specificity then shift equivariance
Slot set	joint-query; independent semantic slots; contrastive state slots	independent top-k; balanced slot cover; diversity-aware slot cover	mean support; minimum-slot coverage; coverage then diversity
Event timeline	single-view retrieval; multiscale retrieval; cross-view audited retrieval	raw timestamp table; signed pairwise contrasts; normalized order matrix	confidence margin; finite completeness; agreement then margin

objectives for type u. Its formal signature space is

$$\mathcal{S}_\Omega = \bigcup_{u \in \mathcal{U}} \{u\} \times \mathcal{B}_u \times \Phi_u \times \mathcal{J}_u. \tag{4}$$

For a specification σ, the composition signature $\mathrm{sig}(\sigma) = (u(\sigma), b, \phi, J)$ selects one evidence type, binder, transformation, and objective. The terms a, δ, and $\mathcal{I}$ instead encode applicability, gap diagnosis, and validity conditions. A proposal is admissible only if its signature belongs to $\mathcal{S}_\Omega$, its graph is well typed, and its applicability predicate covers the triggering cluster.

Table 6 lists the composition axes used by the expanded acquisition study. Every row defines three alternatives for the binder, evidence-transformation, and structural-objective roles. Their Cartesian product yields 27 signatures per evidence type and 81 overall. Type checks and invariants remove invalid graphs before execution. Random intervals, random frames, and shuffled relations remain evaluation controls rather than acquisition primitives.

The held-out interval target illustrates what becomes a library entry. Its specification records applicability and invariants; its typed graph combines semantic-segment binding, multiscale local context, and specificity followed by shift equivariance. The stored entry also retains the graph's primitive revisions, parameters, execution records, and verification result. The slot-set target combines independent semantic slots, balanced slot cover, and the coverage-diversity objective. The event-timeline target combines cross-view audited retrieval, signed pairwise contrasts, and agreement followed by margin. These signatures receive the canonical names QuoteBind, EvidenceSetBind, and OrderContrastBind only after storage.

Once a specification compiles, the same generic executor resolves and runs its `DETECT`, `DIAGNOSE`, `BIND`, `TRANSFORM`, `VERIFY`, `SELECT`, and `PERSIST` nodes. The binder first constructs an interval, slot set, or event timeline from an answer-independent input. The transformation converts that representation into an observation trace, after which the invariant nodes check budget, determinism, outcome exclusion, and type-specific structure. Finally, the selected objective maps the trace to a normalized structural score. No program name or target signature is consulted during this execution path.

The controller aggregates the execution records for each candidate and rejects any graph with a failed invariant or outcome-bearing field. Among the verified candidates, it selects the highest structural score and invokes the graph's `PERSIST` node. Reloading the entry validates the registry version and executes the same graph without reconstructing a hand-coded pipeline. Only after this commit does monitoring resume with the updated library state, removing the newly covered requirement from the next residual set.

After storage, the typed composition signature $\mathrm{sig}(\sigma) = (u(\sigma), b, \phi, J)$ is compared with the three reference signatures to assign a canonical name; structural validity is audited separately by executing the graph on answer-independent inputs. An unmatched entry retains its proposed name, and multiple matches abort the audit. The held-out replay grows from Direct to three requirement-specific entries without answer calls, additional data acquisition, or human edits between cycles. Answer outcomes are attached only after the final library state is fixed.

Table 7: Successive saved library checkpoints on the fixed evidence-type populations. Macro is the unweighted mean of the three type-specific accuracies. Δ and its interval compare each state with $\mathcal{L}_0$ in percentage points.

Checkpoint	Newly retained state	Coverage (%)	Interval	Slot set	Event timeline	Macro	Δ (pp)	95% CI
$\mathcal{L}_0$	Direct initialization	0.00	63.88	55.00	37.08	51.99	–	–
$\mathcal{L}_1$	Interval program	33.23	70.15	55.00	37.08	54.08	+2.09	$[0.43, 3.75]$
$\mathcal{L}_2$	Slot-set program	66.57	70.15	70.00	37.08	59.08	+7.09	$[2.10, 12.33]$
$\mathcal{L}_3$	Event-timeline program	99.90	70.15	70.00	39.17	**59.77**	**+7.78**	$[2.55, 13.20]$

A.3 Composition Acquisition and Library Checkpoints

Unlike held-out recovery, open composition can store a valid graph without matching a reference signature. Its saved states are evaluated on fixed answer populations to test whether the acquired library gains utility as it grows.

A.3.1 Open Composition Acquisition

Search Space and Acquisition Target. Using the 81-signature inventory in Table 6, the acquisition run proposes complete graphs absent from the initial library and proposal examples, though their component primitives remain available. A new composition is identified by its typed executable graph and primitive revisions rather than its generated name. A valid graph compiles, executes on answer-independent inputs, and satisfies its invariants; exact reference agreement is audited separately.

Ten of 18 proposals compile and pass structural verification. Three distinct graphs are stored, none an exact reference match; after the updates, the residual uncovered requirement fraction is 0.0%. The record also contains seven duplicate proposals, and the run takes 1.35 minutes.

A.3.2 Library Checkpoint Evaluation

Intervention and Fixed Population. The checkpoint study isolates program library state as the intervention. The controller saves the Direct-only initialization and each stored update, yielding $\mathcal{L}_0, \ldots, \mathcal{L}_3$. All states use frozen Qwen3.6-27B at temperature 0 with seed 20260916, fixed routing priority, and program-specific budgets. The populations are the 335 LongVideoBench interval questions, 60 HERBench slot-set questions, and an earlier 240-question ASII confirmation cohort, distinct from the fresh four-arm ASII cohort in Table 2. Per-type accuracies retain their original denominators, and their unweighted mean summarizes library-wide progression.

Executing the Acquired State. Checkpoint evaluation reloads each saved graph, validates its registry version, and executes it through the same generic interface used during acquisition. Binding and observation schedules are recomputed for each evaluation input. The interval entry uses its acquired lexical-span binder and 24-frame local-reallocation graph, rather than substituting the separate LongVideoBench deployment with 24 high-resolution local and eight global frames. The checkpoint intervention is therefore the saved library state itself.

Coverage in Table 7 is the unweighted mean of the three fractions routed to a stored non-Direct program. At $\mathcal{L}_3$, these fractions are $334/335$, $60/60$, and $240/240$, giving 99.90%; one interval question falls back to Direct. The 0.0% residual above instead describes uncovered acquisition requirements, not answer-population routing.

Outcome Separation and Statistics. The checkpoint population is fixed by video identity before acquisition and remains disjoint from construction, calibration, and structural-verification data. All predictions are recorded before labels are joined. The primary comparison tests $\mathcal{L}_3$ against $\mathcal{L}_0$. Its interval uses 10,000 paired bootstrap draws, resampling videos within each evidence type before averaging the three accuracies. The three adjacent exact McNemar tests use Holm correction, yielding adjusted p-values of 0.030, 0.098, and 0.458. Thus, the point estimates increase at every checkpoint, but only the first adjacent gain is individually significant at 0.05.

Table 8: Relation checks for the QuoteBind schedule and pixel-matched controls. Grounded Child denotes the speech-bound schedule. All arms share the intended answer-call and pixel budgets. Anchor capture and shift equivariance assess whether local observations follow the bound event.

Program	Route cov.	Anchor capture	Shift equiv.	Calls	Pixels
Grounded Child	100.0%	100.0%	100.0%	1	20.87M
Static Parent	100.0%	2.6%	0.0%	1	20.87M
Random Child	100.0%	7.0%	0.0%	1	20.87M

Table 9: Structural scoring on the 60-question fifth cohort before answer evaluation. Columns follow the lexicographic priority from left to right, and values are cohort means. SlotCover wins the coverage tie through diversity.

Program	Min. cov. ↑	Mean cov. ↑	Diversity ↑	Redundancy ↓
SlotCover	1.0000	1.0000	0.3159	0.0083
SlotTopK	1.0000	1.0000	0.3033	0.0113
SingleQuery	0.9939	0.9983	0.2994	0.0079
Uniform	0.9928	0.9979	0.3434	0.0068
Random	0.9793	0.9939	0.3502	0.0108

Run Records and Scope. The acquisition record retains proposal attempts, compilation or verification failures, structural executions, stored entries, and human-intervention counts. Timings separate proposal, compilation, perception, structural assessment, persistence, and answer inference. The manifest fixes stream ordering, tie-breaking, and random seeds; checkpoint comparisons use fixed populations within this run.

A.4 Structural Verification Objectives

For the QuoteBind schedule checks, route coverage is the fraction of inputs with an accepted binding, and anchor capture is the fraction of the 24 local timestamps inside the padded bound interval. Shift equivariance tests whether shifting the interval shifts the local schedule equally. We test positive and negative shifts of up to two seconds, reduced near video boundaries, and accept a maximum timestamp error of 10^{-6} seconds after undoing the shift. Scores average the two directions and then the routed questions.

For the slot-set evidence type, four HERBench temporal-ordering development cohorts define the EvidenceSetBind structural objective. Its lexicographic order gives priority to the least-covered slot, followed by mean coverage, diversity, and lower redundancy. Once defined, the objective is fixed and applied to 60 questions from 30 previously unused videos in a fifth cohort. Across all candidates, the protocol specifies 64 frames and one answer call, and structural selection precedes answer evaluation.

For each slot, SigLIP similarities on the one-frame-per-second lattice are normalized as $r = \mathrm{clip}((s - q_{.05})/\max(q_{.95} - q_{.05}, 10^{-6}), 0, 1)$, using that slot's 5th and 95th percentiles. Slot coverage is the mean of its three highest normalized scores among the selected frames; minimum and mean coverage aggregate across slots. Diversity is one minus the mean pairwise cosine similarity of normalized selected-frame embeddings, clipped to $[0, 1]$. Redundancy is the fraction of frame pairs with cosine similarity above 0.95. Table 9 averages these quantities over questions.

With the structural objective fixed, the selection sequence keeps structural assessment and answer evaluation separate. The holdout population, candidate schedules, and inference configuration are fixed first. The objective then compares the five structural traces and selects SlotCover. Only afterward are the selected schedule and executed controls evaluated for answer accuracy under the same configuration.

The 100-question Temporal Shot Ordering (TSO100) aggregate examines the two leading structural candidates more closely. Its initial Qwen3-VL evaluation executes SlotCover, SingleQuery, Uniform, and Random. SlotTopK's precomputed schedule is evaluated later with the objective and candidate programs unchanged. SlotCover and SlotTopK obtain 58 and 55 correct answers. Their

4/1 W2R/R2W difference has a video-cluster interval of [−1,7] points, leaving the answer-level comparison statistically unresolved.

A.5 Observation-Program Implementations

QuoteBind Binding Implementations. Interval construction compares semantic and lexical question-to-transcript matching. The held-out recovery entry uses a frozen `all-MiniLM-L6-v2` sentence encoder to embed the whole question and nonempty Whisper transcript segments. It selects the highest-cosine-similarity eligible segment, with eligibility calibrated from mismatched-video scores. This selection uses the structural score without answer outcomes, and the replay registers the resulting semantic implementation.

For the downstream LongVideoBench study, the implementation instead matches an explicit quotation to timestamped text. Through the same interface, it uses benchmark subtitles when available and Whisper transcripts otherwise (Radford et al., 2023). After normalizing case, punctuation, and spacing, it accepts the most similar span when fuzzy similarity is at least 0.72. Given a bound span $[t_s, t_e]$, local observation uses the temporal window

$$W(v, q) = [\max(0, t_s - 4), \min(\mathrm{dur}(v), t_e + 4)]. \tag{5}$$

The resulting schedule places 24 local 896×896 frames inside $W(v, q)$ and retains eight global 448×448 frames. Thus, both binders share the same observation pattern but differ in how they produce the interval. The text-span binder produces the LongVideoBench results. In either case, structural checks assess interval capture, time-shift equivariance, binding specificity, and observation budget.

EvidenceSetBind. EvidenceSetBind represents required visual states as slots $\mathcal{S}(q) = \{s_1, \ldots, s_K\}$, with $2 \le K \le 12$. Its selected concrete implementation, SlotCover, assigns temporally separated support to those slots. A frozen SigLIP-SO400M encoder (Zhai et al., 2023) scores each slot against a one-frame-per-second candidate lattice. SlotCover then resolves collisions through semantic support and visual novelty and fills the remaining local budget. It produces 48 local and 16 uniform global 448×448 frames. The resulting traces are ranked by the coverage, diversity, and redundancy criteria in Table 9.

OrderContrastBind. OrderContrastBind retrieves one timestamp $t(a_i)$ for each event in a five-event candidate $c = (a_1, \ldots, a_5)$ using the frozen visual-text encoder. It summarizes those times with pairwise order violations, the minimum adjacent gap, and the temporal-consistency score

$$s_{\mathrm{temp}}(c) = \frac{1}{10} \sum_{i<j} \tanh\left(\frac{t(a_j) - t(a_i)}{40\,\mathrm{s}} \right). \tag{6}$$

The five summaries accompany the same 64 chronological 448×448 frames as Uniform64. An independently phrased retrieval audit checks the highest-scoring candidate. An option recommendation is included when the audit agrees and the top-two score margin is at least 0.40. Otherwise, the evidence supplies relation scores without a recommendation. Retrieval, thresholds, and formatting are fixed before the prospective cohort.

B Additional Evaluation Evidence

The following studies keep development, fixed-program evaluation, and transfer populations separate rather than pooling their results.

HERBench Population Construction. TSO100 and ASII are task-specific populations from HERBench Full-minus-Lite68, retaining the original questions, options, and answer labels. TSO100 combines 40 Temporal Shot Ordering questions from 20 videos with the later 60-question cohort from 30 different videos, using two questions per video. Consequently, the HERBench and TSO100 columns in Table 2 overlap and are not independent replications. The fresh ASII four-arm cohort contains 12 questions per video across 20 videos, each with five candidate five-event sequences. Seeded hash selection excludes previously contacted videos and does not access outcomes; labels are joined after predictions are frozen.

Table 10: QuoteBind development on all 19 strict-quote Video-MME-v2 questions. Static Parent and Random Child match the mixed-resolution observation budget. Paired transitions compare QuoteBind with each control.

Program	Correct	Accuracy	W2R/R2W/net	Calls	Visual budget
MORPH-V/ QuoteBind	**14/19**	**73.7%**	–	1/question	$8\times448 + 24\times896$
Static Parent	9/19	47.4%	5/0/+5	1/question	matched
Random Child	5/19	26.3%	9/0/+9	1/question	matched
Direct64	7/19	36.8%	10/3/+7	1/question	64×448

Table 11: QuoteBind effects across six major LongVideoBench categories under the same speech-binding rule. The table reports Direct accuracy and QuoteBind's gain on each category's routed questions.

Category	Questions	Direct (%)	Δ (pp)
TAA	57	61.4	**+14.0**
T2A	56	66.1	**+25.0**
T3E	55	58.2	**+16.4**
T3O	54	59.3	**+9.3**
T2O	52	69.2	**+9.6**
T2E	43	72.1	**+9.3**

B.1 QuoteBind Development and Breadth

Within the strict-quote development population, the grounded program exceeds Static Parent by 26.3 points with no paired regression and Random Child by 47.4 points. A separate 80-question Video-MME-v2 slice then evaluates routing, using QuoteBind when applicability and binding succeed and Direct as fallback. Under this router, accuracy rises from 27/80 to 33/80, a 7.5-point intent-to-treat gain, and the official group rating rises from 17.8125 to 27.8125. Both development measurements precede the prospective LongVideoBench study.

The 80-question slice consists of the complete four-question groups for video IDs 281–300 in archive 015, fixed by archive and video identity without consulting outcomes. It is a development slice, not a full-benchmark estimate. Acc. is the fraction of correctly answered questions. Non-Lin averages the official non-linear ratings across the 20 groups (Fu et al., 2026). For a relevance group with k correct answers, its rating is $100(k/4)^2$; logic groups follow the released dependency-aware scoring rules. Thus, Non-Lin cannot be reconstructed from aggregate question accuracy alone.

B.1.1 Category Breadth

The six reported categories share the timestamp-bindable quoted-speech requirement while covering event, object, attribute, change, and before/after questions. Across their 317 questions, QuoteBind reaches 78.23% versus 64.04% for Direct. The resulting 59 repairs and 14 regressions yield a video-cluster 95% interval of [8.79,19.38] points. Thus, the category breakdown shows how one applicability condition spans different question content.

B.2 OrderContrastBind Calibration and Evaluation

B.2.1 Initial Program and Reliability Calibration

The initial program combines event timestamps, ten pairwise order features, and a margin-gated recommendation with the same 64 chronological frames as Direct. On a video-disjoint cohort of 20 videos, it obtains 112/240 (46.67%), versus 103/240 (42.92%) for RawAnchor, 80/240 (33.33%) for Shuffled OrderContrast, and 101/240 (42.08%) for Uniform64. Its +4.58-point primary effect over Uniform64 is inconclusive ($p = 0.207$; video-cluster 95% CI $[-3.75,13.33]$), whereas the +13.33-point difference from shuffled relations is larger ($p = 8.46\times10^{-4}$). This contrast motivates calibrating how retrieved order information becomes an option recommendation.

Table 12: Development and evaluation chronology for OrderContrastBind. The initial four-arm study and recommendation-free variant inform calibration. The later pilot and fresh four-arm cohort evaluate the fixed, reliability-calibrated program. Outcome columns report correct answers.

Stage	Videos	Questions	Program	Uniform	Role in evaluation
Initial four-arm evaluation	20	240	112	101	primary comparison inconclusive
Recommendation-free variant	20	240	115	101	development evidence
Reliability-calibrated pilot	10	120	**65**	48	prospective two-arm pilot
Fresh four-arm evaluation	20	240	**133**	110	prospective mechanism study

The subsequent reliability calibration uses the same cohort. Removing the hard recommendation while retaining timestamps and order scores raises accuracy to 115/240. Based on this comparison, the calibrated rule retains the original relation evidence when an independently phrased retrieval audit agrees with the top candidate and the score margin is below 0.25 or at least 0.40. Otherwise, it supplies a recommendation-free score table. Because the original evidence already withholds a recommendation below 0.25, an explicit recommendation requires agreement and a margin of at least 0.40. This rule and its thresholds are then fixed before the later cohorts.

B.2.2 Prospective Cohorts

With the rule fixed, the calibrated program is first evaluated in a two-arm pilot of 120 questions from 10 videos. Four videos contributed structural traces during development, and six are new. None belongs to the earlier 20-video outcome cohort or a prediction bank. Before inference, the protocol fixes the population, schedules, evidence-formatting rules, Qwen3-VL configuration, 64-frame budget, paired test, cluster bootstrap, and success criteria. Only after both prediction sets are recorded does a separate evaluator read the labels.

Under this pilot protocol, the calibrated program obtains 65/120 (54.17%) versus 48/120 (40.00%) for Uniform64. Its 26 repairs and nine regressions give exact McNemar $p = 0.00599$ and a video-cluster 95% interval of [1.67,25.00] points. The pilot therefore measures the overall effect against frames alone. The next study adds raw-retrieval and shuffled-relation controls to examine its source.

The subsequent four-arm study uses 240 questions from 20 additional videos, disjoint from the 148 videos in earlier outcome or structural stages. Its population, four programs, identical 64-frame schedules, Qwen3-VL configuration, primary comparisons, and success criteria are fixed before inference. As in the pilot, all four prediction sets are recorded before outcome evaluation.

Within this four-arm study, OrderContrastBind leads at 133/240 (55.42%), followed by RawAnchor at 114/240 (47.50%), Uniform64 at 110/240 (45.83%), and Shuffled OrderContrast at 77/240 (32.08%). Against Uniform64, its 37/14 W2R/R2W transitions yield +9.58 points (video-cluster interval [2.08,17.08], exact McNemar $p = 0.00177$). The effect is +7.92 points against RawAnchor (35/16, [1.25,14.58], $p = 0.01097$) and +23.33 points against Shuffled OrderContrast (70/14, [15.00,32.50], $p = 4.07 \times 10^{-10}$).

B.2.3 Robustness and Cross-Model Reuse

Video-Level Robustness and Multiplicity. Video-level analyses test whether a small number of videos drives the result. The exact sign-flip test for OrderContrastBind versus Uniform64 gives $p = 0.0334$, and leaving out any video retains an effect of +7.89 to +11.40 points. Sign-flip values are $p = 0.0513$ against RawAnchor and $p = 6.10 \times 10^{-5}$ against Shuffled OrderContrast, with positive leave-one-video-out effects in both comparisons. Counting the final evaluation as a sixth confirmatory attempt yields a Bonferroni threshold of 0.00833. The pre-specified method-versus-Uniform exact $p = 0.00177$ remains below it.

After the Qwen3-VL evaluation, the model-transfer study reuses the same 240 schedules while changing the reasoner to Qwen3.6 and retaining the program, route, and prompt. Under this transfer, OrderContrastBind obtains 123/240 (51.25%) versus 102/240 (42.50%) for Uniform64. The 35/14 W2R/R2W transitions yield +8.75 points (video-cluster interval [2.50,15.42], exact McNemar $p = 0.00380$, video-level sign-flip $p = 0.0253$). Leave-one-video-out effects remain positive at +7.02 to +10.09 points.

Table 13: Cross-model reuse with fixed programs and route populations. Entries are correct predictions. Each block names its own controls. Unavailable control runs are shown as dashes.

Speech localization: LongVideoBench, 335 questions				
Reasoner	**QuoteBind**	**Direct64**	**Static Parent**	**Random Child**
Qwen3.6	**261**	215	225	200
Qwen3-VL	**246**	196	215	184
Multi-state coverage: TSO100, 100 questions				
Reasoner	**EvidenceSetBind (SlotCover)**	**Direct / Uniform**	**SingleQuery**	**Random**
Qwen3.6	**73**	44	63	35
Qwen3-VL	**58**	19	47	14
Temporal organization: ASII, 240 questions				
Reasoner	**OrderContrastBind**	**Uniform64**	**RawAnchor**	**Shuffled**
Qwen3.6	**123**	102	–	–
Qwen3-VL	**133**	110	114	77

Table 14: EvidenceSetBind reuse on LongVideoBench sequence sorting. Correct counts and gains over Direct are separated by reasoner and evaluation stage. The 96-question aggregate combines the initial 40 and later 56 questions and is descriptive.

Reasoner	**Population**	**EvidenceSetBind (SlotCover)**	**SingleQuery**	**Direct**	**Random**	**vs. Direct**
Qwen3.6	Initial 40	**33/40**	32/40	30/40	31/40	+7.50 pp
Qwen3.6	Completion 56	**38/56**	35/56	35/56	34/56	+5.36 pp
Qwen3.6	All 96	**71/96**	67/96	65/96	65/96	+6.25 pp
Qwen3-VL	Initial 40	**26/40**	23/40	21/40	24/40	+12.50 pp
Qwen3-VL	Completion 56	**32/56**	25/56	29/56	24/56	+5.36 pp
Qwen3-VL	All 96	**58/96**	48/96	50/96	48/96	+8.33 pp

Cross-Model Diagnostic. To interpret this transfer, we examine the predefined reliability strata. At top-two relation margins of at least 0.40 ($n = 57$), gains over Uniform64 are 22.81 points with Qwen3.6 and 21.05 points with Qwen3-VL. When the independently phrased retrieval audit agrees with the top candidate ($n = 123$), both gains are 13.82 points, versus 3.42 and 5.13 points on disagreement cases ($n = 117$). Across models, thirteen questions are repaired by both, and nine of 20 videos have positive paired nets under both. These post-outcome diagnostics associate more reliable retrieval with greater program utility.

B.3 Cross-Population and Cross-Model Reuse

Beyond reasoner reuse, cross-dataset reuse applies the same compiler and structural objective to two LongVideoBench sequence-sorting partitions, selecting EvidenceSetBind's SlotCover implementation in both. Across the 96 questions, Qwen3.6 yields 12/6 W2R/R2W with an interval of [−2.17,15.46] points, and Qwen3-VL yields 14/6 with an interval of [0.96,16.16]. Because the 56-question completion follows the initial 40-question evaluation, the combined result is descriptive. Table 14 therefore reports both stages separately.

A separate router-level analysis uses a development-linked HERBench population of 200 questions. Here, applicability sends 40 temporal-ordering questions to EvidenceSetBind and leaves the remaining 160 with Direct. As a result, the combined router reaches 52.0% versus 46.5% for Direct, a +5.5-point gain with 13/2 W2R/R2W and a video-cluster interval of [2.06,9.28].

Table 15: Adaptive sampling on the LongVideoBench quote route. Table differences and W2R/R2W are relative to Direct on the same 335 questions. The text below reports the separate paired effects between QuoteBind and QCA, together with their intervals.

Method	Correct / 335	Acc. (%)	Δ vs. Direct	W2R / R2W
Direct	215	64.18	–	–
QCA-Allocated	219	65.37	+1.19	31 / 27
QCA-Native	220	65.67	+1.49	25 / 20
QuoteBind	**261**	**77.91**	**+13.73**	**62 / 16**

Table 16: Inference cost per routed question. Latency measures the answer request, not end-to-end runtime. The text reports perception preprocessing separately. Acquisition cost is reported with the expanded DSL study.

Cost	QuoteBind	Direct64	EvidenceSetBind	Matched Direct
Answer / verifier calls	1/0	1/0	1/0	1/0
Frames	32	64	64	64
Pixels (M)	20.87	12.85	12.85	12.85
Prompt tokens	22,236	12,848	13,113	13,113
Latency (s)	5.73	3.68	4.12	4.01

B.4 Matched Adaptive-Sampling Comparisons

MDP3. The HERBench/TSO100 comparison uses MDP3 at commit `45616806d117`, selecting 64 448×448 frames from SlotCover's frozen lattice for Qwen3-VL. The adapter preserves the selector objective and bounds the final segment.

With this matched protocol, EvidenceSetBind (SlotCover), MDP3-64, and Direct obtain 58, 42, and 19 correct answers on TSO100. The SlotCover implementation exceeds MDP3 by 16 points (18/2 W2R/R2W; cluster 95% CI [7,25]; exact $p = 4.02\times10^{-4}$), and MDP3 exceeds Direct by 23 points (28/5 W2R/R2W; interval [12,35]). The comparison therefore supports the additional value of the selected allocation under a matched observation budget.

QCA on the Quote Route. Query- and Content-Aware Keyframe Selection (QCA) (Peng et al., 2026) provides an adaptive frame-selection comparison on the 335-question quote route with Qwen3.6-27B. We use released keyframes from commit `842fae5c4654`, without target refitting. QCA-Native supplies all released indices, yielding 57–64 chronological 448×448 frames per question. QCA-Allocated selects 24 evenly spaced positions from the sorted QCA indices at 896×896 and retains eight global 448×448 frames. It shares QuoteBind's bound transcript context, prompt, and 20.87M-pixel budget; only the 24 selected frame indices change. Added after the primary evaluation, these comparisons provide supporting evidence for localization.

Under both QCA configurations, QuoteBind improves accuracy: +12.54 points against QCA-Allocated (55 W2R, 13 R2W; 95% CI [7.71,17.44]; $p = 2.75\times10^{-7}$) and +12.24 points against QCA-Native (57 W2R, 16 R2W; [7.41,17.06]; $p = 1.53\times10^{-6}$). The consistent advantage across matched-budget and native-budget comparisons supports the utility of explicitly binding the speech reference.

C Cost, Evaluation Scope, and Failure Modes

C.1 Inference Cost

For EvidenceSetBind, the Qwen3-VL comparison holds answer inputs to 64 448×448 frames, 12.85M pixels, a mean of 12,968 prompt tokens, and one call across all five candidates and MDP3. Under this matched input, mean answer-request latencies are 5.08, 5.05, 4.63, 3.76, 4.67, and 5.07 seconds for SlotCover, SlotTopK, SingleQuery, Uniform, Random, and MDP3, respectively. These timings measure the request after selection and therefore exclude selector preprocessing.

For OrderContrastBind, the answer-call count and visual schedule match its controls, but relation construction adds preprocessing. This additional work consists of SigLIP indexing on a one-frame-per-second lattice, event retrieval, and relation construction. Encoding 23,612 cached JPEGs from 20 videos on one RTX 4090 takes 304.3 seconds (77.6 frames/s), or 15.22 seconds per video. With 12 questions per video, indexing amortizes to 1.268 seconds per question. Retrieval and relation construction add 0.116 seconds, for 1.384 seconds combined. This total excludes video decoding and the one-time 8.30-second model load. Separately, OrderContrastBind and Uniform64 have mean answer-request latencies of 3.81 and 3.57 seconds with Qwen3.6, and 4.31 and 1.79 seconds with Qwen3-VL.

C.2 Evaluation Scope and Failure Modes

Unless otherwise noted, uncertainty uses 10,000-draw video-cluster intervals and exact McNemar, sign-flip, or leave-one-video-out tests where applicable.

Reasoner Protocol. All central arms use temperature 0, non-thinking mode, deterministic multiple-choice parsing, and one answer call. The LongVideoBench Qwen3.6 evaluations use seed 20260810; the library-checkpoint evaluation uses 20260916. Qwen3-VL uses `Qwen/Qwen3-VL-32B-Instruct`, revision `0cfaf481...3ccb`, in BF16 with a 262,144-token context. The earlier 32,768-token deployment completed 80 four-arm questions before the next question exceeded the context limit in its three local-program requests. After increasing the context window, a protocol and semantic-message audit retained the 320 successful pilot receipts and excluded failed requests. Execution resumed over the remaining questions to produce 335 valid predictions per arm. The failed pilot and continuation records are retained.

Mechanism Interpretation. At the question level, the LongVideoBench controls show the benefit and cost of targeted detail. QuoteBind, Static Parent, and Random Child share the bound transcript context, eight global frames, and 24 high-resolution frames (20.87M decoded pixels). Direct64 receives the question and choices with 64 uniform 448×448 frames (12.85M pixels), without the bound transcript context. Its comparison therefore measures the combined intervention; the two pixel- and transcript-matched controls isolate local frame allocation. Relative to Direct64, QuoteBind repairs 62 questions and regresses on 16. Against Static Parent, the corresponding counts are 46 and 10. The paired video-cluster 95% intervals for QuoteBind's gains over Direct64, Static Parent, and Random Child are $[8.31, 18.91]$, $[6.25, 15.34]$, and $[13.35, 23.18]$ percentage points, respectively.

Duration and Slot-Count Coverage. At the population level, duration analyses partition TSO100 into 14, 74, and 12 questions at < 120 s, 120–180 s, and ≥ 180 s. SlotCover's gains over Direct are $+14.29$, $+28.38$, and $+50.00$ points with Qwen3.6, and $+7.14$, $+43.24$, and $+50.00$ points with Qwen3-VL. Because the population varies in duration but fixes the evidence-slot count at four, these results characterize duration groups rather than slot-count sensitivity.

Failure Modes. At the program-family level, failure modes follow the evidence type. QuoteBind can bind a repeated phrase to the wrong occurrence or miss an event outside its four-second context. EvidenceSetBind can form unevenly specific slots, and semantic similarity may miss the decisive visual state. Across five HERBench development tasks, coverage gains are consistent for temporal ordering but not for false-action or scene-arrangement questions. OrderContrastBind, in turn, inherits timestamp errors for repeated or visually similar actions. When these specialized entries do not apply, Direct handles unsupported inputs, while the reasoner fuses the evidence returned by every route.

Growth Scope. The growth evidence comes from one deterministic recovery replay and one open-composition run. It shows state-changing persistence and checkpoint utility within those runs, but does not isolate the effects of individual acquisition components or estimate variation across independent acquisitions.

Evaluation Populations. The LongVideoBench routed quote population covers 205 videos; HERBench uses 30 unseen videos; and ASII uses 20 prospective videos. The population definitions in Section 4.1 and Appendices B.1–B.3 distinguish prospective, development-linked, and transfer

evidence. Model-transfer comparisons reuse questions to isolate the reasoner change, whereas benchmark-wide estimates combine disjoint routes within their own benchmark denominators.

D QUALITATIVE REPAIRS AND REGRESSIONS

Figures 5 and 6 show how localization and coverage change the observed evidence in repairs and regressions. Cases are chosen by a fixed manifest order before visual inspection. For each program, the repair is the first case with a correct selected child and three incorrect controls. The regression is the first case with a correct Direct or Uniform prediction and an incorrect child. The same selection rule therefore exposes both outcomes. OrderContrastBind's relation-evidence intervention is examined through the RawAnchor and Shuffled OrderContrast controls.

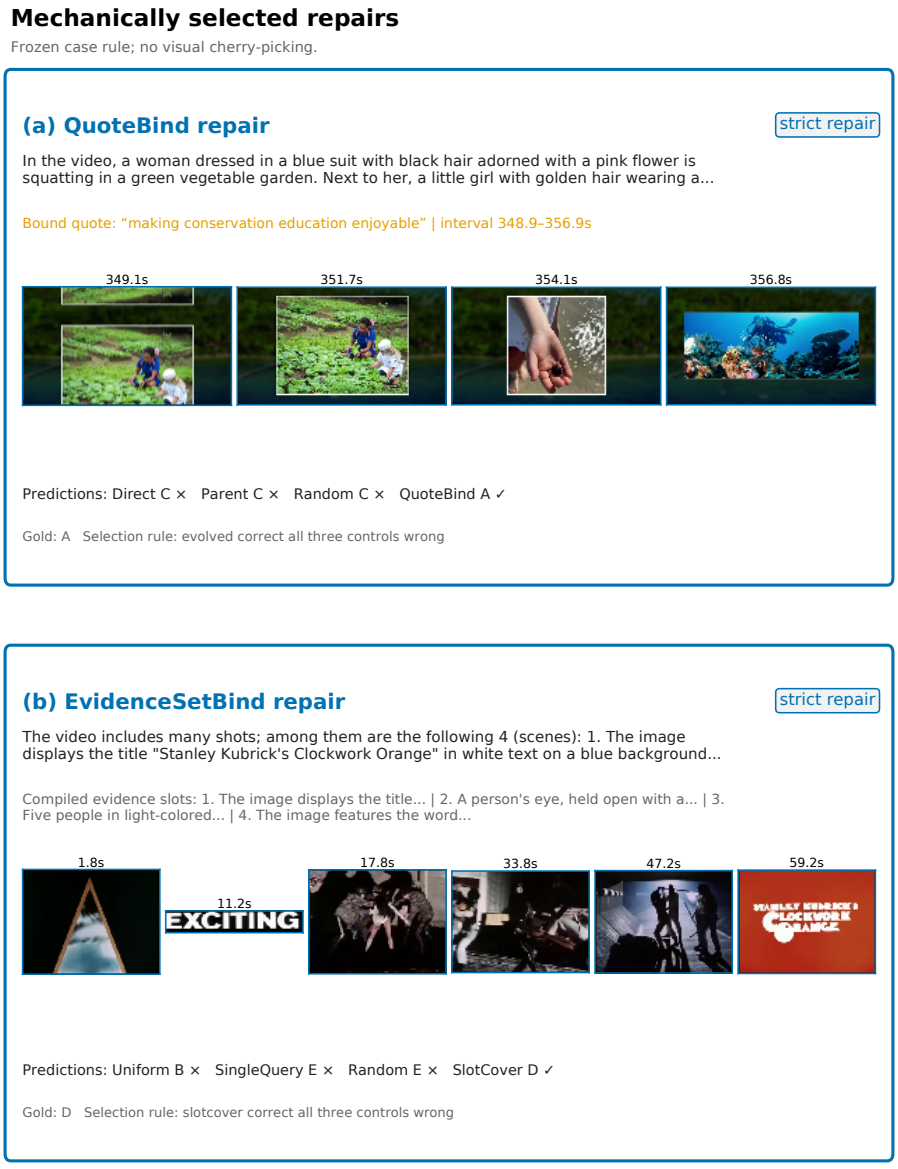

Figure 5: Repairs selected by fixed manifest order. QuoteBind directs high-resolution observations to a timestamp-bound utterance, while EvidenceSetBind distributes observations across required slots. The selected program is correct and every control is wrong in both cases.

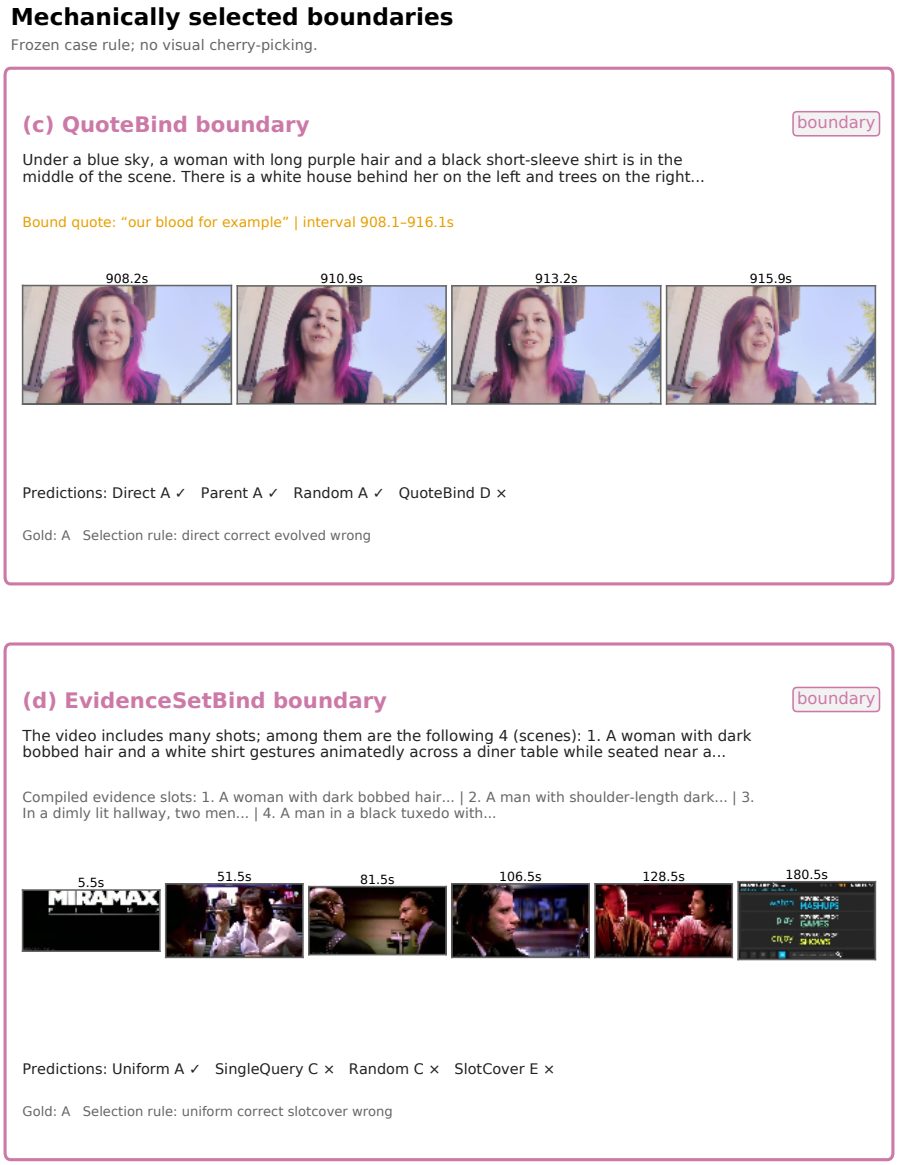

Figure 6: Regressions selected by the same manifest-order rule. Direct or Uniform is correct while the selected program is wrong, illustrating how targeted localization and slot-based allocation can omit or misplace useful evidence.