How Video MLLMs Form Temporal Boundaries: Boundary Flow Tracing and Temporal Revisit

Anonymous authors
Paper under double-blind review

Abstract

Generative video temporal grounding enables video multimodal large language models (MLLMs) to directly produce temporal boundaries, yet their formation remains unclear. We investigate start and end timestamp formation through route interventions and representation probing. Boundary Flow Tracing reveals endpoint- and depth-dependent pathways in TimeLens, with visual-to-prompt effects concentrated at earlier depths than prompt-to-boundary effects. Qwen3-VL and InternVL2.5 also exhibit prompt-to-boundary dependence, with model-specific depth profiles. Restricted readouts recover temporal cues from intermediate visual states, while MiMo-VL probes provide additional evidence of boundary information before generation. These observations motivate Temporal Revisit, a learned intervention that returns intermediate visual evidence to later visual states of a frozen backbone. Broad Temporal Revisit (BTR) tests whether this evidence can improve native boundary generation. Hypothesis-Guided Temporal Revisit (HTR) further examines whether prediction-conditioned reuse with preservation training can improve grounding while limiting prediction changes. BTR improves mIoU on TimeLens-SFT, Qwen3-VL, and MiMo-VL across the three evaluated benchmarks. On TimeLens-SFT, the gains are 3.92, 2.85, and 2.19 percentage points on QVHighlights-TimeLens, Charades-TimeLens, and ActivityNet-TimeLens, respectively. After GRPO, HTR improves native mIoU while modifying fewer predictions than BTR. Post-intervention tracing on TimeLens-SFT shows stronger direct visual-to-boundary dependence after BTR. These results support learned reuse of recoverable temporal evidence, with benefits that depend on the host and intervention design.

1 Introduction

Given an untrimmed video and a natural-language query, video temporal grounding (VTG) aims to identify the temporal boundaries of the queried event. With the development of video multimodal large language models (MLLMs), VTG has increasingly shifted toward a generative formulation, where models directly produce start and end timestamps through temporal representations, grounding-oriented training, and post-training strategies (Ren et al., 2024; Huang et al., 2024b; Chen et al., 2024; Wang et al., 2025; Huang et al., 2024a; Guo et al., 2025; Zeng et al., 2025; Li et al., 2025; Zhang et al., 2026).

Despite these advances, generated timestamps only reveal the final outcome of a complex internal computation process. To produce accurate temporal boundaries, a video MLLM must transform visual observations into temporally localized evidence, integrate this evidence with the query, and finally decode boundary tokens. However, how such temporal boundary decisions are formed inside video MLLMs remains largely unexplored. Existing works primarily focus on improving external model behavior through architecture design, data scaling, or post-training objectives, while the internal states and information pathways supporting native timestamp generation remain unclear.

Mechanistic studies of multimodal models provide a perspective for analyzing these internal processes. Recent work has revealed structured information storage, cross-modal transfer, and layer-wise processing in multimodal models (Basu et al., 2024; Zhang et al., 2025; Yu & Ananiadou, 2024; Yu & Lee, 2025), while VideoLLM analysis has traced cross-frame interaction and video–language information flow (Kim et al., 2026). However, these studies do not explain how generative VTG

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.

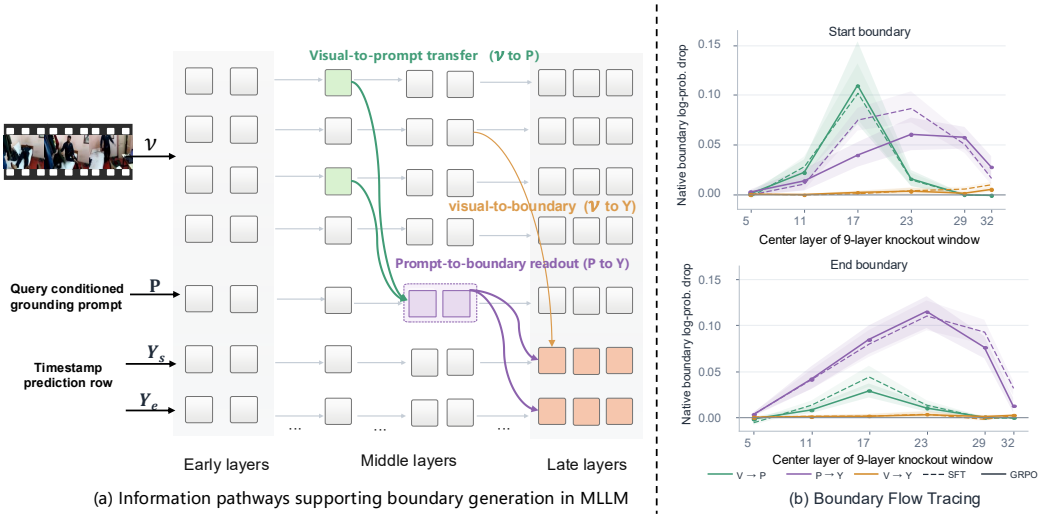

Figure 1: **Boundary Flow Tracing for temporal boundary formation.** **(a)** Candidate information pathways linking visual states $\mathcal{V}$, post-visual prompt states $\mathcal{P}$, and start/end prediction positions $\mathcal{Y}_s$ and $\mathcal{Y}_e$. **(b)** Sensitivity of native timestamp generation to blocking visual-to-prompt, prompt-to-boundary, and direct visual-to-boundary attention across decoder depth. Dashed curves denote SFT and solid curves denote post-GRPO.

models form precise start and end boundaries. In particular, information recoverable from a hidden state does not necessarily correspond to information used by the native timestamp generator. This distinction motivates the following questions: **(1)** Which information pathways support start and end timestamp generation? **(2)** Where and in what form is temporal evidence preserved before boundary generation? **(3)** How does revisiting this evidence change boundary generation while retaining the native decoder? Further comparisons with grounding, mechanistic analysis, and temporal refinement methods are provided in Appendix A.

To answer these questions, we conduct a mechanism-guided study of video MLLMs. We introduce Boundary Flow Tracing to measure how information pathways support native start and end timestamp generation. We analyze intermediate representations and post-training changes using matched TimeLens checkpoints. Additional Qwen3-VL and InternVL2.5 route analyses, together with MiMo-VL probes, examine the scope of these observations (Appendix B.1). We then introduce Temporal Revisit to test whether recoverable intermediate evidence can be reused by the native generator. We further examine whether prediction-conditioned reuse with preservation training can improve grounding while limiting changes to existing predictions.

Our key findings are summarized as follows:

- **We identify the pathways that support start and end timestamps.** Boundary Flow Tracing shows that prompt-to-boundary attention is the strongest tested route in TimeLens, with different depth profiles for the two endpoints and earlier visual-to-prompt support. Prompt-to-boundary dependence also appears in raw Qwen3-VL and InternVL2.5, while GRPO selectively changes the TimeLens profiles (Sections 2.1 and 2.3).
- **We locate boundary cues before timestamp generation.** Linear probes recover boundary information from prompt states, while query-conditioned probes identify localized cues in intermediate visual states. These visual cues do not steadily strengthen toward the final layer. Together, probing and Flow Tracing distinguish where information remains available from the routes supporting native generation (Section 2.2).
- **We use these findings to improve native grounding.** Temporal Revisit returns intermediate visual evidence to later visual states without updating the host or replacing its timestamp generator. BTR improves grounding on TimeLens-SFT, raw Qwen3-VL, and MiMo, and increases direct visual-to-boundary support. HTR uses the native interval to make smaller, more selective revisions on the GRPO host (Sections 3 and 4.2).

2 How Video MLLMs Form Temporal Boundaries

To generate a temporal interval, a video MLLM must transform visual observations into boundary decisions expressed as start and end timestamps. However, the internal process connecting visual evidence to these generated boundaries remains unclear. We investigate this process from the final boundary tokens backward: which internal computations support native timestamp generation, where boundary-related information remains available before generation, and how post-training reshapes these dependencies. Our analysis provides two complementary views of boundary formation. Intervention-based route tracing identifies computations that affect native timestamp generation, whereas probing methods measure information that can be recovered from intermediate representations. These two measurements capture different properties of the model: recoverable information does not necessarily correspond to information used by the native generator, and dependence on prompt states does not imply reliance on language alone, since these states may already contain integrated visual information. We perform our analysis on matched TimeLens-SFT and TimeLens-8B checkpoints, which share the same decoder architecture and visual input protocol. Raw Qwen3-VL provides an additional reference for visual-state analysis, while additional Boundary Flow results on raw Qwen3-VL and InternVL2.5 are reported in Appendix B.1. Let $\mathcal{V}$ denote visual-token positions, $\mathcal{P}$ the post-visual instruction and query positions before the final prompt position a, and $\mathcal{Y}_s$ and $\mathcal{Y}_e$ the positions where start and end timestamp tokens are predicted. We use $\mathcal{Y} = \mathcal{Y}_s \cup \mathcal{Y}_e$ for both boundary types. Since visual tokens precede the query in the causal input sequence, interpreting their hidden states requires distinguishing information availability from query-conditioned generation.

2.1 Boundary Formation through Information Pathways

Finding 1: Boundary generation relies on structured temporal pathways.

A temporal boundary may be supported by visual information transferred through intermediate prompt states or by direct access from boundary prediction positions. To identify which routes contribute to native timestamp generation, we introduce Boundary Flow Tracing. For a route $r : S \to T$ and decoder-layer window w, we mask attention from target queries $i \in T$ to source keys $j \in S$:

$$\widetilde{L}_{ij}^{(\ell)} = \begin{cases} -\infty, & \ell \in w,\ i \in T,\ j \in S, \\ L_{ij}^{(\ell)}, & \text{otherwise.} \end{cases} \tag{1}$$

The remaining attention weights are renormalized while model parameters remain unchanged. We first obtain the model's native timestamp prediction and then teacher-force the same output sequence with and without intervention. For boundary type $b \in \{s, e\}$, we measure the change in support for its native timestamp tokens:

$$\Delta_b^{r,w} = \frac{1}{|\mathcal{Y}_b|} \sum_{t \in \mathcal{Y}_b} \left[\log p(y_t \mid \mathbf{x}, y_{<t}) - \log p_{\mathrm{KO}(r,w)}(y_t \mid \mathbf{x}, y_{<t}) \right]. \tag{2}$$

A positive value indicates that removing route r decreases support for the native boundary tokens. Because attention weights are renormalized after masking, this quantity measures sensitivity to an intervention rather than an additive decomposition of information contribution.

Figure 1 summarizes the measured route sensitivities on the matched TimeLens checkpoints. Visual-to-prompt ($\mathcal{V} \to \mathcal{P}$) dependence is mainly concentrated in earlier and intermediate decoder depths, suggesting that visual evidence is progressively incorporated into prompt-side states. In contrast, prompt-to-boundary ($\mathcal{P} \to \mathcal{Y}$) dependence becomes more prominent at later depths and produces the largest measured effect among the tested routes, particularly for end prediction. Direct visual-to-boundary ($\mathcal{V} \to \mathcal{Y}$) dependence is comparatively smaller. These patterns are consistent with visual evidence passing through prompt states before boundary decoding. Prompt knockouts also disrupt query and instruction information, so they do not isolate visual mediation alone. The two boundary endpoints exhibit different formation patterns. Start prediction shows stronger sensitivity to earlier visual-to-prompt transfer, whereas end prediction relies more strongly on later prompt-to-boundary readout. Blocking direct start-to-end attention has limited effects relative to the principal prompt-to-boundary pathway (Appendix B.3); indirect autoregressive effects remain possible. Cross-model analyses show prompt-to-boundary dependence in both Qwen3-VL and InternVL2.5 (Figure 2). InternVL sensitivity peaks in middle layers, whereas Qwen retains strong late-layer sensitivity for start tokens. Appendix B.1 gives the cohort definitions and complete numerical effects.

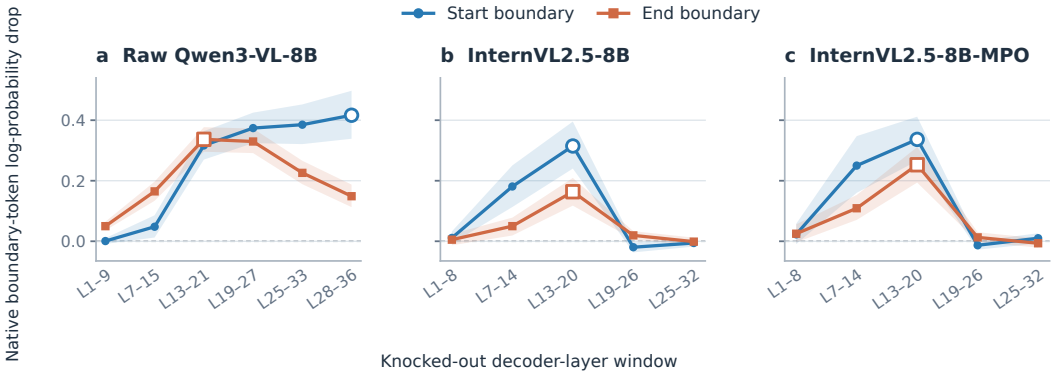

Figure 2: **Prompt-to-boundary Flow across model families.** Mean change in native timestamp-token log-probability after blocking prompt-to-boundary attention in each decoder-layer window. Raw Qwen3-VL-8B uses 512 native-correct examples; InternVL2.5-8B and InternVL2.5-8B-MPO use the same 512 common-correct examples. Start and end are shown separately. Error bands indicate 95% bootstrap intervals. Positive values indicate that the blocked route supported the native token.

2.2 Intermediate States Preserve Temporal Evidence

Finding 2: Intermediate visual states retain temporal cues accessible to query-conditioned readouts.

Boundary Flow Tracing identifies pathways that affect native timestamp generation, but it does not reveal where boundary-related information is available before the final decoding stage. We therefore complement route interventions with layer-wise probing. Unlike interventions, probes measure whether information can be recovered from a representation under a restricted readout, rather than whether the native generator directly uses that information. We first examine the final prompt position a before any timestamp is generated. At each decoder layer ℓ, an independent linear probe predicts normalized start and end boundaries from its hidden representation:

$$\widehat{\mathbf{b}}^{(\ell)} = W^{(\ell)}\mathrm{LN}\left(\mathbf{h}_a^{(\ell)}\right) + c^{(\ell)}, \qquad \mathbf{b} = [s/D, e/D], \tag{3}$$

where D denotes the video duration. The host model remains frozen, and all probes are trained and evaluated on video-disjoint splits. As shown in Figure 3(a), boundary information is recoverable before generation, but its decodability varies across decoder depth rather than monotonically increasing toward the final layer. These measurements characterize the accessibility of boundary information to the fitted readout before timestamp generation, rather than all information contained in each state. We next investigate whether intermediate visual states themselves retain temporal evidence aligned with the grounding query. Because visual tokens appear before query tokens in the causal sequence, their representations cannot directly encode the current query-specific answer. We therefore evaluate a query-conditioned visual readout that combines intermediate visual states with a query representation obtained from a separate query-only forward pass. The readout predicts temporal bins corresponding to candidate start and end locations. Position-only, query-only, and visual-only readouts serve as restricted baselines, while video replacement and temporal-bin shuffling evaluate whether the retrieved evidence depends on visual content and its temporal organization. Probe definitions, complete visual-bin results, and control experiments are provided in Appendix B.2.

Across the examined hosts, query-conditioned visual retrieval is weak in early layers, improves in intermediate layers, and does not continue improving toward the final layer (Figure 3(b)). At intermediate layers, the paired visual-query readout exceeds the position-only prior and responds to video replacement and temporal-bin shuffling (Appendix B.2). However, the absolute localization accuracy remains limited. Therefore, intermediate visual states do not directly act as an independent grounding module; instead, they preserve localized temporal cues that can be recovered by a restricted query-conditioned readout. Together with the Boundary Flow analysis, these results reveal a distinction between *availability* and *utilization*: temporal evidence can exist in intermediate states before generation, while the native model relies on specific downstream pathways to convert

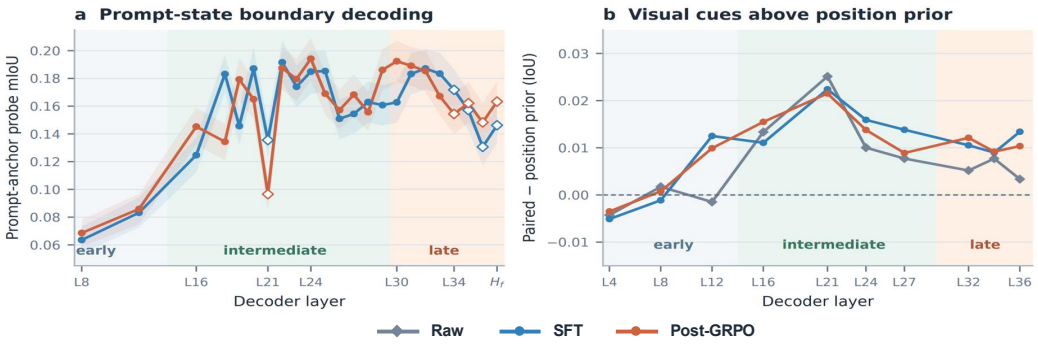

Figure 3: **Intermediate representations preserve recoverable temporal evidence.** **(a)** Prompt-anchor probe mIoU for SFT and post-GRPO. **(b)** Query-conditioned visual-bin retrieval compared with the position-only prior. Probes measure recoverability from intermediate states, not their causal use during native timestamp generation.

this evidence into timestamp tokens. This observation motivates Temporal Revisit, which explicitly returns intermediate visual information to later decoding stages.

2.3 Further Analysis: How Post-training Reshapes Boundary Formation

Beyond characterizing the native boundary formation process, we further investigate how post-training changes the internal mechanisms supporting temporal grounding. The matched TimeLens-SFT and TimeLens-8B checkpoints provide a controlled setting to examine whether improved localization is associated with a general strengthening of boundary representations or with selective changes in specific information pathways. The probes do not reveal a uniform increase in boundary recoverability after GRPO. The reported prompt-state results show both increases and decreases across layers, while changes in the visual-bin readout are small (Figure 3; Appendix B.2). Similarly, Boundary Flow Tracing shows that route dependencies change differently across boundary endpoints and decoder depths rather than exhibiting a universal increase in visual-to-prompt or prompt-to-boundary influence (Figure 1). In these matched checkpoints, post-training changes the measured representations and route sensitivities selectively rather than uniformly strengthening temporal boundary signals. Additional analyses of post-training effects, including native failures and rescued cases, are provided in Appendix B.4. Together, these analyses identify recoverable temporal cues in intermediate visual states and model-specific dependencies supporting native boundary generation. This raises a further question: can such internal evidence be explicitly revisited and reused to improve boundary generation without replacing the model's native timestamp decoder? Section 3 introduces Temporal Revisit to address this question.

3 Temporal Revisit

Section 2 identifies recoverable temporal cues in intermediate visual states and structured dependencies supporting native timestamp generation. We hypothesize that returning these cues to later visual states can improve boundary generation by the frozen decoder. We further hypothesize that prediction-conditioned reuse with preservation training can improve localization while limiting changes to the native outputs.

We introduce Temporal Revisit as a learned intervention to examine these hypotheses. BTR tests broad evidence reuse without an initial prediction. HTR examines prediction-conditioned reuse with preservation constraints; its "hypothesis" is the native interval used to guide refinement. Both variants return intermediate visual information to later visual states while retaining the native timestamp generator. The vision encoder, decoder, and language-head weights remain frozen; only the revisit modules are trained.

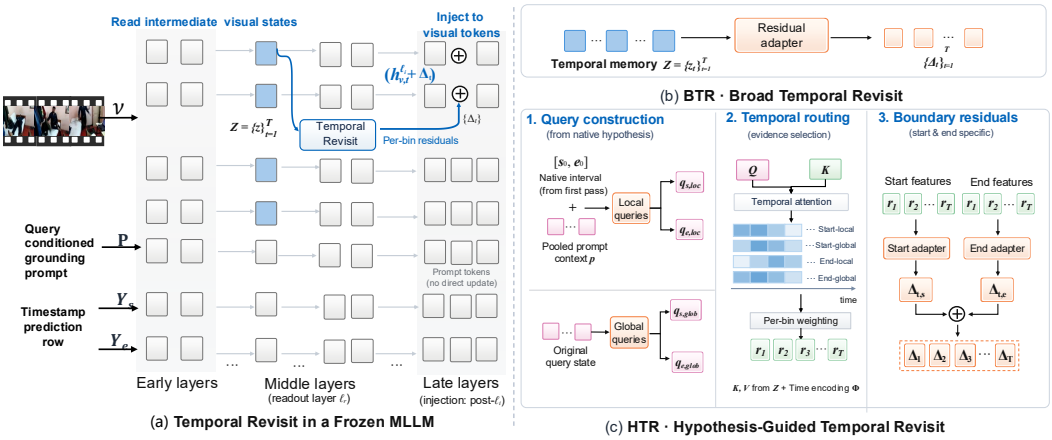

Figure 4: **Overview of TEMPORAL REVISIT.** Temporal Revisit reuses intermediate visual evidence in a frozen video MLLM. Intermediate visual states are aggregated into a temporal memory and injected into later visual states through learned residuals. **(b)** Broad Temporal Revisit (BTR) revisits all temporal bins in one generation pass. **(c)** Hypothesis-Guided Temporal Revisit (HTR) uses the native interval as a hypothesis and performs boundary-specific evidence retrieval through local and global queries. The frozen MLLM remains responsible for final timestamp generation.

3.1 REVISITING INTERMEDIATE TEMPORAL EVIDENCE

Figure 4 illustrates the shared mechanism behind both variants. Intermediate visual states are first aggregated into a temporal memory and then transformed into residual updates that are injected into later visual states. Let $\mathcal{V}_t$ denote the visual tokens assigned to temporal bin $t \in \{1, \dots, T\}$. At readout layer ℓ_r, we pool their hidden states:

$$\mathbf{z}_t = \frac{1}{|\mathcal{V}_t|} \sum_{i \in \mathcal{V}_t} \mathbf{h}_i^{(\ell_r)}, \qquad \mathbf{Z} = \{\mathbf{z}_t\}_{t=1}^{T}. \tag{4}$$

The revisit module transforms temporal memory states into residuals $\{\boldsymbol{\delta}_t\}_{t=1}^{T}$. At a later injection layer ℓ_i, each visual token receives the residual corresponding to its temporal bin:

$$\widetilde{\mathbf{h}}_i^{(\ell_i)} = \mathbf{h}_i^{(\ell_i)} + \boldsymbol{\delta}_{\tau(i)}, \qquad i \in \mathcal{V}, \tag{5}$$

where $\tau(i)$ maps visual token i to its temporal bin. The intervention only modifies later visual states; prompt states and timestamp prediction positions are not directly updated. The remaining frozen decoder layers then integrate the revisited visual information before generating the final timestamps.

3.2 BROAD AND HYPOTHESIS-GUIDED EVIDENCE REUSE

Broad Temporal Revisit. BTR serves as a broad evidence-access intervention: it tests whether exposing the frozen generator to intermediate temporal evidence can improve native boundary generation without requiring an explicit localization hypothesis. It returns information from all temporal bins and maps each memory state to a residual through a shared bottleneck adapter:

$$\boldsymbol{\delta}_t^{\mathrm{BTR}} = W_{\mathrm{up}}\, \mathrm{SiLU}(W_{\mathrm{down}}\, \mathrm{LN}(\mathbf{z}_t))\,. \tag{6}$$

The output projection W_{up} is initialized to zero, ensuring that the revisit module initially preserves the original host behavior. During inference, BTR generates timestamps in a single pass without requiring a native prediction. The frozen model integrates the revisited visual states with the query information through its original decoding process.

Hypothesis-Guided Temporal Revisit. While broad evidence access can expose useful temporal cues, revisiting every temporal region may also modify boundaries that are already accurate. HTR therefore treats the model's native interval $[s_0, e_0]$ as a hypothesis and performs selective evidence

reuse. For each boundary $b \in \{s, e\}$, HTR constructs local and global queries. The local query combines the native interval with prompt-side states from the second pass, while the global query retrieves evidence based on the original query representation:

$$\mathbf{q}_{b,\text{loc}} = W_b^P \overline{\mathbf{h}}_P + W_b^I \phi_I(s_0, e_0), \qquad \mathbf{q}_{b,\text{glob}} = W_b^Q \mathbf{h}_q. \tag{7}$$

Here, $\overline{\mathbf{h}}_P$ pools prompt states following visual tokens, ϕ_I embeds the native interval, and $\mathbf{h}_q$ represents the original query state. The local branch uses the predicted interval to guide retrieval, while the global branch retrieves evidence without relying on the native boundary hypothesis. Both branches attend over temporal memory with positional temporal encoding and generate boundary-specific residuals, which are injected through Eq. 5. The frozen MLLM, rather than an independent correction head, makes the final timestamp decision. Routing details are provided in Appendix D.3.

3.3 Training and Inference

Only the revisit modules are trained, while the vision encoder, decoder, and language head remain frozen. BTR is trained with timestamp supervision under teacher forcing and performs evidence reuse in a single generation pass at inference. HTR additionally uses the native prediction as a hypothesis, with preservation and routing losses during training, and performs one refinement pass at inference. Ground-truth timestamps are used only during training. The principal BTR and HTR configurations read intermediate states after layers 21 and 16, respectively, and inject residuals after layer 34. These locations are selected according to the intermediate stages where Section 2 identifies recoverable temporal evidence. Further training and implementation details are provided in Appendix D.3.

4 Experiments

We evaluate the evidence-reuse hypothesis through localization performance and post-intervention Boundary Flow Tracing. We then examine whether prediction-conditioned reuse with preservation training improves grounding while limiting prediction changes. Comparisons across matched post-training stages test how these behaviors depend on the native host.

4.1 Experimental Setup

Benchmarks and metrics. We evaluate Temporal Revisit on the QVHighlights–TimeLens, Charades–TimeLens, and ActivityNet–TimeLens subsets of TimeLens-Bench (Zhang et al., 2026). We report mIoU and recall at IoU thresholds 0.5, 0.7, and 0.9. Complete results are provided in the Appendix.

Model hosts. Our primary evaluation uses matched TimeLens-SFT and TimeLens-8B checkpoints (Zhang et al., 2026). TimeLens-8B is built on Qwen3-VL-8B-Instruct (Bai et al., 2025); the matched checkpoints share the same decoder architecture and visual input protocol. We further evaluate Qwen3-VL-8B-Instruct and MiMo-VL-7B-RL (Xiaomi LLM-Core Team, 2025) as additional frozen hosts to examine whether evidence reuse extends beyond a single training pipeline.

Evaluation protocol. All comparisons preserve native timestamp generation: the host MLLM remains frozen and produces the final boundaries. Only Temporal Revisit modules are trained. BTR performs evidence reuse in one pass, whereas HTR first obtains a native interval and then performs one hypothesis-guided refinement pass. Therefore, HTR introduces one additional decoding pass compared with native generation, using the same frozen backbone and video input.

4.2 Temporal Revisit Improves Native Boundary Generation

Table 1 reports the paired grounding results and includes both BTR and HTR on the matched TimeLens hosts. On TimeLens-SFT, BTR raises mIoU by 3.92, 2.85, and 2.19 percentage points on QVHighlights–TimeLens, Charades–TimeLens, and ActivityNet–TimeLens. Its improvements at the higher IoU thresholds also support better boundary localization. BTR improves mIoU on Qwen3-VL and MiMo-VL across all three benchmarks, although individual recall thresholds do not improve uniformly. The stronger post-GRPO host responds differently. BTR changes mIoU by -0.34, $+0.02$,

Table 1: **Temporal grounding with frozen hosts.** Native and revisited systems are evaluated on identical instances. Values are reported point estimates. Both revisit variants are shown for TimeLens-SFT and TimeLens-8B; the additional hosts evaluate BTR. BTR uses one generation pass and HTR uses two.

	QVHighlights–TimeLens				Charades–TimeLens				ActivityNet–TimeLens			
Host / Method	R@0.5	R@0.7	R@0.9	mIoU	R@0.5	R@0.7	R@0.9	mIoU	R@0.5	R@0.7	R@0.9	mIoU
MiMo-VL-7B-RL	40.82	26.93	12.26	39.64	41.84	19.42	7.94	39.74	30.89	18.33	9.53	28.68
+ BTR	43.22	27.84	13.82	40.61	46.12	21.41	11.24	41.56	33.76	20.16	8.16	31.42
Qwen3-VL-8B	64.57	49.90	32.90	59.59	53.58	28.01	9.52	48.57	51.27	34.42	17.27	46.94
+ BTR	63.98	50.36	36.73	60.50	60.39	33.10	17.01	52.68	53.64	36.20	22.62	48.80
TimeLens-SFT	59.38	45.04	32.38	57.04	59.53	30.75	19.54	51.68	53.49	36.09	23.04	48.77
+ BTR	64.89	50.49	34.85	60.96	62.77	33.36	19.87	54.53	56.27	38.56	23.29	50.96
+ HTR	59.77	45.55	32.64	57.39	59.77	30.87	19.57	51.76	53.56	36.18	23.11	48.83
TimeLens-8B (GRPO)	71.19	55.55	37.12	65.39	62.98	35.47	17.90	55.19	57.49	40.00	23.89	52.58
+ BTR	71.12	54.90	36.79	65.05	63.22	35.09	18.85	55.21	57.31	40.11	24.04	52.43
+ HTR	71.39	55.87	37.31	65.70	62.98	36.30	19.24	55.72	57.54	40.49	24.36	52.83

and -0.15 points, whereas HTR improves it by 0.31, 0.53, and 0.25 points. HTR also produces small gains on the SFT host, but these remain below BTR. Thus, broad evidence reuse is useful across several hosts but is not universally beneficial after post-training; hypothesis-guided refinement offers a different, more conservative operating behavior under the tested configurations.

4.3 Broad versus Hypothesis-Guided Evidence Reuse

Aggregate mIoU alone does not reveal whether an improvement comes from broad changes to many predictions or from a smaller number of targeted corrections. We therefore analyze prediction changes and their paired IoU outcomes (Appendix C.2). On the GRPO host, HTR modifies 10, 122, and 62 predictions across the three datasets, compared with 206, 634, and 1,072 for BTR. On Charades–TimeLens, for example, HTR improves 65 predictions and worsens 21, whereas BTR improves 289 and worsens 260. At IoU 0.7, GRPO+HTR also produces more rescues than regressions on all three datasets.

The two strategies therefore exhibit different gain–preservation profiles. On TimeLens-SFT, BTR achieves larger overall gains but also changes substantially more predictions, including more degraded cases. HTR makes fewer modifications, particularly on the post-GRPO host, but its aggregate improvements are correspondingly smaller. These results characterize a trade-off between stronger intervention and preservation of native behavior, rather than suggesting that fewer changes are inherently preferable. Importantly, BTR and HTR differ not only in retrieval strategy, but also in readout depth, module capacity, training objectives, and generation passes. The comparison should therefore be interpreted at the level of the complete designs, rather than as an isolated effect of hypothesis-guided retrieval.

4.4 Revisit Changes Boundary-Generation Pathways

To examine whether Temporal Revisit changes how the frozen decoder accesses temporal evidence, we repeat Boundary Flow Tracing on TimeLens-SFT before and after BTR. As shown in Figure 5, the native model exhibits strong prompt-to-boundary dependence, consistent with Section 2.1. After BTR, start and end boundary tokens become more sensitive to direct visual-to-boundary attention, indicating that the decoder makes greater use of the revisited visual states during timestamp generation.

A matched-sequence analysis provides a more targeted test of this change. When Native and BTR are evaluated on the same BTR-generated timestamp sequence, BTR introduces 0.10915 additional log-probability support for the generated boundary tokens. Blocking the visual-to-boundary pathway reduces this added support to 0.01159, eliminating 0.09756 of the increase (Appendix C.4). This result concerns token-level support rather than a decomposition of the task-level mIoU gain, but it is consistent with BTR increasing the decoder's reliance on revisited visual evidence.

Table 2: **Control comparisons for Temporal Revisit on QVHighlights–TimeLens.** Δ denotes the mIoU change relative to the corresponding native host.

Host	Method	Passes	Trainable Params.	mIoU	Δ
TimeLens-SFT	Native	1	0	57.04	–
	BTR	1	270,336	**60.96**	**+3.92**
	Random BTR†	1	270,336	58.13	+1.09
	LoRA (rank 5)	1	2,396,160	58.78	+1.74
TimeLens-8B	Native	1	0	65.39	–
	Native recheck	2	0	65.38	-0.01
	HTR	2	3,312,420	**65.70**	**+0.31**

†Random BTR reuses the trained BTR module and only permutes the temporal assignment of its residuals at inference. For TimeLens-8B, Native recheck and HTR use the same native draft, second-pass instruction, and greedy decoding configuration; only HTR enables the learned revisit module. The LoRA control uses rank 5 on the Q/V projections of all 36 decoder layers and has 2,396,160 trainable parameters, approximately $8.86\times$ the 270,336 trainable parameters of BTR.

Control comparisons. We further test whether the observed grounding improvements can be explained by simpler alternatives. As shown in Table 2, temporally shuffling the learned BTR residuals substantially reduces the gain, indicating that residual injection alone is insufficient. A same-data LoRA control, despite using approximately $8.86\times$ more trainable parameters, also remains below BTR. For the post-GRPO host, a matched native second pass leaves mIoU essentially unchanged, whereas HTR improves it by 0.32 points. Together, these controls disfavor explanations based solely on generic parameter adaptation, temporally unaligned residual injection, or the additional decoding pass. Full control settings and additional HTR tracing are provided in Appendices C.3 and C.4.

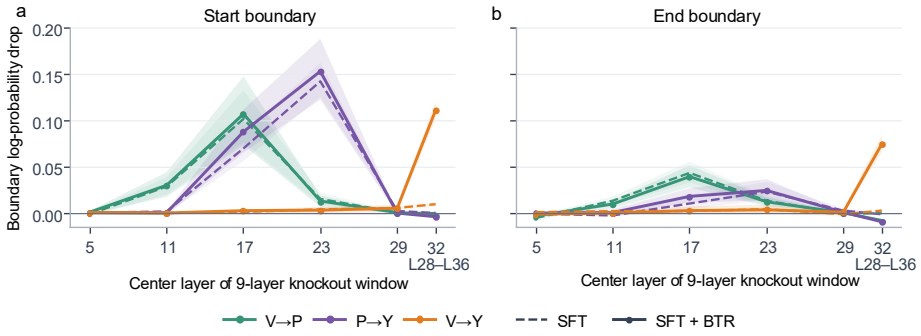

Figure 5: **Boundary Flow before and after BTR on TimeLens-SFT.** BTR increases sensitivity to direct visual-to-boundary attention, indicating greater use of revisited visual states during native timestamp generation. Solid curves denote BTR and dashed curves denote the frozen native host.

5 Conclusion

We studied how generative video MLLMs form temporal boundaries by separating information recoverable from intermediate representations from the pathways that support native timestamp generation. Boundary Flow Tracing reveals structured, endpoint- and depth-dependent route sensitivities in TimeLens, with related prompt-to-boundary dependence in Qwen3-VL and InternVL2.5. Layer-wise probing further shows that temporal cues remain recoverable before generation, while matched TimeLens checkpoints indicate that post-training reshapes these signals selectively rather than uniformly. Motivated by this distinction between evidence availability and utilization, we introduced Temporal Revisit, which returns intermediate visual evidence to later visual states while keeping the host and native timestamp generator frozen. BTR improves grounding across several hosts and increases direct visual-to-boundary dependence, while HTR yields smaller but more selective improvements on the post-GRPO host. Targeted controls further disfavor simpler explanations based solely on temporal misalignment, generic parameter adaptation, or an additional decoding pass. Together, these results connect mechanistic analysis of boundary formation with interventions on internal evidence access. Future work should develop adaptive mechanisms for deciding when and where to revisit evidence while preserving reliable predictions.

AI USE STATEMENT

During the preparation of this manuscript, the authors used AI-based tools for assistance with language refinement, organization of technical writing, improving the clarity of explanations, and literature retrieval and discovery. All retrieved references and related scientific claims were independently checked and verified by the authors. All scientific ideas, experimental designs, implementations, results, analyses, and conclusions were developed and verified by the authors. The authors take full responsibility for the content of this manuscript.

REPRODUCIBILITY STATEMENT

We provide the implementation and evaluation details necessary to reproduce our analyses and Temporal Revisit experiments in the appendix. This includes the model checkpoints and benchmark splits used for evaluation, temporal grounding metrics, Boundary Flow Tracing protocols, probe construction, Temporal Revisit architectures, training objectives, optimization settings, layer selections, decoding procedures, and inference protocols. We also report the sampling criteria and cohort sizes used in mechanistic analyses, together with the video-cluster bootstrap procedure used for uncertainty estimation. Control experiments and layer-sensitivity analyses are included to clarify the dependence of the reported results on temporal alignment, intervention depth, and additional decoding passes. All reported benchmark results are obtained with the frozen host MLLM unless explicitly stated otherwise. Upon acceptance, we will release the code, training configurations, and evaluation scripts to facilitate reproduction of the reported results.

REFERENCES

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

Samyadeep Basu, Martin Grayson, Cecily Morrison, Besmira Nushi, Soheil Feizi, and Daniela Massiceti. Understanding information storage and transfer in multi-modal large language models. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, 2022. doi: 10.1162/coli_a_00422.

Shimin Chen, Xiaohan Lan, Yitian Yuan, Zequn Jie, and Lin Ma. TimeMarker: A versatile video-LLM for long and short video understanding with superior temporal localization ability. *arXiv preprint arXiv:2411.18211*, 2024.

Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. Dissecting recall of factual associations in auto-regressive language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 12216–12235. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.751.

Yongxin Guo, Jingyu Liu, Mingda Li, Dingxin Cheng, Xiaoying Tang, Dianbo Sui, Qingbin Liu, Xi Chen, and Kevin Zhao. VTG-LLM: Integrating timestamp knowledge into video LLMs for enhanced video temporal grounding. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(3):3302–3310, 2025. doi: 10.1609/aaai.v39i3.32341.

Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing moments in video with natural language. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5803–5812, 2017.

John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2733–2743. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-1275.

Bin Huang, Xin Wang, Hong Chen, Zihan Song, and Wenwu Zhu. VTimeLLM: Empower LLM to grasp video moments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14271–14280, 2024a.

De-An Huang, Shijia Liao, Subhashree Radhakrishnan, Hongxu Yin, Pavlo Molchanov, Zhiding Yu, and Jan Kautz. LITA: Language instructed temporal-localization assistant. In *Computer Vision – ECCV 2024*, volume 15122 of *Lecture Notes in Computer Science*, pp. 202–218. Springer, 2024b. doi: 10.1007/978-3-031-73039-9_12.

Minji Kim, Taekyung Kim, and Bohyung Han. Map the flow: Revealing hidden pathways of information in VideoLLMs. In *International Conference on Learning Representations*, 2026.

Jie Lei, Tamara L. Berg, and Mohit Bansal. Detecting moments and highlights in videos via natural language queries. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Zeqian Li, Shangzhe Di, Zhonghua Zhai, Weilin Huang, Yanfeng Wang, and Weidi Xie. Universal video temporal grounding with generative multi-modal large language models. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Kevin Qinghong Lin, Pengchuan Zhang, Joya Chen, Shraman Pramanick, Difei Gao, Alex Jinpeng Wang, Rui Yan, and Mike Zheng Shou. UniVTG: Towards unified video-language temporal grounding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2794–2804, 2023.

Lingchen Meng, Jianwei Yang, Rui Tian, Xiyang Dai, Zuxuan Wu, Jianfeng Gao, and Yu-Gang Jiang. DeepStack: Deeply stacking visual tokens is surprisingly simple and effective for LMMs. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-0739.

Shuhuai Ren, Linli Yao, Shicheng Li, Xu Sun, and Lu Hou. TimeChat: A time-sensitive multimodal large language model for long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14313–14323, 2024.

Haibo Wang, Zhiyang Xu, Yu Cheng, Shizhe Diao, Yufan Zhou, Yixin Cao, Qifan Wang, Weifeng Ge, and Lifu Huang. Grounded-VideoLLM: Sharpening fine-grained temporal grounding in video large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 959–975. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-emnlp.50.

Xizi Wang, Feng Cheng, Ziyang Wang, Huiyu Wang, Md Mohaiminul Islam, Lorenzo Torresani, Mohit Bansal, Gedas Bertasius, and David Crandall. TimeRefine: Temporal grounding with time refining video LLM. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 5067–5078, 2026.

Xiaomi LLM-Core Team. MiMo-VL technical report. *arXiv preprint arXiv:2506.03569*, 2025.

Zuhao Yang, Yingchen Yu, Yunqing Zhao, Shijian Lu, and Song Bai. TimeExpert: An expert-guided video LLM for video temporal grounding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 24286–24296, 2025.

Zeping Yu and Sophia Ananiadou. Understanding multimodal llms: the mechanistic interpretability of llava in visual question answering. *arXiv preprint arXiv:2411.10950*, 2024.

Zhuoran Yu and Yong Jae Lee. How multimodal LLMs solve image tasks: A lens on visual grounding, task reasoning, and answer decoding. In *Conference on Language Modeling*, 2025.

Xiangyu Zeng, Kunchang Li, Chenting Wang, Xinhao Li, Tianxiang Jiang, Ziang Yan, Songze Li, Yansong Shi, Zhengrong Yue, Yi Wang, Yali Wang, Yu Qiao, and Limin Wang. Timesuite: Improving mllms for long video understanding via grounded tuning. In *International Conference on Learning Representations*, 2025.

Hao Zhang, Aixin Sun, Wei Jing, and Joey Tianyi Zhou. Span-based localizing network for natural language video localization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 6543–6554. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.acl-main.585.

Jun Zhang, Teng Wang, Yuying Ge, Yixiao Ge, Xinhao Li, Ying Shan, and Limin Wang. TimeLens: Rethinking video temporal grounding with multimodal LLMs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

Zhi Zhang, Srishti Yadav, Fengze Han, and Ekaterina Shutova. Cross-modal information flow in multimodal large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19781–19791, 2025.

APPENDIX

- § A: **Related Work**
- § B: **Additional Analysis of Temporal Boundary Formation**
  - § B.1: Cross-backbone Boundary Information Analysis
  - § B.2: Probe Definitions and Extended Results
  - § B.3: Start-End Boundary Dependency
  - § B.4: Post-training Effects on Boundary Formation
- § C: **Additional Temporal Revisit Experiments**
  - § C.1: Supplementary Benchmark Results
  - § C.2: Prediction Changes and Interval Behavior
  - § C.3: Control Experiment Details
  - § C.4: Post-Revisit Boundary Flow
  - § C.5: Layer Sensitivity Analysis of Temporal Revisit
- § D: **Implementation Details**
  - § D.1: Models, Datasets, and Input Protocols
  - § D.2: Boundary Flow Tracing and Probe Settings
  - § D.3: Temporal Revisit Architecture, Training, and Inference
  - § D.4: Evaluation Protocol

A RELATED WORK

Generative video temporal grounding. Earlier VTG methods formulate localization through proposal retrieval, boundary regression, or direct moment prediction (Hendricks et al., 2017; Zhang et al., 2020; Lei et al., 2021; Lin et al., 2023). With video MLLMs, recent work has advanced temporal grounding through timestamp-aware representations, temporal tokens, grounding-oriented instruction tuning, and long-video modeling (Ren et al., 2024; Huang et al., 2024b; Chen et al., 2024; Wang et al., 2025; Huang et al., 2024a; Guo et al., 2025; Zeng et al., 2025; Li et al., 2025). More recent studies further investigate data quality, reinforcement-based post-training, and task-specific routing for timestamp generation (Zhang et al., 2026; Yang et al., 2025). Together, these works improve grounding through representation, training, and architectural design, but primarily study external localization behavior. Our complementary question is how native start and end timestamps are formed internally, and whether the resulting internal evidence can be reused without replacing the timestamp generator.

Mechanistic analysis of multimodal models. Probing methods characterize information recoverable from hidden representations, whereas interventions test sensitivity to specific computations (Hewitt & Liang, 2019; Belinkov, 2022; Geva et al., 2023). Recent mechanistic studies of image MLLMs have examined visual information storage, causal transfer, cross-modal integration, and layer-wise processing (Basu et al., 2024; Zhang et al., 2025; Yu & Ananiadou, 2024; Yu & Lee, 2025). Most closely, Kim et al. (2026) analyze cross-frame interaction and video–language information flow in VideoLLMs. These studies motivate our diagnostics, but do not explain how a generative VTG model forms two explicit temporal boundaries. We therefore analyze recoverability and route sensitivity separately for start and end timestamps and across post-training stages.

Temporal refinement and intermediate representation reuse. Temporal localization can also be improved by refining predictions or exposing visual information at multiple model depths. TimeRefine generates successive boundary refinements autoregressively, with an auxiliary prediction head used only during training (Wang et al., 2026). DeepStack distributes visual tokens across language-model layers to improve multimodal processing (Meng et al., 2024). These methods provide relevant precedents for refinement and multi-layer visual access, but do not ask whether temporal evidence already present in intermediate decoder states can be reused by the native timestamp generator. TEMPORAL REVISIT first identifies such internal evidence and then returns it to later visual states of

a frozen MLLM; its hypothesis-guided variant further uses the native interval to selectively retrieve evidence while preserving the host's own timestamp generation.

B Additional Analysis of Temporal Boundary Formation

This section provides additional analyses of how video MLLMs form temporal boundaries. We study whether the observed boundary-generation pathways extend across model families, whether endpoint-specific differences can be explained by autoregressive dependencies, and how post-training reshapes internal boundary formation.

B.1 Cross-backbone Boundary Information Analysis

The main paper analyzes Boundary Flow Tracing on matched TimeLens-SFT and TimeLens-8B checkpoints. We further examine additional video MLLM families to evaluate whether the observed boundary-generation patterns depend on a specific architecture or grounding-oriented training procedure.

Because model families differ in architecture, decoder depth, and training objectives, route effects are interpreted within each host rather than as direct comparisons across models. We focus on whether timestamp generation exhibits structured dependencies between intermediate representations and boundary prediction positions.

B.1.1 Boundary Flow across Model Families

We extend Boundary Flow Tracing to raw Qwen3-VL-8B and the independent InternVL2.5 family. The Raw Qwen analysis uses 512 native-correct, timestamp-traceable QVHighlights query–video examples. InternVL2.5-8B and InternVL2.5-8B-MPO are evaluated on the same 512 common-correct query–video examples. Native correctness is defined by an interval IoU of at least 0.5. For every host, we teacher-force its own native timestamp sequence and apply the attention intervention defined in Eq. 2; ground-truth intervals are used to select the analysis cohort, not as replacement answers.

The prompt-to-boundary route uses prompt positions after the visual tokens as its source and native start or end prediction positions as its target. For InternVL, $\mathcal{P}^+$ denotes positions after the last visual token through the final prompt position. We test six nine-layer windows in the 36-layer Raw Qwen decoder and five eight-layer windows in the 32-layer InternVL decoders. Start and end effects are measured separately. Because the models differ in architecture and tokenization, we interpret each depth profile within its own host.

Figure 2 in the main paper shows the depth profiles; Table 3 supplies the complete numerical effects. Prompt-to-boundary sensitivity is present in both model families, although the strongest windows need not coincide across architectures.

B.1.2 Layer-wise Temporal Decodability on MiMo-VL

To examine temporal boundary recoverability beyond the TimeLens training pipeline, we analyze frozen MiMo-VL-7B through layer-wise temporal decodability measurements.

Figure 6 reports boundary linear-probe performance, boundary regression error, and native query-key temporal alignment across decoder layers. Boundary-related information becomes recoverable in later decoder layers, providing an additional cross-model reference that temporal structure can exist before explicit timestamp generation.

B.2 Probe Definitions and Extended Results

Prompt-position boundary probe. For TimeLens-SFT and TimeLens-8B, we extract the hidden state at the final prompt position before timestamp generation. At each examined decoder layer, an independent LayerNorm followed by an affine two-dimensional readout predicts the ground-truth start and end times normalized by video duration. The host is frozen. Each probe is fitted with mean squared error on the same 6,000 training examples and evaluated on the same 1,024 video-disjoint development examples. Training uses AdamW for 2,000 steps with batch size 256 and learning

Table 3: **Prompt-to-boundary Flow across model families.** Mean log-probability drop of each model's native start and end timestamp tokens after blocking $P \rightarrow Y$ attention in the indicated decoder-layer window. Positive values indicate that the blocked route supported the native token.

Host	Knockout window	Start	End
Raw Qwen3-VL-8B	L1–9	0.0007	0.0499
	L7–15	0.0480	0.1648
	L13–21	0.3166	0.3369
	L19–27	0.3740	0.3297
	L25–33	0.3852	0.2260
	L28–36	0.4164	0.1489
InternVL2.5-8B	L1–8	0.0111	0.0052
	L7–14	0.1810	0.0500
	L13–20	0.3148	0.1637
	L19–26	-0.0197	0.0195
	L25–32	-0.0058	-0.0013
InternVL2.5-8B-MPO	L1–8	0.0229	0.0249
	L7–14	0.2500	0.1091
	L13–20	0.3367	0.2528
	L19–26	-0.0132	0.0128
	L25–32	0.0101	-0.0066

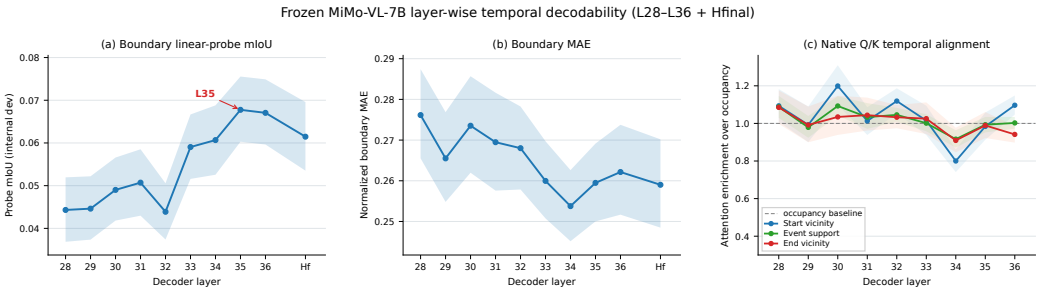

Figure 6: **Layer-wise temporal boundary decodability in MiMo-VL-7B.** (a) Boundary linear-probe mIoU, (b) boundary regression error, and (c) native query-key temporal alignment across decoder layers. The analysis provides an additional cross-model reference for recoverable temporal structure before explicit timestamp generation.

rate 3×10^{-3}. We evaluate the fixed final probe at every layer rather than selecting a different checkpoint per layer.

Query-conditioned visual-bin probe. We additionally examine raw Qwen3-VL-8B, TimeLens-SFT, and TimeLens-8B. Visual tokens at each sampled timestamp are mean-pooled into a temporal bin. The query representation is extracted from a separate frozen text-only forward pass, so the query carrier has not seen the video. Visual and query features undergo the same fixed Gaussian projection from 4,096 to 512 dimensions. The projection is not fitted to boundary labels.

For each boundary, the probe scores sampled temporal bins using linear terms in the visual state, a visual–query interaction, the query and bin time, and a position term. Four restricted versions use (i) position alone, (ii) position and query, (iii) position and visual state, or (iv) all terms (paired). The training target is the sampled bin nearest the ground-truth boundary. Separate start and end scores are trained using bin cross-entropy. Each host uses 4,096 training examples and 512 video-disjoint development examples. Training uses AdamW for 1,500 steps with batch size 24 and learning rate 10^{-3}; the final step is evaluated at every sampled layer.

We also evaluate the fitted paired probe after a duration- and bin-matched video replacement or a shuffle of temporal-bin assignments. These controls change the supplied visual evidence without

retraining the probe. A same-video query replacement is recorded separately because it also changes the query and its target; its score is therefore not a fixed-target degradation comparison.

Table 4: **Complete visual-bin probe results on the 512-example development set.** Entries are mean interval IoU. Pos., Q-only, and V-only denote the restricted position, query, and visual readouts. V-swap and Bin-shuf. apply video replacement and temporal-bin shuffling to the fitted paired probe. The raw, SFT, and GRPO hosts use the same layer grid.

Host	Layer	Paired	Pos.	Q-only	V-only	V-swap	Bin-shuf.
Raw	L4	0.085	0.089	0.078	0.082	0.076	0.060
Raw	L8	0.091	0.089	0.079	0.079	0.073	0.071
Raw	L12	0.088	0.089	0.075	0.083	0.073	0.072
Raw	L16	0.103	0.089	0.078	0.077	0.075	0.072
Raw	L21	0.114	0.089	0.082	0.081	0.078	0.076
Raw	L24	0.099	0.089	0.079	0.077	0.087	0.072
Raw	L27	0.097	0.089	0.085	0.080	0.078	0.073
Raw	L32	0.095	0.089	0.082	0.079	0.082	0.071
Raw	L34	0.097	0.089	0.081	0.082	0.082	0.076
Raw	L36	0.093	0.089	0.077	0.082	0.073	0.074
SFT	L4	0.084	0.089	0.078	0.081	0.078	0.061
SFT	L8	0.088	0.089	0.079	0.080	0.077	0.070
SFT	L12	0.102	0.089	0.072	0.083	0.071	0.071
SFT	L16	0.100	0.089	0.077	0.079	0.071	0.079
SFT	L21	0.112	0.089	0.087	0.078	0.077	0.074
SFT	L24	0.105	0.089	0.082	0.081	0.082	0.078
SFT	L27	0.103	0.089	0.084	0.083	0.079	0.074
SFT	L32	0.100	0.089	0.080	0.083	0.077	0.081
SFT	L34	0.098	0.089	0.078	0.080	0.079	0.080
SFT	L36	0.103	0.089	0.074	0.078	0.077	0.066
GRPO	L4	0.086	0.089	0.079	0.080	0.075	0.064
GRPO	L8	0.090	0.089	0.079	0.080	0.076	0.068
GRPO	L12	0.099	0.089	0.076	0.081	0.073	0.075
GRPO	L16	0.105	0.089	0.078	0.069	0.075	0.078
GRPO	L21	0.111	0.089	0.086	0.089	0.078	0.079
GRPO	L24	0.103	0.089	0.081	0.082	0.082	0.077
GRPO	L27	0.098	0.089	0.083	0.084	0.078	0.080
GRPO	L32	0.101	0.089	0.084	0.085	0.079	0.080
GRPO	L34	0.098	0.089	0.079	0.082	0.083	0.075
GRPO	L36	0.100	0.089	0.077	0.083	0.074	0.069

B.3 START-END BOUNDARY DEPENDENCY

Because end timestamps are generated after start timestamps, their different depth profiles could potentially arise from direct autoregressive dependency. We therefore examine whether direct access from generated start tokens explains endpoint-specific boundary formation.

We block attention from start timestamp positions to end prediction positions across decoder-layer windows while keeping other pathways unchanged. The resulting effects remain substantially smaller than the principal prompt-to-boundary dependencies reported in the main paper.

These results suggest that direct start-to-end attention alone cannot explain the observed endpoint-specific patterns, although indirect autoregressive effects may still contribute.

B.4 POST-TRAINING EFFECTS ON BOUNDARY FORMATION

The main paper shows that GRPO changes boundary-related pathways selectively rather than uniformly strengthening all temporal signals. We further analyze three prediction cohorts: examples failed by both hosts, examples rescued by GRPO, and examples degraded after GRPO.

For each cohort, we measure the strongest tested prompt-to-boundary dependency. The analysis characterizes pathway sensitivity associated with different prediction outcomes rather than identifying causal sources of errors.

B.4.1 JOINT-ROUTE DIAGNOSTIC

Finally, we perform a joint-route diagnostic to examine whether combining visual-to-intermediate and intermediate-to-boundary interventions introduces nonlinear sensitivity.

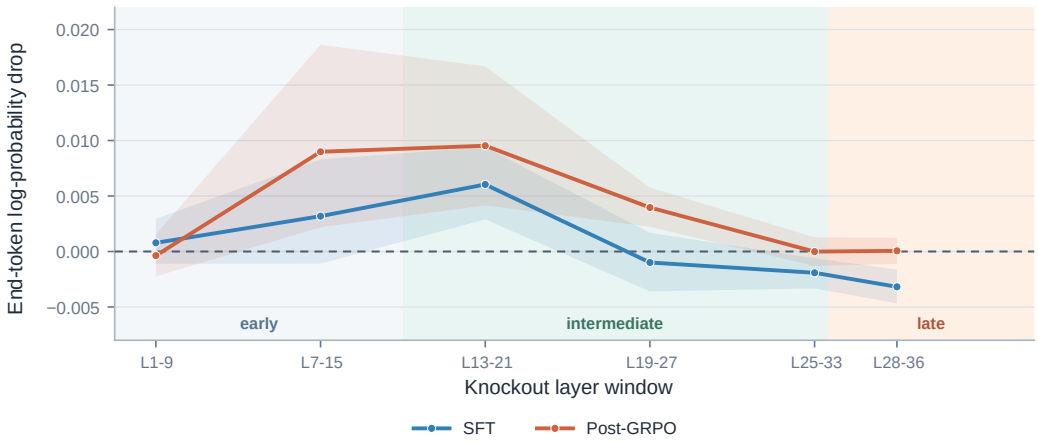

Figure 7: **Start-to-end autoregressive dependency analysis.** Blocking direct attention from generated start-token positions to end prediction positions produces limited effects compared with the main boundary-generation pathways.

Table 5: **Prompt-to-boundary dependency across post-training cohorts.** Values denote the strongest tested mean log-probability drop under attention-edge knockout.

Cohort	Host	Start	End	Peak window
Common failure	SFT	0.1728	0.1351	L19–27/L13–21
Common failure	GRPO	0.1136	0.1377	L19–27/L13–21
GRPO rescued	SFT	0.1292	0.1639	L19–27/L13–21
GRPO rescued	GRPO	0.0885	0.1742	L19–27/L25–33
GRPO degraded	SFT	0.1617	0.0887	L19–27/L13–21
GRPO degraded	GRPO	0.1495	0.1284	L19–27/L19–27

The interaction term measures the difference between the joint knockout effect and the sum of individual route effects. It describes nonlinear sensitivity among tested pathways and should not be interpreted as a formal causal mediation estimate.

Table 6: **Joint-route interaction diagnostic.** Maximum observed interaction effects among tested route pairs.

Host	Boundary	Maximum interaction
SFT	Start	+0.0052
SFT	End	-0.0013
GRPO	Start	+0.0064
GRPO	End	+0.0010

C Additional Temporal Revisit Experiments

This section provides additional analyses of Temporal Revisit beyond the main results. We report complementary benchmark results, analyze how revisit strategies modify predictions, compare against simpler alternatives, examine post-revisit boundary-generation pathways, and study the sensitivity to layer placement.

C.1 Supplementary Benchmark Results

The main paper reports the principal Temporal Revisit configurations and selected metrics for readability. This section provides the omitted R@0.3 metric and the complete recall metrics for

all TimeLens-host configurations reported in the main table. The common implementation and evaluation protocol is specified in Appendix D.

Complete TimeLens-host results. Tables 7, 8, and 9 report the complete recall metrics and mIoU for all TimeLens-SFT and TimeLens-8B configurations summarized in the main table.

Table 7: **Supplementary results on QVHighlights-TimeLens.**

Method	R@0.3	R@0.5	R@0.7	R@0.9	mIoU
TimeLens-SFT	69.31	59.38	45.04	32.38	57.04
SFT+BTR	73.65	64.89	50.49	34.85	60.96
SFT+HTR	69.63	59.77	45.55	32.64	57.39
TimeLens-8B	80.14	71.19	55.55	37.12	65.39
GRPO+BTR	79.49	71.12	54.90	36.79	65.05
GRPO+HTR	80.51	71.39	55.87	37.31	65.70

Table 8: **Supplementary results on Charades-TimeLens.**

Method	R@0.3	R@0.5	R@0.7	R@0.9	mIoU
TimeLens-SFT	74.01	59.53	30.75	19.54	51.68
SFT+BTR	76.75	62.77	33.36	19.87	54.53
SFT+HTR	74.01	59.77	30.87	19.57	51.76
TimeLens-8B	76.63	62.98	35.47	17.90	55.19
GRPO+BTR	76.93	63.22	35.09	18.85	55.21
GRPO+HTR	76.88	62.98	36.30	19.24	55.72

Table 9: **Supplementary results on ActivityNet-TimeLens.**

Method	R@0.3	R@0.5	R@0.7	R@0.9	mIoU
TimeLens-SFT	62.62	53.49	36.09	23.04	48.77
SFT+BTR	65.62	56.27	38.56	23.29	50.96
SFT+HTR	62.71	53.56	36.18	23.11	48.83
TimeLens-8B	68.04	57.49	40.00	23.89	52.58
GRPO+BTR	67.80	57.31	40.11	24.04	52.43
GRPO+HTR	68.82	57.54	40.49	24.36	52.83

Additional frozen hosts. Table 10 summarizes additional frozen-host evaluations on Qwen3-VL-8B and MiMo-VL-7B-RL. These results provide a cross-pipeline check of Temporal Revisit rather than a separate benchmark comparison. On MiMo-VL-7B-RL, both BTR and HTR improve or maintain the native grounding performance across the evaluated TimeLens subsets. These results further show that Temporal Revisit can be applied beyond the TimeLens training pipeline, while the magnitude of improvement depends on the host model and revisit strategy.

Table 10: **Additional frozen-host results.** mIoU comparisons on Qwen3-VL-8B and MiMo-VL-7B-RL. These evaluations examine whether evidence reuse generalizes beyond the TimeLens training pipeline.

Host	Method	QVH-TL	Charades-TL	ANet-TL
Qwen3-VL-8B	Native	59.59	48.57	46.94
	+BTR	60.50	52.68	48.80
MiMo-VL-7B-RL	Native	39.64	39.74	28.68
	+BTR	40.61	41.56	31.42
	+HTR	40.09	40.39	31.56

C.2 Prediction Changes and Interval Behavior

Temporal Revisit changes the visual states available to the frozen decoder, and the resulting predictions may either improve inaccurate intervals or alter existing boundaries. This section analyzes such changes beyond aggregate grounding scores.

C.2.1 Quantitative Prediction Changes

Table 11 reports the overall prediction changes under different revisit strategies. A changed prediction indicates that the generated interval differs from the native output, while improved and worsened predictions are determined by the corresponding IoU change.

Broad revisit modifies more predictions across hosts, reflecting its global evidence-access strategy. In contrast, HTR changes fewer intervals under its combined prediction-conditioned retrieval and preservation-training design. Because HTR also differs from BTR in module capacity, training objectives, and generation passes, these counts characterize the complete designs rather than isolating retrieval selectivity.

Table 11: **Overall prediction changes under Temporal Revisit.** Changed, Improved, and Worsened denote the numbers of modified predictions, IoU-improved predictions, and IoU-degraded predictions relative to the native host, respectively. A changed prediction can have unchanged IoU and therefore need not be counted as either Improved or Worsened.

Host	Dataset	Method	Changed	Improved	Worsened
SFT	QVH-TL	BTR	659	388	203
		HTR	38	24	7
SFT	Charades-TL	BTR	1549	787	558
		HTR	27	15	10
SFT	ANet-TL	BTR	2143	990	670
		HTR	18	9	6
GRPO	QVH-TL	BTR	206	71	111
		HTR	10	8	1
GRPO	Charades-TL	BTR	634	289	260
		HTR	122	65	21
GRPO	ANet-TL	BTR	1072	370	423
		HTR	62	31	12

Table 12 further reports rescue and regression statistics for high-overlap boundary changes. A rescue is defined as a prediction whose IoU changes from below 0.7 before revisit to at least 0.7 after revisit, whereas a regression represents the opposite transition.

These two analyses characterize complementary aspects of revisit behavior: overall modification frequency and threshold-crossing boundary corrections. Broad revisit introduces larger-scale changes with more rescues and regressions, whereas hypothesis-guided revisit performs fewer modifications while retaining selective corrections.

C.2.2 Revisit Behavior by Native Error Geometry

To further characterize the different behaviors of broad and selective evidence reuse, we stratify a shared Charades–GRPO cohort according to the geometry of the native interval. The revisit checkpoints in this diagnostic are independent of those used for the main results. Native predictions are grouped into non-overlap, too-wide, too-narrow, shifted, and native-correct cases.

Figure 8 shows distinct response patterns across these groups. BTR yields larger positive changes on no-overlap and too-wide predictions, whereas HTR shows its largest positive response on too-narrow intervals, although this smaller cohort has greater uncertainty. Both strategies can slightly degrade initially correct predictions, illustrating the trade-off between correcting native errors and preserving reliable boundaries.

Table 12: **High-overlap rescue and regression counts.**

Host / Method	Dataset	Rescue	Regression
SFT+BTR	QVH-TL	135	51
	Charades-TL	259	171
	ANet-TL	302	191
SFT+HTR	QVH-TL	9	1
	Charades-TL	6	2
	ANet-TL	5	1
GRPO+BTR	QVH-TL	19	29
	Charades-TL	62	75
	ANet-TL	84	79
GRPO+HTR	QVH-TL	5	0
	Charades-TL	31	3
	ANet-TL	26	4

These categories describe prediction geometry rather than causal error sources; the analysis is intended as a behavioral diagnostic of revisit strategies.

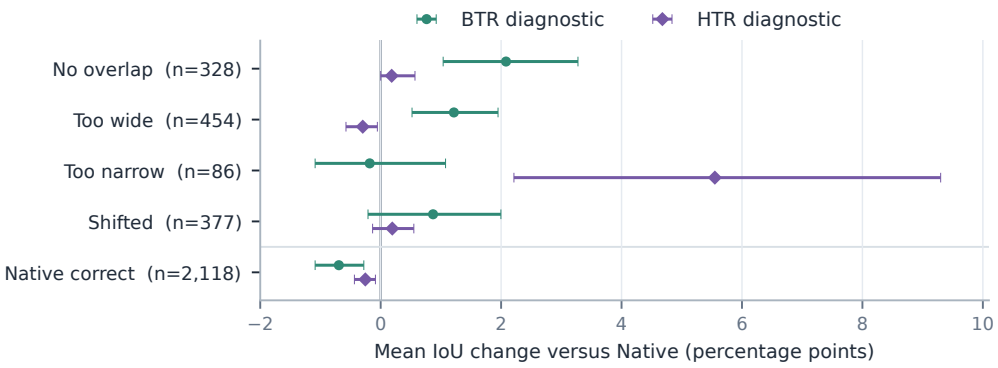

Figure 8: **Revisit behavior by native interval geometry on Charades–GRPO.** Points show mean paired IoU changes relative to the native prediction with 95% video-cluster bootstrap intervals. Categories describe native prediction geometry rather than causal error sources.

C.2.3 Qualitative Boundary Revision Examples

Figure 9 provides qualitative examples of Temporal Revisit behavior. BTR corrects several types of native boundary errors, including severe temporal misalignment, incomplete event coverage, and partial localization. The final example illustrates a limitation of broad evidence reuse: revisiting intermediate evidence may expand an interval that is already accurate.

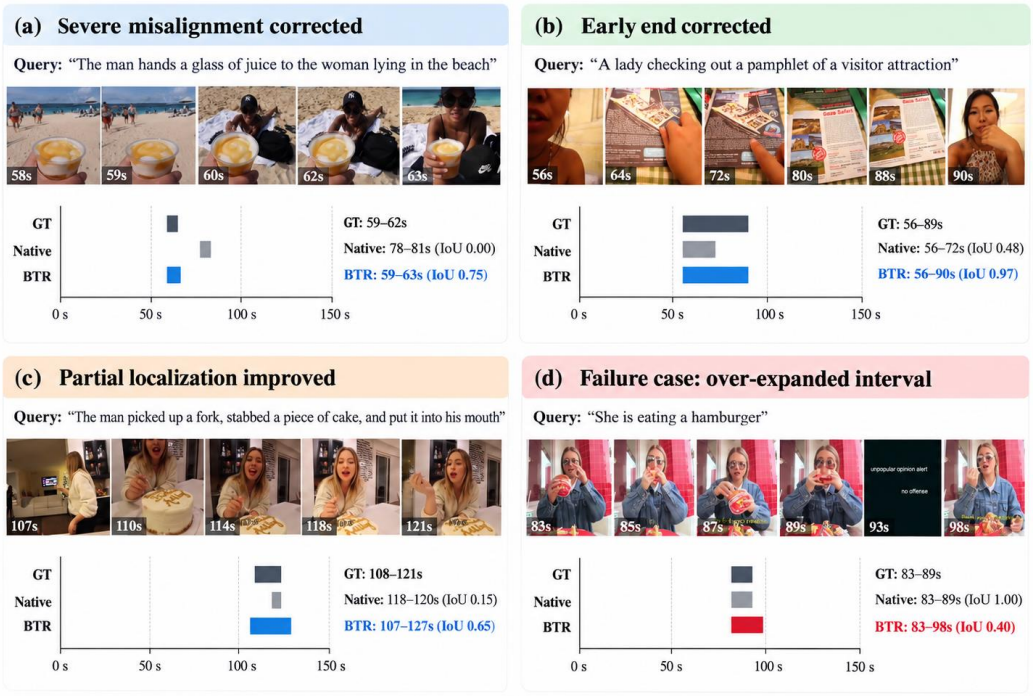

Figure 9: **Qualitative examples of temporal boundary revision.** BTR corrects severe temporal misalignment, incomplete event coverage, and partial localization errors, while the failure case shows that broad evidence reuse may expand an already accurate interval.

C.3 Control Experiment Details

Table 2 in the main paper reports the control comparisons. We specify the matched procedures here.

A native recheck generates timestamps again without evidence reuse. For the TimeLens-SFT adaptation control, we use rank-5 LoRA on the Q/V projections of all 36 decoder layers, trained on the same supervised data used for the reported BTR comparison. This LoRA configuration has 2,396,160 trainable parameters, which is 8.86× the 270,336 trainable parameters of BTR. It is therefore a same-data adaptation control with a substantially larger trainable-parameter budget rather than a parameter-matched baseline. The random temporal revisit control keeps the trained BTR adapter and residual magnitude unchanged but randomly permutes the temporal assignment of residual updates. This control tests whether the temporal correspondence between revisited evidence and visual states is important.

For TimeLens-8B, the native recheck and HTR use the same native draft, second-pass instruction, and greedy decoding settings. Only HTR enables the learned revisit module in the second pass. These controls test the complete systems; the LoRA comparison does not match trainable-parameter counts.

C.4 Post-Revisit Boundary Flow

The main paper shows that Temporal Revisit changes boundary-generation pathways after intervention. This section provides additional analysis of how revisited evidence contributes to timestamp generation.

Post-BTR added-support analysis. We compare the native TimeLens-SFT model and its BTR variant on the same BTR-generated timestamp sequence. This controlled evaluation measures how different attention routes support the additional boundary-token probability introduced after evidence reuse.

Table 13: **Post-BTR added-support knockout on TimeLens-SFT.** The values measure token-level support changes rather than task-level mIoU differences.

Knockout	Remaining support G_k	Eliminated support E_k
None	+0.10915	–
$\mathcal{V} \to \mathcal{P}$	+0.11603	-0.00688
$\mathcal{P} \to \mathcal{Y}$	+0.11502	-0.00588
$\mathcal{V} \to \mathcal{Y}$	+0.01159	+0.09756

Removing the visual-to-boundary pathway accounts for most of the additional token-level support introduced by BTR under this evaluation protocol, whereas blocking the tested visual-to-prompt or prompt-to-boundary pathways has smaller effects. This pattern is consistent with the design of Temporal Revisit, where intermediate visual evidence is returned to later visual states before boundary decoding.

HTR pathway analysis. Figure 10 provides additional Boundary Flow analysis for HTR on the TimeLens-8B host. Since HTR introduces a native interval hypothesis and a second refinement pass, these curves describe the complete refinement procedure rather than isolating the learned residual alone.

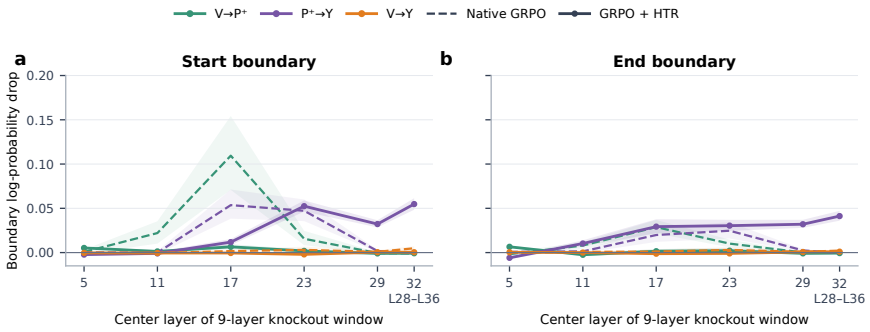

Figure 10: **Boundary Flow before and after HTR on TimeLens-8B.** Attention knockouts are applied to tested information pathways for paired queries. Each system is teacher-forced on its own generated timestamps. HTR includes a native draft and a second-pass refinement prompt.

C.5 Layer Sensitivity Analysis of Temporal Revisit

Temporal Revisit requires selecting intermediate layers to read and inject visual evidence. To examine the sensitivity of this choice, we perform a fixed-weight layer sweep on the TimeLens-SFT BTR model.

A single trained L21→L34 adapter is reused throughout, while only the inference-time readout or injection location is changed. Therefore, these experiments characterize layer sensitivity rather than independently optimized layer configurations.

Table 14 reports representative readout/injection pairs. All tested configurations improve over the native host, indicating that the benefit of Temporal Revisit does not depend on one exact layer choice. The tested configurations vary in performance under the same trained adapter; this comparison does not isolate representation quality from adapter compatibility. In the table, Δ compares against the native host, whereas the 95% confidence intervals compare against the L21→L34 configuration.

The highest fixed-weight diagnostic score is obtained by moving the injection location from L34 to L30 while keeping the L21 readout. However, the paired bootstrap interval includes zero, and the original L21→L34 configuration achieves slightly higher R@0.7. We therefore retain L21→L34 as the principal configuration and interpret this analysis as a sensitivity study rather than a layer selection procedure.

Table 14: **Layer sensitivity analysis of Temporal Revisit.** A fixed trained L21→L34 adapter is reused while changing inference-time readout and injection layers. Representative pairs are shown; the sweep is a sensitivity analysis rather than a layer optimization.

Readout	Injection	mIoU	Δ	95% CI	R@0.7
L16	L30	59.99	+2.94	[-1.72,-0.24]	48.48
L21	L30	61.62	+4.58	[-0.14,+1.47]	50.03
L21	L32	61.14	+4.10	[-0.56,+0.94]	50.03
L21	L34	60.96	+3.92	–	50.49
L24	L30	60.44	+3.40	[-1.22,+0.19]	49.06
L27	L34	59.34	+2.30	[-2.32,-0.93]	47.83

D Implementation Details

This section records the model settings and evaluation protocols used for Boundary Flow Tracing, representation probing, and Temporal Revisit. The shared temporal memory and visual-state injection are defined in Eqs. 4 and 5; we specify the remaining routing, training, and evaluation details below.

D.1 Models, Datasets, and Input Protocols

The main mechanism experiments compare matched TimeLens-SFT and TimeLens-8B checkpoints built on Qwen3-VL-8B-Instruct. Additional grounding evaluations use Qwen3-VL-8B-Instruct and MiMo-VL-7B-RL as frozen hosts. All reported evaluations use the same timestamp output format and scoring protocol. Videos are sampled at 2 frames per second, with at most 448 frames per video.

Grounding is evaluated on the QVHighlights-TimeLens, Charades-TimeLens, and ActivityNet-TimeLens subsets. We report mean IoU and recall at IoU thresholds of 0.3, 0.5, 0.7, and 0.9.

Training data and evaluation isolation. The 30,000 query–video pairs used to train Temporal Revisit are selected from the training portion of TimeLens-100K (Zhang et al., 2026). The selected pairs comprise 14,733 from CosMoCap, 7,144 from InternVid-VTime, 4,523 from DiDeMo, 2,339 from QueryD, and 1,261 from the HiREST components. They cover 15,082 videos, with at most four queries per video.

We canonicalize video identifiers before splitting the source data or selecting training pairs. Clips from the same original QVHighlights video share a parent-video identifier and remain in the same split. Video identifiers appearing in any of the three evaluation sets are excluded from the training pool. The resulting training selection has no canonical-video-ID overlap with these evaluation sets. This audit concerns annotation and parent-video identities; it does not include a visual-content near-duplicate search.

D.2 Boundary Flow Tracing and Probe Settings

Boundary Flow Tracing measures how masking selected attention edges changes the log-probability of each model's own native timestamp tokens. We first record the native output and then teacher-force that same token sequence with and without the intervention. Start and end effects are computed separately. These measurements quantify sensitivity to the tested route, not a decomposition of all information contributing to the prediction.

Representation probes are fitted to frozen hidden states independently at each decoder layer and evaluated on video-disjoint splits. For the query-conditioned visual probe, visual tokens are grouped into temporal bins. Restricted readouts score candidate boundary locations using visual representations, query information, and temporal features. Probe scores measure what can be recovered from the states, not whether the native generator uses every recoverable signal.

D.3 Temporal Revisit Architecture, Training, and Inference

Temporal Revisit reads visual states after layer ℓ_r and pools tokens within each temporal bin as defined in Eq. 4. The resulting temporal memory is transformed into residuals and returned to visual

tokens after layer ℓ_i according to Eq. 5. The vision encoder, language decoder, and generation head remain frozen; only the revisit modules are trained. Table 15 summarizes the principal TimeLens configurations.

Table 15: **Principal Temporal Revisit configurations on TimeLens.** Layer numbers refer to outputs after the corresponding decoder layers. BTR generates an interval in one pass; HTR first generates a native interval and then performs a refinement pass.

Strategy	Readout	Injection	Passes	Conditioning
BTR	L21	L34	1	Query, without a native draft
HTR	L16	L34	2	Query and native interval

The readout layers lie in the intermediate region where the mechanism analysis identifies recoverable temporal evidence. The configurations are fixed across the three reported TimeLens benchmarks. For the TimeLens-SFT BTR configuration, the residual adapter uses a bottleneck width of 32 and has 270,336 trainable parameters. This bottleneck width is an internal BTR adapter dimension and is not a LoRA rank. Appendix C.5 examines BTR layer placement by reusing a trained adapter at alternative readout and injection sites. That experiment measures fixed-weight sensitivity rather than comparing independently trained configurations.

Hypothesis-guided routing. In the first pass, the frozen host generates a native interval $[s_0, e_0]$. During refinement, HTR constructs the four queries in Eq. 7: local and global queries for each boundary. The local queries combine the native interval with prompt context, whereas the global queries use the original query representation.

Let $u_t \in [0, 1]$ denote the normalized temporal position of bin t, and let $\phi_T(u_t)$ be its temporal encoding. HTR derives keys and values from the encoded temporal memory. For boundary $b \in \{s, e\}$ and branch $c \in \{\mathrm{loc}, \mathrm{glob}\}$, routing is computed as

$$\alpha_{b,c,t} = \mathrm{softmax}_t\left(\frac{\mathbf{q}_{b,c}^{\top}\mathbf{k}_{b,c,t}}{\sqrt{d}}\right), \qquad \mathbf{r}_{b,c,t} = T\alpha_{b,c,t}\mathbf{v}_{b,c,t}, \tag{8}$$

where d is the key dimension. Multiplication by T preserves the value scale under uniform attention. Local and global features retain their temporal-bin indices and enter separate start and end adapters:

$$\boldsymbol{\delta}_{b,t} = f_b([\mathbf{r}_{b,\mathrm{loc},t}; \mathbf{r}_{b,\mathrm{glob},t}]), \qquad \boldsymbol{\delta}_t^{\mathrm{HTR}} = \boldsymbol{\delta}_{s,t} + \boldsymbol{\delta}_{e,t}. \tag{9}$$

The residual adapters have zero-initialized output projections. Their combined residual updates only visual-token states; the frozen decoder generates the final timestamps.

Timestamp supervision. Let $\mathbf{y}^\star$ be the ground-truth timestamp sequence and $\mathcal{J}_{\mathrm{time}}$ its timestamp-token positions. Both BTR and HTR use teacher-forced generation loss:

$$\mathcal{L}_{\mathrm{gen}} = -\frac{1}{|\mathcal{J}_{\mathrm{time}}|}\sum_{j\in\mathcal{J}_{\mathrm{time}}} \log p_{\Theta,\varphi}\big(y_j^\star \mid x, y_{<j}^\star\big), \tag{10}$$

where Θ denotes the frozen host and φ the trainable revisit parameters. BTR applies this loss in its single generation pass. HTR applies it to the refined output in the second pass. Its first-pass native interval serves only as a routing hypothesis and is never replaced by the ground-truth interval.

Boundary-routing supervision. HTR also supervises where its local and global branches retrieve evidence. Let D be the video duration and $\bar{y}_b^\star = y_b^\star/D$ the normalized ground-truth location of boundary $b \in \{s, e\}$. We define a normalized Gaussian target over temporal bins:

$$g_b^\star(t) = \frac{\exp\big[-(u_t - \bar{y}_b^\star)^2/(2\sigma^2)\big]}{\sum_{k=1}^{T}\exp[-(u_k - \bar{y}_b^\star)^2/(2\sigma^2)]}, \qquad \sigma = \max(0.04,\ 1.5\ \mathrm{median}_{k=2,\ldots,T}\,|u_k - u_{k-1}|). \tag{11}$$

For a single temporal bin, we set $\sigma = 0.04$. The two branches associated with each boundary share the same target:

$$\mathcal{L}_{\mathrm{route}} = -\frac{1}{4}\sum_{b\in\{s,e\}}\sum_{c\in\{\mathrm{loc},\mathrm{glob}\}}\sum_{t=1}^{T} g_b^\star(t)\log\alpha_{b,c,t}. \tag{12}$$

Preserving accurate native endpoints. For each training example, let $\hat{y}_b^{(0)}$ be the endpoint generated in the native first pass and $y_b^\star$ the corresponding ground-truth endpoint, both measured in the original video time coordinate. An endpoint is marked as requiring no revision when $|\hat{y}_b^{(0)} - y_b^\star| \le 0.5\,\mathrm{s}$. Let $\mathcal{J}_b$ contain all timestamp-token positions of endpoint $b \in \{s, e\}$. The positions used for preservation are

$$\mathcal{J}_{\mathrm{keep}} = \bigcup_{\substack{b \in \{s,e\}: \\ |\hat{y}_b^{(0)} - y_b^\star| \le 0.5\,\mathrm{s}}} \mathcal{J}_b. \tag{13}$$

This label is constructed during training only. Under the same second-pass prompt, we compare the output distribution with the HTR residual enabled against the distribution with that residual disabled:

$$\mathcal{L}_{\mathrm{pres}} = \frac{1}{|\mathcal{J}_{\mathrm{keep}}|} \sum_{j \in \mathcal{J}_{\mathrm{keep}}} D_{\mathrm{KL}}\Big(p^{(2)}_{\Theta,0}(\cdot \mid x, y^\star_{<j}) \,\Big\|\, p^{(2)}_{\Theta,\varphi}(\cdot \mid x, y^\star_{<j}) \Big). \tag{14}$$

Here, the superscript (2) denotes the refinement pass. We set $\mathcal{L}_{\mathrm{pres}} = 0$ when $\mathcal{J}_{\mathrm{keep}}$ is empty. The complete objectives are

$$\mathcal{L}_{\mathrm{BTR}} = \mathcal{L}_{\mathrm{gen}}, \qquad \mathcal{L}_{\mathrm{HTR}} = \mathcal{L}_{\mathrm{gen}} + 0.2\,\mathcal{L}_{\mathrm{pres}} + 0.05\,\mathcal{L}_{\mathrm{route}}. \tag{15}$$

Routing targets and ground-truth timestamp tokens are used only during training.

Optimization and inference. For the verified TimeLens-SFT BTR run reported in the main results, we train on 30,000 query–video pairs for one epoch with global batch size 24, giving 1,250 optimizer steps. We use AdamW with weight decay 10^{-4} and a peak learning rate of 10^{-4}. The learning rate warms up for 62 steps (approximately 5% of training) and then follows cosine decay. The reported checkpoint is the completed step-1,250 checkpoint from seed 42.

At inference, BTR generates the interval in one pass. HTR first obtains the frozen host's native interval and then performs one refinement pass. Neither strategy has access to ground-truth boundaries at inference, and neither introduces a separate timestamp prediction head.

Parameter count and temporal resolution. BTR has 270,336 trainable parameters, and HTR has 3,312,420; the host MLLM remains frozen in both cases. The number of pooled temporal bins T depends on the sampled video rather than being a fixed hyperparameter. Across the complete QVHighlights-TimeLens, Charades-TimeLens, and ActivityNet-TimeLens evaluations, the median values of T are 150, 30, and 140, respectively. The corresponding observed ranges are 120–150, 7–72, and 36–224. BTR uses one generation pass, whereas HTR generates a native draft and then performs one refinement pass using the frozen host.

D.4 Evaluation Protocol

Native and revisited predictions are evaluated on identical query–video instances. The final interval is always generated by the host language decoder; neither BTR nor HTR introduces a separate timestamp head. IoU is computed against the corresponding ground-truth interval in the original video time coordinate.

When reported, paired confidence intervals use 2,000 video-cluster bootstrap resamples unless a different resampling unit is stated for a specific diagnostic. Qualitative examples are visualized in the original video time coordinate.