THREADNIAH: REASONING OVER RELATIONAL NEEDLES IN LONG VIDEO HAYSTACKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Multimodal large language models can increasingly process hour-long videos, where relevant evidence may be sparse and distributed over time. Effective long-video understanding therefore requires models to locate relevant evidence, link related evidence across the video, and combine multiple pieces of evidence. Building on the controlled setting of Video Needle-in-a-Haystack (NIAH), we introduce ThreadNIAH to systematically evaluate these capabilities. ThreadNIAH organizes five task types into three levels that progressively evaluate locating, linking, and combining evidence: Single-Needle Retrieval (Details), Multi-Needle Threading (Tracking and Comparison), and Multi-Needle Synthesis (Counting and Logic). To instantiate these task types in a controlled and scalable manner, we design 29 task templates that specify the evidence-bearing needles, their relations, and the questions defined over them. Since a correct final answer does not necessarily imply correct intermediate steps, each Main QA is paired with diagnostic Sub-QAs that probe its supporting evidence and intermediate steps. Together, they form a QA chain, the basic evaluation unit of ThreadNIAH. We use Main Acc, Sub Acc, and All Pass Rate (APR) to measure final-task, intermediate-step, and all-question chain accuracy, respectively. The resulting benchmark contains 870 QA chains, comprising 870 Main QAs and 2,580 Sub-QAs. Across ten video MLLMs and five thinking variants, most configurations perform substantially better on single-needle retrieval than on multi-needle tasks. Moreover, thinking does not ensure consistent success across QA chains: even Gemini 3.6 Flash with thinking achieves 67.7% Main Acc but only 49.6% APR. Similar difficulty patterns on real-world long-video questions further support the relevance of ThreadNIAH to real-world long-video understanding.

1 INTRODUCTION

As multimodal large language models (MLLMs) rapidly evolve toward increasingly long multimodal contexts (Google DeepMind, 2026b; Alibaba Cloud, 2026), evaluating their long-video understanding capabilities has become increasingly important. Effective long-video understanding requires not only locating key evidence but also reasoning over sparse evidence scattered throughout the video. We decompose this capability into three progressively demanding stages: **locating** key evidence, **linking** related evidence across locations within the video, and **combining** information from multiple pieces of evidence. General long-video benchmarks (Fu et al., 2025; Wu et al., 2024), however, primarily evaluate final-answer accuracy and cannot assess these stages separately. Recent clue-grounding benchmarks (Chen et al., 2024) additionally annotate supporting video segments for answers, but such annotation requires substantial manual effort, limiting scalability as evidence configurations grow more complex.

Needle-in-a-Haystack (NIAH) has emerged as a standard paradigm for evaluating long-context understanding, offering fine-grained control over evidence placement and scalable data synthesis (Kamradt, 2023). In the text domain, NIAH benchmarks have evolved from single-needle retrieval to multi-needle retrieval and reasoning (Hsieh et al., 2024; Kuratov et al., 2024). Extending this paradigm to video, however, is non-trivial due to the richer and more continuous nature of visual content. Existing video NIAH benchmarks explore different aspects: V-NIAH (Zhang et al., 2024) evaluates image retrieval, VNBench (Zhao et al., 2025) extends to ordering and counting, VideoChat-Flash (Li et al., 2026) introduces linked clues for multi-hop search, and VideoRoPE

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.

(Wei et al., 2025) further considers visually similar distractors. Together, these efforts extend video NIAH beyond simple retrieval, yet none systematically evaluates the three-stage process of locating, linking, and combining evidence. Moreover, final-answer evaluation alone cannot diagnose *at which stage* a model fails.

To address these limitations, we introduce **ThreadNIAH**, a benchmark that uses controlled relations among needles to evaluate reasoning over multiple needles in long videos. It organizes evidence use into three levels (L1: locating, L2: linking, L3: combining) and five task types: *Single-Needle Retrieval* of L1 contains *Details*; *Multi-Needle Threading* of L2 contains *Tracking* and *Comparison*; and *Multi-Needle Synthesis* of L3 contains *Counting* and *Logic*. At the construction level, 29 templates instantiate these task types by specifying the roles of target, reference, and decoy needles, how they are distributed and related within the video, and how those relations determine the answer. Models must therefore identify the relevant needles and use their relations to track, compare, count, or combine evidence before deriving an answer.

The controlled construction of NIAH question-answer pairs also allows us to examine the evidence and intermediate operations underlying each complete task. Starting from a *Main QA*, we construct progressively related *Sub-QA* that probe its supporting evidence and operations; together they form a *QA chain*, the basic evaluation unit of ThreadNIAH. All questions in a chain use the same video, but each is presented, answered, and scored independently (earlier responses are not passed to later questions), so that each metric reflects an isolated capability rather than accumulated context. We report *Main-Acc* for the Main QA, *Sub-Acc* for the associated Sub-QA, and *All Pass Rate* (APR) for chains in which every question is answered correctly, separating final-task performance from diagnostic-question performance and complete-chain consistency.

We benchmark frontier MLLMs and specialized long-video models on ThreadNIAH. Reasoning over multiple needles remains difficult: most configurations achieve higher Main-Acc on Details than on Tracking, Comparison, Counting and Logic. QA-chain metrics further reveal that correct Main QA answers do not guarantee correct answers to all associated diagnostic questions: even with thinking enabled, Gemini 3.6 Flash achieves 67.7% Main-Acc but only 49.6% APR, an 18.1-point gap between final-task success and complete-chain success. Thinking improves overall performance for most models but does not eliminate this gap. Beyond synthetic haystacks, we additionally construct a companion evaluation over natural long videos that mirrors the five task types; the two settings yield consistent task-type difficulty rankings, indicating that ThreadNIAH reflects performance patterns in natural long-video QA. Taken together, ThreadNIAH and these analysis provide a controlled basis for diagnosing the strengths and limitations of long-video models and offer practical guidance for future model development and training.

Our main contributions are: 1) We introduce **ThreadNIAH**, which exploits the controllability of NIAH to test reasoning over multiple needles in long videos. We design three levels and five task types to systematically evaluate the progressively demanding capabilities of *locating*, *linking*, and *combining* evidence. 2) We introduce **QA chains**, which pair each complete task with diagnostic questions on its supporting evidence and intermediate operations, together with Main-Acc, Sub-Acc, and APR to separate final-task, diagnostic, and complete-chain performance. 3) Through systematic evaluation of frontier MLLMs and specialized long-video models, we show that multi-needle reasoning remains the primary bottleneck, that thinking does not close the gap between final-answer and complete-chain success, and that task-type difficulty rankings transfer from synthetic haystacks to natural long videos.

2 Related Work

Long-Video MLLMs and General Video Benchmarks: Long-video MLLMs increasingly rely on compression, memory, and active search to identify and use relevant evidence. Recent methods explore query-aware compression and memory (Fei et al., 2026; Chen et al., 2026; Yashima et al., 2026), as well as iterative clue discovery and video navigation (Zeng et al., 2026; Gao et al., 2026; Qiu et al., 2026), highlighting the importance of evidence handling in long-video understanding. Meanwhile, long-video benchmarks evaluate diverse domains and capabilities. Video-MME covers varying domains, durations, and modalities (Fu et al., 2025); MLVU and ALLVB include narrative, temporal, and counting tasks (Zhou et al., 2025; Tan et al., 2025); and LongVideoBench evaluates fine-grained retrieval and reasoning over video and subtitles (Wu et al., 2024).

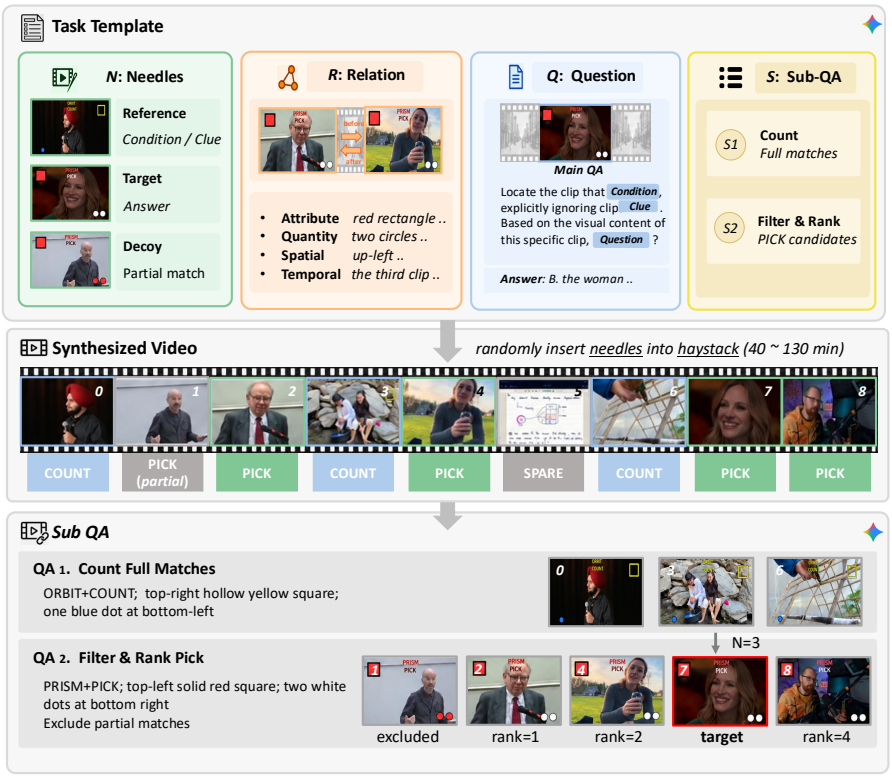

Figure 1: Task construction and an example in ThreadNIAH. Top: the components of a task template. Middle: an example synthesized video containing target, reference and decoy needles. Bottom: Sub-questions progressively probe reference-needle counting, candidate filtering, and ranking. Questions within each QA chain are answered independently.

Long-Video NIAH Benchmarks: Video NIAH benchmarks place sparse evidence within longer video contexts. V-NIAH tests retrieval of an inserted image as context length and needle position vary, while NIAVH also tests text and video needles (Zhang et al., 2024; Wang et al., 2025). VNBench evaluates retrieval, ordering, and counting as separate tasks (Zhao et al., 2025), and VideoRoPE tests retrieval among visually similar distractors (Wei et al., 2025). VideoChat-Flash requires models to follow linked clues across clips to find a target (Li et al., 2026). Causal2Needles asks models to connect cause and effect events at different video locations (Li et al., 2025).

These benchmarks extend video NIAH beyond retrieval, but each focuses on a limited set of relations or operations across evidences. Their final-answer evaluation alone also makes it difficult to diagnose errors in supporting evidence and intermediate steps.

3 The ThreadNIAH Benchmark

ThreadNIAH evaluates three core capabilities in long-video understanding: *locating*, *linking*, and *combining* evidence distributed across long videos. Each instance is built from a *task template* that specifies which clips carry evidence (*Needle*), how they relate (*Relation*), how the answer can be derived from the evidence (*Question*), and which supporting evidence and intermediate operations the ordered Sub-QA steps probe (*Sub-QA*). For each instance, the resulting main question and sub-questions form a QA chain. Building on these templates, we introduce two key designs: a task taxonomy covering different levels of evidence use, and QA chains for diagnosing performance at intermediate steps. Figure 1 illustrates the task-template components and an example synthesized video with its associated QA chain.

3.1 TASK TAXONOMY

We organize five task types into three levels of increasing evidence complexity: *L1. Single-Needle Retrieval* (Details), *L2. Multi-Needle Threading* (Tracking and Comparison), and *L3. Multi-Needle Synthesis* (Counting and Logic), corresponding to the progressively demanding capabilities of *locating*, *linking*, and *combining* evidence, respectively.

L1 - Details tests whether a model can locate a described clip in a long video and answer questions about its content. The model must identify the target from the description and read the requested detail from that clip.

L2 - Tracking tests whether a model can use temporal relations across clips to determine the target. The model must combine the stated visual conditions with relations such as occurrence before or after a reference, position within a time window, or order among eligible clips to select the relevant evidence.

L2 - Comparison tests whether a model can reason about visual similarities and differences across clips. The model must compare corresponding attributes or relations in different clips and use the resulting matches or differences to answer the question or select the target.

L3 - Counting tests whether a model can determine quantities by aggregating evidence across clips under the stated conditions. The model must decide which candidates qualify and combine their counts, using the resulting quantity or a comparison between quantities either as the answer or to guide subsequent target selection.

L3 - Logic tests multi-step reasoning over evidence from multiple clips, where each step depends on the results of previous steps. Intermediate results, such as a retrieved clue or a comparison of group frequencies, determine which evidence to select next and how to use it.

3.2 QA CHAIN AS THE EVALUATION UNIT

A correct final answer does not necessarily indicate that a model correctly understands the supporting evidence and intermediate steps. Our task templates explicitly specify the evidence and relations involved in a task, allowing us to construct sub-questions that vary in the evidence and operations they require. We organize the main question and its sub-questions into a QA chain for fine-grained evaluation.

For each instance, the *Main QA* defines the complete task, whose solution requires identifying relevant evidence and performing the necessary reasoning steps. Intermediate evidence may include clip attributes, visual clues, or temporal positions, while reasoning steps may involve comparison, temporal selection, or counting. We construct *Sub-QAs* to probe progressive steps toward solving the Main QA, from identifying local evidence to performing broader evidence integration and reasoning. The bottom panel of Figure 1 illustrates this progression through identifying labeled clips, filtering and ranking candidates, and examining the target needle's content. Together, the Sub-QA and Main QA form a *QA Chain*, $\mathcal{C} = (QA_1, \ldots, QA_L)$, where QA_L is the Main QA. Each chain is the basic evaluation unit: all questions share the same video and are answered independently without access to previous responses.

3.3 METRIC

To examine whether models maintain a consistent understanding across related questions about the same video, we evaluate both individual question performance and joint success within each QA chain. We report **Main Acc** for Main QA, **Sub Acc** for the associated Sub-QA, and **All Pass Rate** for chains in which every question is answered correctly.

Specifically, for task type k, let N_k be the number of QA chains, L_{ki} the length of chain i, and $c_{kij} \in \{0, 1\}$ indicate whether its j-th question is answered correctly. With the Main QA placed last in each chain, we compute

$$\mathrm{MainAcc}_k = \frac{1}{N_k} \sum_{i=1}^{N_k} c_{kiL_{ki}}, \qquad \mathrm{SubAcc}_k = \frac{1}{N_k} \sum_{i=1}^{N_k} \frac{\sum_{j=1}^{L_{ki}-1} c_{kij}}{L_{ki} - 1},$$

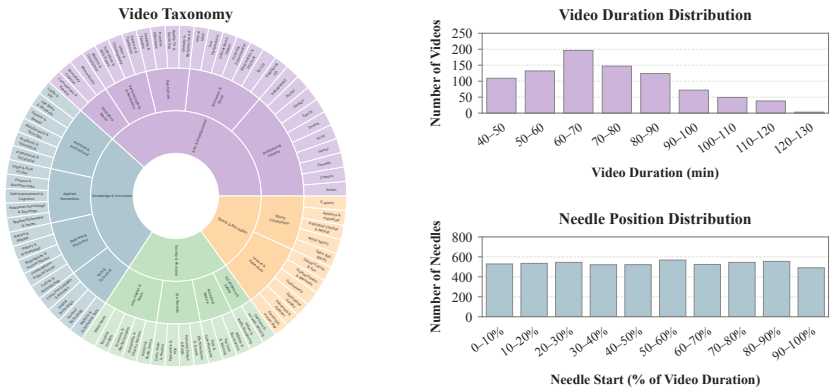

Figure 2: Haystack taxonomy and dataset statistics. Left: three-level category distribution of the collected haystack videos, covering four domains, 15 content groups, and 73 fine-grained categories. Top right: durations of 870 synthesized evaluation videos in 10-minute bins. Bottom right: start times of 5,335 needle insertions, normalized by synthesized video duration, in bins of width 0.1.

$$\text{AllPassRate}_k = \frac{1}{N_k} \sum_{i=1}^{N_k} \prod_{j=1}^{L_{ki}} c_{kij}.$$

Sub Acc first averages correctness over the Sub-QA within each chain and then over chains. For each metric M, we report the overall score as $M = \frac{1}{5}\sum_{k=1}^{5} M_k$, giving equal weight to the five task types.

3.4 Construction

3.4.1 Haystack Video Collection

Existing video NIAH benchmarks often vary context length and needle position over a limited set of source videos (Zhang et al., 2024; Li et al., 2026). While well controlled, such designs may not fully capture model performance across diverse video domains. To broaden haystack diversity, we collect videos from YouTube[1] and Bilibili[2], drawing on 199 source videos across four domains. Figure 2 shows the video taxonomy (left) and the duration distribution of the synthesized evaluation videos (top right). The 870 synthesized videos last 40–130 minutes, with a mean length of 72.2 minutes.

3.4.2 Task Templates

Each of the 29 task templates is represented as $\mathcal{T} = (\mathcal{N}, \mathcal{R}, \mathcal{Q}, \mathcal{S})$. $\mathcal{N}$ specifies the roles of the inserted clips (*needles*), including references, targets, and decoys. $\mathcal{R}$ defines *relations* among these roles through attribute matches and differences, spatial configurations, temporal order, and quantity-based conditions. These relations are expressed through embedded visual clues—code words, shapes, colors, fill modes, marker counts, and on-screen positions—and, in some tasks, natural visual attributes. A template may combine conditions or use an intermediate result to determine which clips to examine next. $\mathcal{Q}$ specifies the *Main QA question*, with slots for instance-specific clues and conditions. These clues and conditions guide the selection of the target needle, while the Main QA primarily asks about its natural visual content. $\mathcal{S} = (s_1, \ldots, s_m)$ specifies an ordered sequence of *Sub-QA steps*. Each step identifies the supporting evidence or intermediate operation that its question probes, while its concrete question and answer options are instantiated for each instance in the next step. The top panels of Figure 1 illustrate $\mathcal{N}$, $\mathcal{R}$, $\mathcal{Q}$, and $\mathcal{S}$, while the bottom panel shows an example progression specified by $\mathcal{S}$. Appendix A illustrates the construction of a QA chain from a single template.

[1] https://www.youtube.com/
[2] https://www.bilibili.com/

Table 1: Comparison of video NIAH benchmarks in scale, capability coverage, cross-clip reasoning, and cross-QA consistency evaluation.

Benchmark	Video Length	# QA	Capabilities			Cross-clip Reasoning	Cross-QA Consistency
			Locate	Link	Combine		
V-NIAH[a]	≤50 min	5 seeds	✓	×	×	×	×
NIAVH[b]	1–320 s	NR	✓	×	×	×	×
VNBench[c]	10–180 s	1,350	✓	✓	✓	✓	×
V-NIAH-D[d]	≤25 min	5 seeds	✓	×	×	×	×
MH-NIAH-Video[e]	NR	NR	✓	✓	×	✓	×
Causal2Needles[f]	438 s (mean)	4,100	✓	✓	×	✓	×
ThreadNIAH	**40–130 min**	**3,450**	✓	✓	✓	✓	✓

[a]Zhang et al. (2024); [b]Wang et al. (2025); [c]Zhao et al. (2025); [d]Wei et al. (2025); [e]Li et al. (2026); [f]Li et al. (2025).
Locate: retrieve clips and local details. Link: use temporal, visual, or causal relations between clips. Combine: aggregate across clips or compose different reasoning operations. NR: duration or QA total not reported. V-NIAH and V-NIAH-D reuse five question seeds across insertion settings; Cross-clip reasoning uses information from multiple clips. Cross-QA consistency examines answer patterns across independently evaluated, related questions.

3.4.3 From Templates to Instances

To create an instance, we assign source clips to the roles in $\mathcal{N}$ and select clue values from predefined sets according to $\mathcal{R}$, coordinating their assignments across clips to establish the required matches, differences, and links. Figure 1 illustrates this process with a concrete example. The number of reference needles tagged as `COUNT` determines the ranking index used to select the target from those tagged as `PICK`.

We instantiate $\mathcal{Q}$ with the assigned clues and conditions. Gemini 3.1 Pro generates a question about the target clip's natural visual content, four options, and a reference answer from frames sampled before the clues are overlaid. We combine this question with the template's clue-based selection instructions to form the Main QA. Given the Main QA, assigned clues, planned clip order, and scene descriptions, Gemini 3.1 Pro then generates the questions, options, and answers for the predefined Sub-QA steps in $\mathcal{S}$. We burn the assigned clues into the source clips and insert them into a haystack, preserving the specified temporal relations and distributing insertion positions as evenly as possible to reduce systematic position bias. The Sub-QA, followed by the Main QA, form a QA chain for the synthesized video, following the progression described in §3.2.

3.4.4 Quality Control

Every instance must pass two automatic checks using GPT-5.2; instances that fail either check are regenerated or rejected. For *visual reproducibility*, the model answers the question from sampled frames of the relevant needles without seeing the reference answer; the instance passes only if its answer matches. To check for *textual shortcuts*, we ask the model the same question twice in separate calls using only the question and options, and reject it if both answers match the reference answer.

3.5 Dataset Statistics

Table 1 compares ThreadNIAH with prior video NIAH benchmarks, highlighting ou broader task coverage and diagnostic evaluation. ThreadNIAH contains 870 QA chains constructed from 29 templates across five task types, comprising 870 Main QAs and 2,580 Sub-QAs, with an average chain length of 3.97. Detailed statistics are provided in Appendix A.1.

Figure 2 presents the haystack taxonomy and temporal distributions of the evaluation set. We draw haystacks from 199 source videos across four domains. The 870 synthesized videos last 40–130 minutes, with a mean of 72.2 minutes. Needle starts span the full normalized timeline in all five task types. Each video contains 3–5 needles in Details, 3–8 across L2, and 4–10 across L3.

Table 2: Main Acc, Sub Acc, and All Pass Rate (APR) by task type and equal-weight overall mean (%). APR is shaded blue. Bold marks the highest nonzero model score in each group.

Model	Frames	Metric	Details	Tracking	Comp.	Counting	Logic	Overall
Human Baseline								
Human Expert	–	Main	92.9	92.3	85.7	85.7	90.9	89.5
		Sub	100.0	90.4	92.9	92.4	93.9	93.9
		APR	92.9	69.2	71.4	57.1	72.7	72.7
Proprietary Models (Thinking)								
Gemini 3.6 Flash	1 fps	Main	**76.1**	**68.5**	**80.9**	**59.0**	**53.8**	**67.7**
		Sub	**88.0**	**83.9**	**91.6**	79.1	**90.0**	**86.5**
		APR	**67.4**	**44.4**	**64.7**	**26.2**	**45.3**	**49.6**
Qwen3.7-Plus	1 fps	Main	65.2	55.3	56.5	50.5	48.1	55.1
		Sub	84.0	77.8	84.4	**80.6**	89.8	83.3
		APR	55.4	25.0	29.4	12.6	35.7	31.6
Qwen3.7-Max	1 fps	Main	63.2	55.0	51.2	41.5	41.4	50.5
		Sub	86.5	77.1	82.7	79.9	89.3	83.1
		APR	53.5	24.2	27.9	10.6	29.7	29.2
GLM-5.3-Flash	Default	Main	29.5	28.1	29.1	23.5	23.1	26.7
		Sub	48.9	43.1	44.1	35.9	39.0	42.2
		APR	14.3	2.5	1.7	0.0	0.6	3.8
MiMo V2.6 Pro	1 fps	Main	27.6	27.4	19.8	16.0	21.2	22.4
		Sub	47.5	42.6	44.1	36.4	35.6	41.2
		APR	13.0	1.6	3.4	0.0	0.7	3.7
Proprietary Models								
Gemini 3.6 Flash	1 fps	Main	**70.8**	**54.3**	**52.5**	43.3	42.2	**52.6**
		Sub	**78.5**	**56.6**	52.8	44.1	**56.8**	57.8
		APR	**56.7**	**12.9**	10.8	**3.3**	**6.7**	**18.1**
Qwen3.7-Max	1 fps	Main	54.3	49.7	51.2	38.3	46.3	48.0
		Sub	76.4	53.7	60.6	51.1	53.5	59.1
		APR	43.5	9.7	**15.1**	2.1	6.0	15.3
Qwen3.7-Plus	1 fps	Main	57.8	49.3	48.2	**44.8**	**50.4**	50.1
		Sub	78.1	54.3	**62.8**	**52.3**	49.7	**59.4**
		APR	44.3	7.2	12.9	3.1	3.0	14.1
Doubao Seed 2.1 Pro	1 fps	Main	37.7	48.8	45.9	43.4	25.6	40.3
		Sub	70.3	53.5	53.3	47.6	47.8	54.5
		APR	27.4	6.4	8.1	1.2	2.2	9.1
GLM-5.3-Flash	Default	Main	35.7	33.0	34.5	33.9	31.6	33.7
		Sub	53.2	49.9	45.9	41.0	44.6	46.9
		APR	19.3	3.3	2.5	1.7	2.3	5.8
MiMo V2.6 Pro	1 fps	Main	31.6	25.5	27.4	25.4	26.8	27.3
		Sub	47.3	32.5	33.5	30.1	30.0	34.7
		APR	14.8	0.5	1.7	0.9	0.7	3.7
Open-Weight Models								
Qwen3.5-27B	768	Main	**49.2**	**40.0**	**36.7**	**44.2**	**31.7**	**40.3**
		Sub	**64.2**	**48.6**	**50.4**	**38.8**	**40.0**	**48.4**
		APR	**34.2**	**5.2**	**2.5**	0.0	**1.1**	**8.6**
Qwen3.5-9B	768	Main	34.2	35.2	30.0	30.8	25.0	31.0
		Sub	43.5	35.3	38.5	33.1	36.4	37.4
		APR	13.8	1.4	1.7	0.0	0.6	3.5
VideoChat-Flash-7B	512	Main	38.3	28.6	31.7	25.0	29.4	30.6
		Sub	45.2	27.9	30.8	28.1	28.4	32.1
		APR	17.1	0.0	0.0	0.0	0.6	3.5
LongVT-RFT	768	Main	34.5	26.9	30.4	28.7	22.8	28.6
		Sub	35.3	29.3	30.6	25.1	28.5	29.8
		APR	9.8	0.5	0.9	0.0	0.6	2.4

4 EXPERIMENTS

4.1 MODELS

We evaluate ten video MLLMs on ThreadNIAH: six proprietary models accessed through APIs and four open-weight models run locally. The six proprietary models are Gemini 3.6 Flash (Google DeepMind, 2026a), Qwen3.7-Max and Qwen3.7-Plus (Qwen Team, 2026), Doubao Seed 2.1 Pro (ByteDance, 2026), GLM-5.3-Flash (Zhipu AI, 2026), and MiMo V2.6 Pro (Xiaomi, 2026). We also evaluate Qwen3.5-9B and Qwen3.5-27B (Qwen Team, 2026) as well as VideoChat-Flash-Qwen2-7B (res448) (Li et al., 2026) and LongVT-RFT (Yang et al., 2025). We exclude Gemini 3.1 Pro, which was used for QA generation, from the baselines to avoid potential bias in favor of the

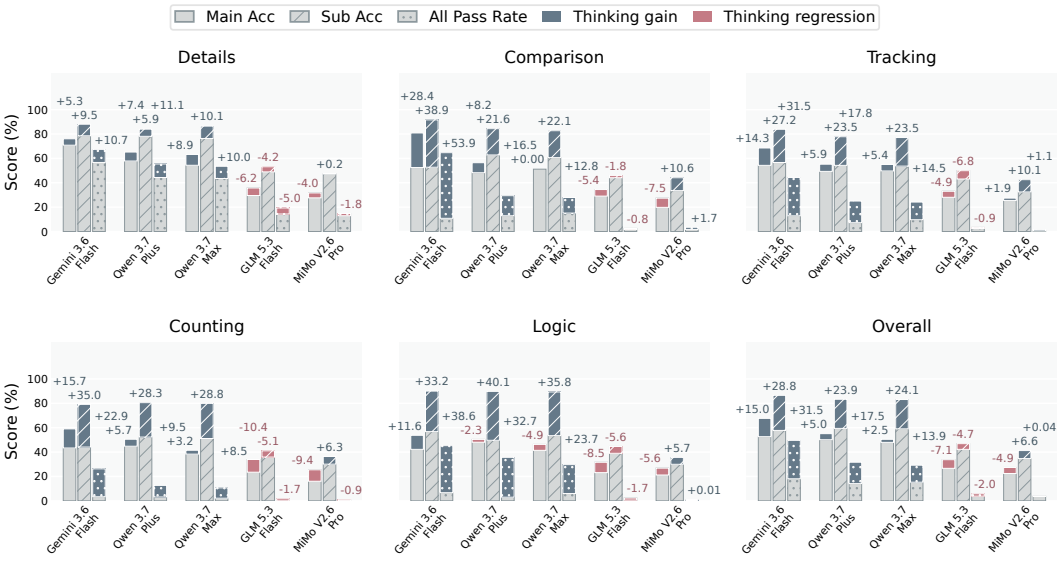

Figure 3: Effect of thinking on Main Acc, Sub Acc, and APR across five task types and overall. Bar patterns denote the metrics; dark and rose segments indicate gains and regressions from thinking, with labels showing changes in percentage points.

generating model (Xu et al., 2025). We sample videos at 1 fps where the interface allows it, and cap the frame count at each model's own budget—512 for VideoChat-Flash and 768 for LongVT and the two local Qwen3.5 models—with the frames spread evenly over the whole video. Other inference settings follow each model's recommendation.

4.2 Main Results

Table 2 reports performance across the five task types of ThreadNIAH. We summarize the main findings below.

1) Human experts outperform the evaluated models. Following prior video benchmarks (Pătrăucean et al., 2023; Mangalam et al., 2023), we assess human performance on a subset of 230 questions spanning all five task types. Human experts achieve 89.5% Main Acc, 93.9% Sub Acc, and 72.7% APR, compared with 67.7%, 86.5%, and 49.6%, respectively, for Gemini 3.6 Flash with thinking, the best-performing model on the full evaluation set.

2) Proprietary models lead the overall comparison. Gemini 3.6 Flash with thinking achieves the highest overall APR of 49.6%, compared with 8.6% for the strongest open-weight model, Qwen3.5-27B. It also leads in Main Acc and Sub Acc, scoring 67.7% and 86.5%, respectively.

3) Cross-clip reasoning remains difficult for most models. Most evaluated configurations achieve higher Main Acc on Details than on tasks requiring cross-clip reasoning. Qwen3.5-27B scores 49.2% on Details, compared with 40.0% on Tracking, 36.7% on Comparison, 44.2% on Counting, and 31.7% on Logic. Gemini 3.6 Flash at the minimal thinking level shows a similar pattern, scoring 70.8% on Details but 43.3% on Counting and 42.2% on Logic. These results suggest that models are substantially better at locating local details than linking and combining evidence across clips.

4) QA chains reveal failures hidden by final-answer accuracy. Models that answer the Main QA correctly can still fail on related questions about the same evidence and operations. Gemini 3.6 Flash with thinking achieves an overall Main Acc of 67.7%, but its APR is 49.6%, a gap of 18.1 percentage points. For Qwen3.7-Plus with thinking, the gap reaches 23.5 points, despite a Sub Acc of 83.3%. These gaps show that a correct Main QA can coexist with errors in related Sub-QAs, which are not captured by final-answer accuracy alone.

5) Thinking improves most models overall, with larger gains on cross-clip reasoning. Figure 3 shows overall APR gains for three of the five model pairs, led by Gemini 3.6 Flash (+31.5 points),

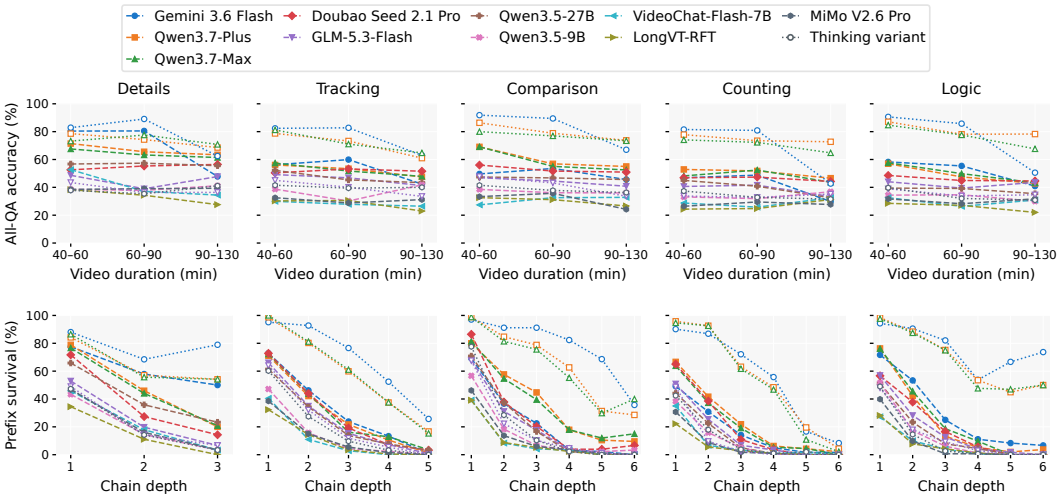

Figure 4: Performance across video duration and QA-chain depth for the five task types. Top: All-QA accuracy across three duration ranges, computed jointly over Main and Sub-QAs with equal weight per valid answer. Bottom: prefix survival at depth d, i.e., the percentage of eligible chains with all questions through depth d answered correctly.

Qwen3.7-Plus (+17.5), and Qwen3.7-Max (+13.9). The gains are generally larger on cross-clip tasks than on Details; for example, Gemini improves by 53.9 points on Comparison and 38.6 on Logic, versus 10.7 on Details. However, the benefit is model-dependent, with GLM declining slightly and MiMo showing little overall change.

4.3 Analysis

1) Effect of video duration. We divide the 870 evaluation videos into short (40–60 min), medium (60–90 min), and long (90–130 min) splits, and compute accuracy jointly over Main and Sub-QAs with equal weight per valid answer. As shown in the top row of Figure 4, all evaluated configurations degrade from short to long videos in at least two task types. Even Gemini 3.6 Flash with thinking drops from 81.5% to 42.7% on Counting and from 90.6% to 50.5% on Logic, with similar degradation observed for open-weight models. These results show that increasing video duration remains challenging even with thinking enabled.

2) Prefix survival across QA-chain depth. The bottom row of Figure 4 provides a depth-wise view of consecutive correctness. At depth d, prefix survival measures the percentage of eligible chains whose first d questions are all answered correctly. It is not the independent accuracy of question d: once an earlier question is answered incorrectly, that chain does not survive at any later depth.

Survival drops sharply with depth across all five task types, despite high accuracy on early diagnostic questions, which primarily probe evidence localization. For example, Gemini 3.6 Flash with thinking drops from 95.2% to 25.7% on Tracking (depth 1–5) and from 90.2% to 8.3% on Counting (depth 1–6), showing that success at early stages often fails to extend to an entirely correct chain.

5 Conclusion

We introduce ThreadNIAH, a comprehensive and diagnostic benchmark for long-video understanding that systematically evaluates the progressively demanding capabilities of *locating*, *linking*, and *combining* evidence. Our task-template design enables controlled and scalable construction across three levels and five task types, while QA chains provide fine-grained diagnosis of supporting evidence and intermediate steps. Experiments across frontier MLLMs and specialized long-video models reveal persistent challenges in multi-needle understanding, long-video processing, and maintaining correctness across QA chains. Overall, ThreadNIAH provides a low-cost and scalable framework for broader and more diagnostic evaluation of long-video understanding.

AI USE STATEMENT

We used generative AI tools during benchmark construction, as detailed in the paper. We additionally used them to assist with language editing and LaTeX formatting.

ETHICS STATEMENT

This work studies long-video multimodal reasoning and does not involve new human-subject data collection. ThreadNIAH is constructed from existing video sources and follows their original licenses and terms of use. We do not release personally identifiable information or attempt to identify individuals. Long-video assistants may produce incorrect, premature, or privacy-invasive outputs; ThreadNIAH is intended for research evaluation, not for deployment in surveillance or safety-critical applications without human oversight. We encourage users to audit the benchmark and models for bias, privacy, and misuse.

REPRODUCIBILITY STATEMENT

We provide full details of ThreadNIAH, including its three evidence-use levels, five task types, 29 templates, needle roles and relations, and the construction of 870 QA chains comprising 870 Main QA and 2,580 Sub-QA, in the main paper and appendix. We will release the benchmark annotations, construction and evaluation code, prompts, and configuration files. We also document the evaluated model versions, thinking variants, inference settings, random seeds, and compute resources to support reproduction.

REFERENCES

Alibaba Cloud. Vision understanding models: Qwen3.8-Max-0902. Model Studio documentation, 2026. URL https://help.aliyun.com/zh/model-studio/vision-model/. Qwen3.8-Max-0902 supports video input and 1M context; accessed 2026-09-25.

ByteDance. Doubao seed 2.1 pro. Model documentation, 2026.

Guo Chen, Yicheng Liu, Yifei Huang, Yuping He, Baoqi Pei, Jilan Xu, Yali Wang, Tong Lu, and Limin Wang. CG-Bench: Clue-grounded question answering benchmark for long video understanding. *arXiv preprint arXiv:2412.12075*, 2024. URL https://arxiv.org/abs/2412.12075.

Tao Chen, Kun Zhang, Qiong Wu, Xiao Chen, Chao Chang, Xiaoshuai Sun, Yiyi Zhou, and Rongrong Ji. Scaling the long video understanding of multimodal large language models via visual memory mechanism. *arXiv preprint arXiv:2603.29252*, 2026. URL https://arxiv.org/abs/2603.29252.

Junjie Fei, Jun Chen, Zechun Liu, Yunyang Xiong, Chong Zhou, Wei Wen, Junlin Han, Mingchen Zhuge, Saksham Suri, Qi Qian, Shuming Liu, Lemeng Wu, Raghuraman Krishnamoorthi, Vikas Chandra, Mohamed Elhoseiny, and Chenchen Zhu. Small vision-language models are smart compressors for long video understanding. *arXiv preprint arXiv:2604.08120*, 2026. URL https://arxiv.org/abs/2604.08120.

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*, pp. 24108–24118, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Fu_Video-MME_The_First-Ever_Comprehensive_Evaluation_Benchmark_of_Multi-modal_LLMs_in_CVPR_2025_paper.html.

Zhe Gao, Shiyu Shen, Taifeng Chai, Weinong Wang, Haotian Xu, Xing Wu, Wenbin Li, Qi Fan, Yang Gao, and Dacheng Tao. VideoTIR: Accurate understanding for long videos with efficient tool-integrated reasoning. *arXiv preprint arXiv:2603.25021*, 2026. URL https://arxiv.org/abs/2603.25021.

Google DeepMind. Gemini 3 models. Model documentation, 2026a. URL https://ai.google.dev/gemini-api/docs/models.

Google DeepMind. Gemini 3.8 Flash. Official model specifications, 2026b. URL https://deepmind.google/models/gemini/flash/. Video input and 1M input tokens; accessed 2026-09-25.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. RULER: What's the real context size of your long-context language models? In *Proceedings of the First Conference on Language Modeling*, 2024. URL https://arxiv.org/abs/2404.06654.

Greg Kamradt. Needle in a haystack: Pressure testing LLM long-context retrieval. GitHub repository, 2023. URL https://github.com/gkamradt/needle-in-a-haystack.

Yuri Kuratov, Aydar Bulatov, Petr Anokhin, Ivan Rodkin, Dmitry Sorokin, Artyom Sorokin, and Mikhail Burtsev. BABILong: Testing the limits of LLMs with long context reasoning-in-a-haystack. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-3381. URL https://arxiv.org/abs/2406.10149.

Miaoyu Li, Qin Chao, and Boyang Li. Two causally related needles in a video haystack. In *Advances in Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2505.19853. Datasets and Benchmarks Track.

Xinhao Li, Yi Wang, Jiashuo Yu, Xiangyu Zeng, Yuhan Zhu, Haian Huang, Jianfei Gao, Kunchang Li, Yinan He, Chenting Wang, Yu Qiao, Yali Wang, and Limin Wang. VideoChat-Flash: Hierarchical compression for long-context video modeling. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=MUjdNcfNPv.

Karttikeya Mangalam, Raiymbek Akshkulakov, and Jitendra Malik. EgoSchema: A diagnostic benchmark for very long-form video language understanding. *arXiv preprint arXiv:2308.09126*, 2023. URL https://arxiv.org/abs/2308.09126.

Viorica Pătrăucean, Lucas Smaira, Ankush Gupta, Adrià Recasens Continente, Larisa Markeeva, Dylan Banarse, Skanda Koppula, Joseph Heyward, Mateusz Malinowski, Yi Yang, Carl Doersch, Tatiana Matejovicova, Yury Sulsky, Antoine Miech, Alex Frechette, Hanna Klimczak, Raphael Koster, Junlin Zhang, Stephanie Winkler, Yusuf Aytar, Simon Osindero, Dima Damen, Andrew Zisserman, and João Carreira. Perception test: A diagnostic benchmark for multimodal video models. *arXiv preprint arXiv:2305.13786*, 2023. URL https://arxiv.org/abs/2305.13786.

Jihao Qiu, Lingxi Xie, Xinyue Huo, Qi Tian, and Qixiang Ye. LongVideo-R1: Smart navigation for low-cost long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 40505–40515, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Qiu_LongVideo-R1_Smart_Navigation_for_Low-cost_Long_Video_Understanding_CVPR_2026_paper.html.

Qwen Team. Qwen3.5 and qwen3.7 model family. Model documentation, 2026. URL https://github.com/QwenLM.

Xichen Tan, Yuanjing Luo, Yunfan Ye, Fang Liu, and Zhiping Cai. ALLVB: All-in-one long video understanding benchmark. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 7211–7219, 2025. URL https://ojs.aaai.org/index.php/AAAI/article/view/32775.

Yuxuan Wang, Yiqi Song, Cihang Xie, Yang Liu, and Zilong Zheng. VideoLLaMB: Long streaming video understanding with recurrent memory bridges. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 24170–24181, 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Wang_VideoLLaMB_Long_Streaming_Video_Understanding_with_Recurrent_Memory_Bridges_ICCV_2025_paper.html.

Xilin Wei, Xiaoran Liu, Yuhang Zang, Xiaoyi Dong, Pan Zhang, Yuhang Cao, Jian Tong, Haodong Duan, Qipeng Guo, Jiaqi Wang, Xipeng Qiu, and Dahua Lin. VideoRoPE: What makes for good video rotary position embedding? In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 66118–66136. PMLR, 2025. URL https://proceedings.mlr.press/v267/wei25h.html.

Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. LongVideoBench: A benchmark for long-context interleaved video-language understanding. In *Advances in Neural Information Processing Systems*, volume 37, pp. 28828–28857, 2024. doi: 10.52202/079017-0907. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/329ad516cf7a6ac306f29882e9c77558-Abstract-Datasets_and_Benchmarks_Track.html.

Xiaomi. MiMo-V2.6-Pro. Model documentation, 2026. URL https://mimo.mi.com/models/zh-CN/mimo-v2.6-pro.

Wenda Xu, Sweta Agrawal, Vilém Zouhar, Markus Freitag, and Daniel Deutsch. When llms benchmark themselves: Deconstructing self-bias in automated evaluation, 2025. URL https://arxiv.org/abs/2509.26600.

Zuhao Yang, Sudong Wang, Kaichen Zhang, Keming Wu, Sicong Leng, Yifan Zhang, Bo Li, Chengwei Qin, Shijian Lu, Xingxuan Li, and Lidong Bing. LongVT: Incentivizing "thinking with long videos" via native tool calling. *arXiv preprint arXiv:2511.20785*, 2025. URL https://arxiv.org/abs/2511.20785.

Daichi Yashima, Shuhei Kurita, Yusuke Oda, and Komei Sugiura. ReMoRa: Multimodal large language model based on refined motion representation for long-video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 31845–31855, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Yashima_ReMoRa_Multimodal_Large_Language_Model_based_on_Refined_Motion_Representation_CVPR_2026_paper.html.

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

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

Zijia Zhao, Haoyu Lu, Yuqi Huo, Yifan Du, Tongtian Yue, Longteng Guo, Bingning Wang, Weipeng Chen, and Jing Liu. Needle in a video haystack: A scalable synthetic evaluator for video MLLMs. In *International Conference on Learning Representations*, 2025.

Zhipu AI. GLM-5.3-Flash. Model documentation, 2026.

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*, pp. 13691–13701, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Zhou_MLVU_Benchmarking_Multi-task_Long_Video_Understanding_CVPR_2025_paper.html.

A Construction Details

We trace `count_driven_transfer_034` from a construction template to its synthesized long video and QA chain.

A.1 STATISTICS

Table 3 summarizes the evaluation set of ThreadNIAH. It contains 870 QA chains constructed from 29 task templates across five task types, including 870 Main QAs and 2,580 Sub-QAs. Each template is instantiated into 30 QA chains. The average chain length is 3.97, with chain lengths ranging from 2 to 6 depending on the task template and the required intermediate steps.

A.2 A REPRESENTATIVE TASK TEMPLATE

The L3 Counting template `count_driven_transfer` follows $\mathcal{T} = (\mathcal{N}, \mathcal{R}, Q, \mathcal{S})$ in §3.4. A *signature* combines a code word and clue text with the shape's type, color, fill, and location and the dots' count, color, and location. Here $\mathcal{S} = (s_1, s_2, s_3)$; each step specifies what its Sub-QA probes, while the question and options are written for the instance.

Count-Driven Transfer (`count_driven_transfer`)

$\mathcal{N}$ **(needle roles).** Nine clips appear in this order: COUNT, partial PICK, PICK, COUNT, PICK, unrelated, COUNT, target PICK, PICK.

$\mathcal{R}$ **(relations).** The three COUNT clips share one full signature; the four PICK clips share another. The partial PICK matches its family except for dot color and is excluded. The number of COUNT clips determines which full PICK clip to select in temporal order: here, the third.

$\mathcal{Q}$ **(Main QA pattern).** Count [COUNT signature] clips to obtain n; select the n-th clip matching [PICK signature] in temporal order; answer [a question about the selected clip's natural scene].

$\mathcal{S}$ **(Sub-QA steps).** Step 1 counts clips matching the full COUNT signature. Step 2 excludes the partial PICK and recovers the full PICK clips in temporal order. Step 3 uses the count to select a PICK clip and identifies its natural scene without asking for the Main QA's answer.

A.3 FROM TEMPLATE TO INSTANCE

For `count_driven_transfer_034`, we drew nine short clips from collected videos in nine different categories and assigned them to the roles above. The COUNT and PICK signatures were sampled from predefined clue values, subject to the template's constraints: COUNT has a hollow shape, while PICK has a solid shape and two dots. We copied each signature to its full matches, changed only the dot color for the partial PICK, and sampled a separate signature for the unrelated clip. Table 4 groups the clips by role and lists their positions in the synthesized video.

The three COUNT clips set $n = 3$. Among full PICK matches, positions 3, 5, 8, and 9 are first through fourth; position 2 is excluded because its dots are red. The target is therefore position 8. Gemini received eight sampled frames from this target clip before any clues were overlaid. Its generation prompt is shown below; the {`focus`} slot is sampled from preferences for object detail, readable text or digits, color or state, small counts, spatial relations, and local actions.

Table 3: The evaluation set of ThreadNIAH. Avg. L is the mean chain length, counting the Sub-QA and the Main QA.

Level	Task type	# Templates	# Chains	Avg. L
L1	Details	8	240	2.12
L2	Tracking	7	210	4.57
	Comparison	4	120	4.75
L3	Counting	4	120	5.00
	Logic	6	180	4.50
	Total	29	870	3.97

Table 4: Inserted clips in `count_driven_transfer_034`. Positions refer to their order in the synthesized video.

Position	Role	Overlaid clues
1, 4, 7	COUNT	`ORBIT`/`COUNT`; hollow yellow square at top right; one blue dot at bottom left.
2	Partial PICK	Same as PICK, except the two dots are red.
3, 5, 8, 9	PICK	`PRISM`/`PICK`; solid red square at top left; two white dots at bottom right.
6	Unrelated	`DELTA`/`SPARE`; does not match either signature.

Gemini prompt for target-clip QA generation

System: You are designing a hard natural-scene visual QA. Return JSON only.

User: Create one difficult English multiple-choice visual QA about this short clip.
Requirements:
– Ask about the natural scene content, not any overlay that may be added later.
– Do not mention clip numbers, timestamps, "this clip", "the video", overlay text, codewords, shapes, or dots.
– Prefer {`focus`}.
– Make exactly four plausible options A-D of similar style and length.
– The correct answer must be visually grounded and not obvious from commonsense alone.
– Output JSON only with keys `question`, `options`, and `answer`.

Gemini generated the scene question, four options, and reference answer from the target clip. The template code prepended the clue-based selection instructions to the scene question to obtain the following Main QA:

Final Main QA: `count_driven_transfer_034`

Question. First count how many inserted clips simultaneously show code `ORBIT`, clue text `COUNT`, a hollow yellow square in the top right, and 1 blue dot in the bottom left. Call that number N. Then, across the whole long video, keep only clips that simultaneously show code `PRISM`, clue text `PICK`, a solid red square in the top left, and 2 white dots in the bottom right. Ignore all partial matches in the PICK family. Choose the N-th remaining PICK-family clip in physical order. In that chosen clip, answer this: What color is the top worn by the woman?

Options. (A) Solid green (B) Bright red (C) Light gray (D) Navy blue.

Answer. A (Solid green).

A.4 From Main QA to QA Chain

To instantiate $\mathcal{S}$, we use the Main QA, the assigned clues and planned clip order, and captions and observations of the source clips. We package these as text for Gemini; it does not receive the nine video clips in this step. The task-specific prompt requests one four-option question and answer for each of the three steps. Each question states the conditions it needs, so it can be answered from the long video without using an earlier answer. We check the step sequence, option format, and answer fields before placing the Main QA fourth in the chain.

Selected instructions from the Sub-QA generation prompt

System: You are a ThreadNIAHlong-video benchmark dataset engineer. Given the authoritative context for one main video QA, generate its fixed diagnostic sub-QA prefix chain in English. Return strict JSON only.

User: Task type: `count_driven_transfer`. Generate exactly 3 prefix sub-QAs. Do not generate the terminal main QA; it is appended deterministically by the runtime.
Fixed blueprint: (1) count full COUNT matches; (2) exclude the partial PICK and order the full PICK matches; (3) transfer the count to the PICK ordinal and identify the selected scene without revealing the Main QA answer.
Every question must be independently answerable from the full video and must restate all visible conditions it needs. Questions and options may mention only visible overlay facts and caption-supported scene content. Use exactly four balanced A-D options with one correct answer. Do not test the terminal answer in a prefix. Before returning JSON, solve each prefix sub-QA using only its question and options; rewrite steps that permit text-only shortcuts.
Return one object with the key `sub_qa_chain`, containing three steps with `step`, `primitive`, `question`, `options`, and `answer` fields.

[Worked example]

[Instance-specific context pack]

For `count_driven_transfer_034`, the first Sub-QA asks for the number of full COUNT matches; its answer is three (option C). The second asks which partial PICK scene to exclude and how to order the four full PICK scenes; option A excludes the man in a dark vest gesturing with papers and retains the clips at positions 3, 5, 8, and 9 in order. The third transfers the count to the PICK family and asks for the selected scene. Its question and options are:

Sub-QA at Step 3: Target Scene

Question. Across the full video, first count the inserted clips that simultaneously show code `ORBIT`, clue text `COUNT`, an upper-right hollow yellow square, and 1 lower-left blue dot; call the count N. Then retain only clips that show code `PRISM`, clue text `PICK`, an upper-left solid red square, and 2 lower-right white dots, exclude every partial match, order them physically, and select the N-th full PICK clip. Which natural scene does it show?

Options. (A) An elderly man wearing glasses and a red tie speaks at a podium. (B) A smiling woman with blonde hair talks in a dimly lit setting. (C) A young woman sits outdoors at a wooden bench holding a beverage can. (D) A man with glasses and a hoodie sits in front of computer screens.

Answer. B (the blonde woman in a dimly lit setting).

We burn the assigned clues into the nine short clips and insert them, in the specified order, at dispersed positions in a separate long gastronomy video. The resulting video is paired with this four-question QA chain.

A.5 Automatic Quality Checks

Answer verification. We give a verification model sampled frames from the clips with the assigned clues rendered, together with the question and its four options. The reference answer is withheld from the model. The question passes this check only if the model returns the reference answer.

Prompt for answer verification

System: You are a strict long-video visual QA validator. Use only visual evidence from the provided clips and output only A, B, C, or D.
User: Answer the following long-video visual multiple-choice question using the provided clips. Output only one letter: A, B, C, or D.
`Question: {question}`
`Options: {options}`

Text-only shortcut check. We give a model the question and options without any clip frames and ask it to answer twice in separate calls. If both answers match the reference answer, we reject the question.

Prompt for the text-only shortcut check

System: You are guessing from text only.
User: Answer without seeing the clip. Output only A, B, C, or D.
`Question: {question}`
`Options: {options}`

B Human Quality Check

We assessed QA quality through independent human review. We recruited graduate students working in AI, provided common evaluation instructions, and used a practice round to screen for outlying responses, retaining ten annotators for the formal check. We sampled five QA chains from each of the 29 task templates, yielding 145 chains and 575 questions. Each chain was reviewed independently by two different annotators. A question passed only if it was answerable from the video, had a unique answer, and the provided answer was correct. Table 5 summarizes the paired judgments. Gwet's AC1 was 0.914, and both annotators passed 521 of the 575 questions (90.6%). Across all 1,150 individual judgments, 1,086 were pass, giving an overall pass rate of 94.4%.

Table 5: Paired quality judgments on 575 questions.

First rating	**Second rating**	
	Pass	Fail
Pass	521	18
Fail	26	10

C Baseline Settings

Each question is evaluated independently on the full synthesized video. For the six API models and two Qwen3.5 models served with vLLM, the prompt requests a JSON object such as `{"answer":"A"}`. The evaluator accepts only one A–D answer and repeats an invalid request up to three times with the same input. Unspecified decoding parameters retain the deployed model or provider defaults.

Gemini 3.6 Flash. We submit native video at 1 fps. The standard run sets the thinking level to minimal, and the additional thinking run sets it to high. Both use API-level JSON Schema for the answer.

Qwen3.7-Plus. We submit native video at 1 fps. The standard run disables thinking and uses JSON Schema; the additional run enables thinking and uses JSON Object.

Qwen3.7-Max. We submit native video at 1 fps. The standard run disables thinking and uses JSON Schema, while the additional run enables thinking and uses JSON Object.

Doubao Seed 2.1 Pro. We submit native video at 1 fps, disable thinking, and use JSON Schema for the answer.

GLM-5.3-Flash. We use native video input without setting an FPS parameter. Thinking remains enabled: the standard run uses low reasoning effort and the additional run uses high effort. Both use JSON Object.

MiMo V2.6 Pro. We submit native video at 1 fps. The standard run disables thinking, and the additional run enables it. Both use JSON Object for the answer. The thinking-enabled results combine responses from MiMo V2.6 Pro and its ultraspeed variant.

Table 6: Accuracy (%) on natural long-video QA. Overall is computed over all valid questions. Bold marks the highest score in each column.

Model	Details	Tracking	Comparison	Counting	Logic	Overall
Gemini 3.6 Flash	**70.0**	45.0	40.0	26.7	43.3	**48.6**
GLM-5.3-Flash	52.5	25.0	**47.1**	26.7	26.7	39.7
MiMo V2.6 Pro	50.0	26.3	29.0	**30.0**	**46.2**	38.2
Qwen3.5-27B	52.5	**50.0**	34.3	20.0	23.3	37.2
Qwen3.5-9B	52.5	25.0	37.1	15.0	26.7	34.5
VideoChat-Flash-7B	41.0	15.0	31.4	25.0	30.0	30.6
LongVT-RFT	34.3	10.0	38.2	10.0	33.3	28.1

Qwen3.5-9B. We serve the model with vLLM, targeting 1 fps with a 768-frame processor limit and dynamic spatial scaling. Thinking is disabled, and the answer uses JSON Schema. The deployed generation defaults are temperature 1.0, top-p 1.0, and top-k 0.

Qwen3.5-27B. We serve the model with vLLM, targeting 1 fps with a 768-frame processor limit and dynamic spatial scaling. Thinking is disabled, and the answer uses JSON Schema. The deployed generation defaults are temperature 0.6, top-p 0.95, and top-k 20.

VideoChat-Flash-Qwen2-7B (res448). We use its native inference procedure, with dynamic 1 fps sampling capped at 512 frames. Decoding is greedy (temperature 0, no sampling, one beam) with at most 1,024 new tokens. We parse the native final answer deterministically as an A–D choice.

LongVT-RFT. Its native procedure samples the full video at 1 fps, capped at 768 frames and 50,176 pixels per frame. It can make up to five video-crop calls, each sampling at 1 fps with at most 128 frames. Decoding uses temperature 0 and at most 49,152 tokens per model call; we parse the native final answer deterministically as an A–D choice.

D Agreement in Task Difficulty with Natural Long-Video QA

We examine whether task difficulty in ThreadNIAHaligns with that in natural long-video QA. Human annotators construct 145 questions, five per template, preserving each template's reasoning requirements while grounding the questions in natural video content. API content filtering limits evaluation coverage for some baselines. Table 6 reports the seven models with at least 90% valid-answer coverage in both settings.

We compare these results with performance on the 870 Main QA in ThreadNIAHat two levels: the five task types and the 29 templates. In each setting, we first average correctness across valid model responses to each question. We then average these question scores within each task type or template, giving every question equal weight. We rank the task types and templates separately by these scores and compute Spearman's correlation between the corresponding rankings in the two settings, using average ranks for ties.

Across the five task types, Spearman's correlation is $\rho = 0.70$ (95% CI: $[0.10, 0.90]$); across the 29 templates, it is $\rho = 0.569$ (95% CI: $[0.111, 0.611]$). We estimate percentile confidence intervals from 100,000 bootstrap replicates, independently resampling questions with replacement within each template in each setting while keeping the model set and template sample sizes fixed. We recompute group scores and rankings in each replicate. The positive association at both levels suggests that harder tasks in ThreadNIAH also tend to be harder in natural long-video QA.

E Task Template Catalog

This section describes the 29 multiple-choice templates summarized in Table 3. For each template, $\mathcal{N}$ names the roles of the inserted clips, $\mathcal{R}$ states the relation used to identify or combine them, $\mathcal{Q}$ gives the Main QA pattern, and $\mathcal{S}$ lists the diagnostic Sub-QA steps. Clue values and natural-scene questions vary across instances. The Sub-QA steps describe what is probed; each question is independently answerable from the video.

E.1 DETAILS (L1)

Single-Anchor Detail Lookup

$\mathcal{N}$ **(needle roles).** A target clip appears among unrelated clips.
$\mathcal{R}$ **(relations).** A natural-scene description identifies the target.
$\mathcal{Q}$ **(Main QA pattern).** Find that clip and answer a fine visual detail within it.
$\mathcal{S}$ **(Sub-QA steps).** Locate the target and bind it to an answer-neutral scene detail.

Single-Anchor Color or State

$\mathcal{N}$ **(needle roles).** One target and several other clips.
$\mathcal{R}$ **(relations).** A combination of natural visual cues singles out the target.
$\mathcal{Q}$ **(Main QA pattern).** Report the color, material, appearance, or state of an element in that clip.
$\mathcal{S}$ **(Sub-QA steps).** Find the target and identify other visible evidence from the same occurrence.

Single-Anchor Text Reading

$\mathcal{N}$ **(needle roles).** A target clip and distractor clips.
$\mathcal{R}$ **(relations).** Natural visual cues locate the clip containing the relevant text.
$\mathcal{Q}$ **(Main QA pattern).** Read a word, label, logo, or number in its natural scene.
$\mathcal{S}$ **(Sub-QA steps).** Locate the clip and identify a separate visible attribute without giving away the requested text.

Single-Anchor Spatial Relation

$\mathcal{N}$ **(needle roles).** A target clip and other clips.
$\mathcal{R}$ **(relations).** A natural-scene cue identifies the clip in which two entities must be compared.
$\mathcal{Q}$ **(Main QA pattern).** Answer where one entity is positioned relative to the other.
$\mathcal{S}$ **(Sub-QA steps).** Locate the target using a visible property separate from the queried spatial relation.

Single-Anchor Action Phase

$\mathcal{N}$ **(needle roles).** A target clip and other clips.
$\mathcal{R}$ **(relations).** Natural visual cues identify the relevant action scene.
$\mathcal{Q}$ **(Main QA pattern).** Determine a person's action or motion state within that scene.
$\mathcal{S}$ **(Sub-QA steps).** Locate the clip and verify it using co-occurring visual evidence that does not reveal the action answer.

Single-Anchor Counting

$\mathcal{N}$ **(needle roles).** One target clip and distractors.
$\mathcal{R}$ **(relations).** A natural-scene description selects the clip before any count is made.
$\mathcal{Q}$ **(Main QA pattern).** Count a small set of visible objects or people in that clip.
$\mathcal{S}$ **(Sub-QA steps).** Locate the target and identify answer-neutral evidence from the same scene.

Single-Clip Exclusion Logic

$\mathcal{N}$ **(needle roles).** A desired clip, clips satisfying only part of the description, and other distractors.

$\mathcal{R}$ **(relations).** Positive and negative conditions leave a single eligible scene; some instances express these conditions through visible SET labels.

$\mathcal{Q}$ **(Main QA pattern).** Answer a natural visual detail in the surviving clip.

$\mathcal{S}$ **(Sub-QA steps).** Bind the relevant labels to scenes, then apply the inclusion and exclusion conditions to identify the survivor.

Decoy Detail Identification

$\mathcal{N}$ **(needle roles).** A target appears among visually similar clips, with other clips outside the relevant group.

$\mathcal{R}$ **(relations).** A visible SET label narrows the group, and a natural visual cue distinguishes the target within it.

$\mathcal{Q}$ **(Main QA pattern).** Answer a fine detail in the selected clip.

$\mathcal{S}$ **(Sub-QA steps).** Identify the target scene using the group label and the distinguishing cue without asking for the final detail.

E.2 TRACKING (L2)

First-Match Attribute

$\mathcal{N}$ **(needle roles).** Several clips share a visible SET label; other clips do not.

$\mathcal{R}$ **(relations).** Temporal order distinguishes the earliest eligible occurrence.

$\mathcal{Q}$ **(Main QA pattern).** Answer a natural visual detail in that first occurrence.

$\mathcal{S}$ **(Sub-QA steps).** Read the labeled clips in order, construct the eligible set, and bind its first member to a scene.

Last-Match Attribute

$\mathcal{N}$ **(needle roles).** Several clips share a visible SET label alongside other clips.

$\mathcal{R}$ **(relations).** Temporal order distinguishes the latest eligible occurrence.

$\mathcal{Q}$ **(Main QA pattern).** Answer a natural visual detail in that last occurrence.

$\mathcal{S}$ **(Sub-QA steps).** Read the labeled clips in order, construct the eligible set, and bind its last member to a scene.

Paired Role Binding

$\mathcal{N}$ **(needle roles).** Two relevant clips share a SET label, with additional clips as distractors.

$\mathcal{R}$ **(relations).** Their order assigns distinct roles, such as earlier and later, to the two scenes.

$\mathcal{Q}$ **(Main QA pattern).** Choose the attributes or actions that belong to the respective roles.

$\mathcal{S}$ **(Sub-QA steps).** Retrieve the labeled clips, identify the relevant pair, and bind each role to its scene.

Double-Condition Recurrence Tracking

$\mathcal{N}$ **(needle roles).** Repeated full matches occur among clips satisfying only one of two conditions.

$\mathcal{R}$ **(relations).** A clip qualifies only when it matches both a family signature and a second gate condition; qualifying clips are ordered by appearance.

$\mathcal{Q}$ **(Main QA pattern).** Select the specified occurrence and answer a detail in its natural scene.

$\mathcal{S}$ **(Sub-QA steps).** Recover the clip order, read the clue profiles, filter by both conditions, and identify the selected occurrence.

Compound First-After-Key

$\mathcal{N}$ **(needle roles).** A reference clip, full matches before and after it, and partial-match decoys.

$\mathcal{R}$ **(relations).** Only clips after the reference that satisfy the complete target signature qualify.

$\mathcal{Q}$ **(Main QA pattern).** Answer a natural-scene detail in the first eligible later clip.

$\mathcal{S}$ **(Sub-QA steps).** Recover the clip order, read the clue profiles, filter after the key, and identify the first full match.

Compound Last-Before-Key

$\mathcal{N}$ **(needle roles).** A reference clip, earlier full and partial matches, and a later full-match decoy.

$\mathcal{R}$ **(relations).** Only full signature matches before the reference qualify.

$\mathcal{Q}$ **(Main QA pattern).** Answer a natural-scene detail in the last eligible earlier clip.

$\mathcal{S}$ **(Sub-QA steps).** Recover the clip order, read the clue profiles, filter before the key, and identify the last full match.

Dual-Key Window Binding

$\mathcal{N}$ **(needle roles).** Two key clips bound a temporal window; full and partial matches appear inside and outside it.

$\mathcal{R}$ **(relations).** A candidate must lie strictly between the keys and match the complete target signature.

$\mathcal{Q}$ **(Main QA pattern).** Choose the specified in-window occurrence and answer about its natural scene.

$\mathcal{S}$ **(Sub-QA steps).** Recover the order and clue profiles, filter within the window, and bind the selected match to a scene.

E.3 COMPARISON (L2)

Paired Attribute Match

$\mathcal{N}$ **(needle roles).** Two clips share a query SET label among other labeled clips.

$\mathcal{R}$ **(relations).** Their natural visual attributes must be compared across the pair.

$\mathcal{Q}$ **(Main QA pattern).** Choose the option that correctly describes the requested attributes in the two clips.

$\mathcal{S}$ **(Sub-QA steps).** Retrieve the labeled clips and identify the relevant pair before making the comparison.

Cross-Clip Micro Comparison

$\mathcal{N}$ **(needle roles).** Several clips form an eligible labeled set, while similar clips belong to another set.

$\mathcal{R}$ **(relations).** The comparison is restricted to eligible scenes and depends on a fine natural visual difference.

$\mathcal{Q}$ **(Main QA pattern).** Identify the clip or attribute that satisfies the stated comparison.

$\mathcal{S}$ **(Sub-QA steps).** Retrieve and label the scenes, construct the eligible set, and compare the requested attributes within it.

Reference Standard Drift Comparison

$\mathcal{N}$ **(needle roles).** A reference clip is followed by changed candidates, an exact copy, and unrelated clips.

$\mathcal{R}$ **(relations).** Eligible later candidates preserve the reference's identifying fields; drift is counted over specified visual-clue attributes, and an exact copy is excluded.

$\mathcal{Q}$ **(Main QA pattern).** Select the candidate with the smallest positive drift and answer a natural-scene detail.

$\mathcal{S}$ **(Sub-QA steps).** Recover the order, identify the reference, form the later candidate set, and find the minimum positive drift.

Mirror-Pair Later Binding

$\mathcal{N}$ **(needle roles).** One valid pair appears among false pairs and unrelated clips.

$\mathcal{R}$ **(relations).** The valid pair shares specified clue fields while placing its shape in opposite corners and using opposite fill modes.

$\mathcal{Q}$ **(Main QA pattern).** Select the later member of that pair and answer a natural-scene detail.

$\mathcal{S}$ **(Sub-QA steps).** Retrieve the ordered clips, bind their clue profiles to scenes, identify the valid pair, and select its later member.

E.4 COUNTING (L3)

Conditional Counting

$\mathcal{N}$ **(needle roles).** Clips with an eligible SET label appear alongside excluded labeled clips.

$\mathcal{R}$ **(relations).** Both the label and a natural visual property determine which clips count.

$\mathcal{Q}$ **(Main QA pattern).** Report how many eligible clips show the stated property.

$\mathcal{S}$ **(Sub-QA steps).** Retrieve the labeled scenes, separate eligible from excluded clips, and classify the eligible clips by the visual property.

Count-Difference Transfer

$\mathcal{N}$ **(needle roles).** Two count families, an ordered pick family, and a partial pick decoy.

$\mathcal{R}$ **(relations).** Subtract the second family's count from the first and use the difference as an ordinal in the full-match pick family.

$\mathcal{Q}$ **(Main QA pattern).** Answer a natural-scene detail in the selected pick clip.

$\mathcal{S}$ **(Sub-QA steps).** Bind scenes to clue profiles, compute the count difference, order the valid pick clips, and transfer the difference to the target.

Count-Driven Ordinal Transfer

$\mathcal{N}$ **(needle roles).** A count family, an ordered pick family, a partial pick decoy, and unrelated clips.

$\mathcal{R}$ **(relations).** The number of full count-family matches gives the ordinal in the full-match pick family.

$\mathcal{Q}$ **(Main QA pattern).** Answer a natural-scene detail in that pick clip.

$\mathcal{S}$ **(Sub-QA steps).** Count the first family, filter and order the second, then bind the selected clip to its scene. Section A.2 gives a full example.

Route-Then-Count Transfer

$\mathcal{N}$ **(needle roles).** A start key precedes count and pick families, including a partial pick decoy.

$\mathcal{R}$ **(relations).** Text on the start key names the two families; only later full matches count, and the count selects an ordinal pick clip.

$\mathcal{Q}$ **(Main QA pattern).** Read the route, count the designated family, and answer a natural-scene detail in the selected pick clip.

$\mathcal{S}$ **(Sub-QA steps).** Locate the key, read its clue, count the routed family, order the valid pick clips, and identify the selected scene.

E.5 LOGIC (L3)

Branch Route Tracking

$\mathcal{N}$ **(needle roles).** A start clip and several possible clips at each of three clue-guided hops.

$\mathcal{R}$ **(relations).** Each clue names the next code and an additional keep condition; same-code clips that fail that condition are decoys.

$\mathcal{Q}$ **(Main QA pattern).** Follow the three-hop route and answer a natural-scene detail in the final clip.

$\mathcal{S}$ **(Sub-QA steps).** Locate the start, read its clue, and resolve each successive branch using both the next code and keep condition.

Exception Then Next Match

$\mathcal{N}$ **(needle roles).** A near-matching family contains one exception; a separate later family contains full and partial matches.

$\mathcal{R}$ **(relations).** The exception is identified by a differing visual-clue field and sets the start of a forward search.

$\mathcal{Q}$ **(Main QA pattern).** Choose the first later full match of the second family and answer its natural-scene detail.

$\mathcal{S}$ **(Sub-QA steps).** Read the first family's clues, identify its exception, and find the first full match after it.

Majority-Break Neighbor

$\mathcal{N}$ **(needle roles).** A clue-defined family has one member that differs from the majority, with other clips nearby.

$\mathcal{R}$ **(relations).** The outlier's immediate predecessor in the video, rather than the outlier itself, is the target.

$\mathcal{Q}$ **(Main QA pattern).** Answer a natural-scene detail in that preceding clip.

$\mathcal{S}$ **(Sub-QA steps).** Construct the family, determine the majority pattern and its exception, then retrieve the predecessor.

Majority-Minority Vote

$\mathcal{N}$ **(needle roles).** A clue-defined family has two marker-pattern subgroups, plus unrelated clips.

$\mathcal{R}$ **(relations).** The less frequent subgroup is selected, and temporal order distinguishes its later member.

$\mathcal{Q}$ **(Main QA pattern).** Answer a natural-scene detail in that later minority clip.

$\mathcal{S}$ **(Sub-QA steps).** Identify family members, compare subgroup frequencies, and bind the later minority member to its scene.

Windowed Ordering

$\mathcal{N}$ **(needle roles).** Start and end keys surround several distinct full-match clips and a partial match; full-match decoys also lie outside the window.

$\mathcal{R}$ **(relations).** Only complete target signatures strictly between the keys enter the local order.

$\mathcal{Q}$ **(Main QA pattern).** Select the requested position in that order and answer a natural-scene detail.

$\mathcal{S}$ **(Sub-QA steps).** Identify the boundaries, build the ordered in-window set, and select the requested ordinal.

XOR Penultimate Selector

$\mathcal{N}$ **(needle roles).** Candidate clips share a common clue pattern; some satisfy one of two signatures, while decoys satisfy both or neither.

$\mathcal{R}$ **(relations).** Eligibility requires exactly one signature, and eligible clips are ordered by appearance.

$\mathcal{Q}$ **(Main QA pattern).** Select the penultimate eligible clip and answer a natural-scene detail.

$\mathcal{S}$ **(Sub-QA steps).** Retrieve the candidates, read the two signatures, classify them by exclusive-or, and select the penultimate match.