GROUNDING CONFUSABLE MOVIE MOMENTS

Anonymous authors
Paper under double-blind review

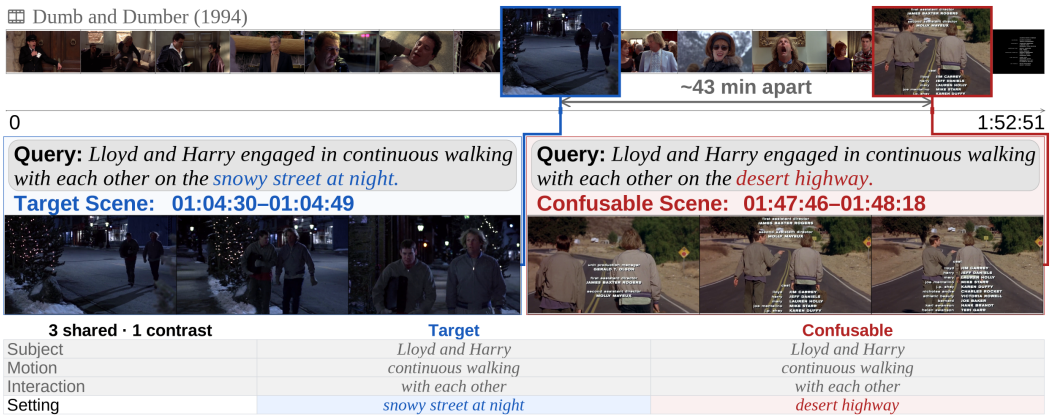


Figure 1: **An example of confusable moment grounding in a full-length movie from CoM^2-Bench.** Each pair in CoM^2-Bench consists of two moments whose queries share subject, motion, and interaction but differ only in setting: a snowy street at night vs. a desert highway. Each query is grounded independently over the full movie, so success requires verifying not only the shared attributes that dominate query-moment similarity but also the distinguishing details. CoM^2-Bench contains 46 movies totaling 90.4 hours, with 3,014 queries forming 1,507 confusable query pairs.

ABSTRACT

Characters, locations, and actions recur throughout a movie, so many of its moments differ only in fine-grained details. Grounding such a moment in a full-length movie requires resolving that detail rather than matching the attributes it shares with other moments, yet existing temporal movie grounding benchmarks rarely test this. We introduce CoM^2-Bench, a benchmark of confusable query pairs, where the two queries in each pair share most attributes but differ in a single distinguishing detail within a full movie, posing a challenge for recent temporal grounding approaches. To locate where grounding breaks down, we analyze a retrieve-then-localize pipeline, in which a retriever first selects a short candidate chunk and a localizer then performs temporal grounding within it. Separating the two stages lets us attribute each failure to either retrieval or localization. We find that retrieval is the major bottleneck: state-of-the-art retrievers often select a moment's confusable counterpart, because their choices follow the content the two moments share rather than the detail that tells them apart. To address these failures, we propose a simple, training-free remedy, Agentic Window Search (AWS): an MLLM agent *Decomposes* the query into attribute-level sub-queries, then *Inspects* each candidate window against them and *Expands* the window when evidence is cut off at its edge. AWS works with existing retrievers and localizers without additional training, raising mIoU from 18.5 to 26.8.

1 INTRODUCTION

Movies repeatedly bring the same characters, locations, and actions back on screen, producing look-alike moments that differ only in fine-grained details. For example, in *Dumb and Dumber* (Figure 1),

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Lloyd and Harry walk together in two moments. Grounding one specific moment therefore requires another detail: whether they are on a snowy street at night or on a desert highway. Video temporal grounding (VTG) localizes the start and end times of an event described by a natural-language query (Gao et al., 2017). As grounding extends from short clips to full-length movies, such repetition becomes the norm rather than the exception: the longer the video, the more moments tend to match most of a query. A grounding model for movie search, editing, or audio description must therefore find the one moment that satisfies every part of the query, not merely its most salient part.

Existing benchmarks, however, cannot tell whether models do this. Their queries can often be localized to a single salient entity or action, so a model that ignores the remaining details can still score well, and its errors on confusable moments go unmeasured. Long-video benchmarks (Soldan et al., 2022; Yuan et al., 2025) enlarge the search space, and compositional benchmarks (Zou et al., 2026) combine temporal and spatial conditions, but neither controls whether another moment in the same video shares most of the query. Some construction pipelines even remove repeated events within each video (Seo & Kim, 2026), discarding exactly the cases where this distinction matters. Although repeated descriptions within a movie are a recognized source of grounding ambiguity (Soldan et al., 2022), to our knowledge no benchmark deliberately constructs and verifies such cases. Current scores therefore cannot reveal whether a model grounds the full query or only the attributes it shares with other moments.

We therefore study whether temporal grounding models distinguish paired *confusable* movie moments that share most attributes but differ in one identifying detail, requiring them to match the query's combination of attributes. We build CoM^2-Bench to answer this, with 3,014 confusable queries over the original videos of 46 movies. Its design makes correct grounding depend on the distinguishing detail. Each pair shares three attributes and differs in one, and the two queries are written in parallel so that only this detail, not wording or sentence structure, separates them. We ground each query on its own, without its partner as a contrast, so a model cannot lean on the comparison and must locate the detail in the video. Because a confusable pair is resolved only when both of its moments are grounded correctly, we introduce PairSuccess, a pair-level metric that counts a pair as correct only when both independently grounded queries exceed a given IoU threshold. Human annotators finally check each query for validity, temporal boundaries, and uniqueness under explicit guidelines, with blind cross-review and adjudication, keeping the labels reliable. CoM^2-Bench is the only benchmark that pairs verified confusable moments in full-length movies (see Table 1).

On CoM^2-Bench, we first ask where the state-of-the-art retrieve-then-localize pipeline fails, following prior work that narrows the temporal search space before precise localization (Hannan et al., 2025; Seo & Kim, 2026). We adopt Movie RAG (Kim et al., 2026) for this analysis, with a chunk retriever that ranks video chunks by embedding similarity to the query and a temporal localizer, hereafter the retriever and the localizer. We use TimeLens2 (Zhu et al., 2026) as the localizer, which predicts boundaries within a window, initially one retrieved chunk. Our analysis identifies two limitations. First, the retriever is the dominant bottleneck: replacing the top-1 retrieved chunk with the *oracle* chunk, the one overlapping the ground truth most, raises grounding sharply (R@1@0.3 from 25.8 to 77.3, Table 2), so improving retrieval is the primary lever for performance. We trace this low retrieval accuracy to the retriever's inability to separate confusable pairs (Figure 4). Second, boundary errors persist even with the best overlapping chunk, partly because fixed chunks cut events at their edges, and extending that chunk within a small budget raises R@1@0.7 from 34.8 to 43.2 (Table 2).

To address these failures, we propose training-free Agentic Window Search (AWS), which operates within the retrieve-then-localize pipeline: between the two stages, an agent verifies the candidate window directly against the query or adjusts its length before passing one to the localizer. AWS first *Decomposes* the query into attribute-level sub-queries, so that the distinguishing detail is checked on its own rather than absorbed into the shared content. It then *Inspects* the top-ranked candidate windows in retrieval order, accepting a window only when every sub-query holds, and *Expands* a window within a fixed budget when evidence is cut off at its edge. In addition, movie queries often name characters, but a name alone does not reveal who appears on screen. We therefore build a character memory that links names to reference frames using the movie's dialogue and nearby frames, without evaluation queries or ground truth. The agent uses these references only to identify named characters during inspection, without adding appearance constraints absent from the query.

Table 1: **Comparison with existing video temporal grounding benchmarks.** *Confusable moments*: other moments in the same video that match most of the query (✓ deliberately constructed and verified, △ present but uncontrolled). *Attribute contrast*: paired queries differ in exactly one attribute with parallel phrasing. *Pair metric*: success requires both queries of a pair to be grounded. CoM^2-Bench is the only benchmark in which a model must separate a target from a verified confusable moment in the same full-length movie.

Benchmark	Scope (video len., m)	Avg. target len. (s)	# Queries	Confusable moments	Attribute contrast	Pair metric	Video access
Charades-STA (Gao et al., 2017)	clip (0.51)	8.1	3,720	✗	✗	✗	✓
QVHighlights (Lei et al., 2021)	clip (2.50)	24.6	1,542	✗	✗	✗	✓
MomentSeeker (Yuan et al., 2025)	long video (20.03)	12.8	1,800	✗	✗	✗	✓
CoMET-Bench (Zou et al., 2026)	long video (33.81)	6.9	2,789	✗	✗	✗	✓
MAD (Soldan et al., 2022)	full movie (110.77)	4.1	72,000	△	✗	✗	✗
CoM^2-Bench (Ours)	full movie (117.8)	60.2	3,014	✓	✓	✓	✓

Incorporated into the existing retrieve-then-localize pipeline of Movie RAG, AWS raises mIoU from 18.5 to 26.8. These gains show that verifying candidate evidence improves grounding without additional training, while the remaining gap to the oracle leaves room to improve both window selection and boundary prediction. Our contributions are summarized below.

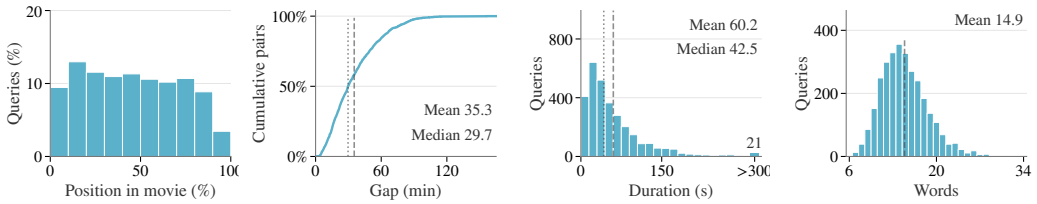
- We introduce CoM^2-Bench, a full-movie temporal grounding benchmark of human-verified confusable query pairs, in which the two queries of each pair share most attributes but differ in a single distinguishing detail.
- We identify chunk retrieval as the main bottleneck and show that the retriever fails to account for all attributes of a multi-attribute query: its choices are dominated by the attributes a moment shares with its confusable counterpart, while the single attribute that distinguishes them rarely changes the retrieved chunk.
- We propose Agentic Window Search (AWS), a training-free, plug-in agent that verifies candidate windows against attribute-level sub-queries between retrieval and localization, raising mIoU from 18.5 to 26.8 without modifying or retraining the retriever or localizer.

2 CoM^2-Bench: Confusable Movie Temporal Grounding

We introduce CoM^2-Bench (Confusable Movie Moments-Bench), a full-movie temporal grounding benchmark designed to evaluate whether a model can distinguish semantically similar moments occurring at different times in the same movie. We summarize the differences from existing benchmarks in Table 1 and defer the full discussion of related work to Appendix A.

Task and Benchmark Design. Unlike conventional benchmarks in which a query can often be localized using a single salient entity or action, CoM^2-Bench organizes queries into *confusable query pairs*. Each pair refers to two temporally distant moments that share semantic attributes but differ in one attribute (*i.e.*, confusable moments).

Given a full movie V of duration T and a text query q, a model predicts an interval $\hat{\tau} = [s, e]$, where $0 \le s < e \le T$. Each pair comprises a target query q_t and a paired confusable query q_c, with ground truth intervals τ_t and τ_c in the same movie. Both queries are evaluated independently as targets without the paired query as input, while the other moment serves as a confusable alternative. For the i-th of N pairs, we write its two queries as $q_{i,t}$ and $q_{i,c}$, with ground-truth intervals $\tau_{i,t}$ and $\tau_{i,c}$ and predictions $\hat{\tau}_{i,t}$ and $\hat{\tau}_{i,c}$. We define $\mathrm{PairSuccess}@\theta = (100/N)\sum_{i=1}^{N}\prod_{j\in\{t,c\}}\mathbf{1}[\mathrm{IoU}(\hat{\tau}_{i,j}, \tau_{i,j}) \ge \theta]$, the percentage of confusable pairs whose two queries are both grounded with temporal IoU of at least θ. Missing or invalid predictions count as failures, and all N pairs remain in the denominator. We call the ground-truth interval of a query a moment. During construction, moments originate from events that an MLLM proposes within subtitle-anchored clips (Appendix B.1). Pair identity is used only for aggregation, and the paired query, contrast label, and ground truth are never provided as inference input. Paired moments share three semantic attributes and differ in one, so that success requires resolving the discriminative attribute. Attributes include participating characters, action, object, setting, or temporal relation. Both queries use the same attribute roles and parallel linguistic structure. Figure 1 illustrates events sharing characters, motion, and interaction but differing in setting. This construction tests sensitivity to the distinguishing attribute without guaranteeing that every query requires all four attributes to be resolved jointly.



(a) GT position (b) Temporal distance (c) Moment duration (d) Query length

Figure 2: **Benchmark statistics.** Distributions of ground truth positions within movies, temporal distance between paired moments, durations of moments, and query lengths. Positions use interval midpoints normalized by movie duration, dashed and dotted lines mark means and medians, and the duration overflow bin includes all intervals longer than 300 s.

Dataset Construction. We source movies and subtitles from ALLVB (Tan et al., 2025). Subtitle-anchored clip segmentation groups consecutive cues into clips and expands their visual context using adjacent clip boundaries. A multimodal large language model (MLLM) uses the expanded video and subtitle derived clip descriptions to generate event intervals and structured attributes. We construct parallel confusable queries from shared and contrasting attributes, retaining pairs only after rigorous human verification. Details about construction and human verification are in Appendix B.

Benchmark Statistics. The current verified release of CoM^2-Bench contains 46 full-length movies with a combined duration of 90.4 hours. The average movie duration is 117.8 minutes. Human verification retains 1,507 pairs, corresponding to 3,014 temporal-grounding queries. The average query length is 14.9 words. Ground-truth moments have an average duration of 60.2 s and a median duration of 42.5 s. The paired moments are separated by an average of 35.3 minutes, with a median of 29.7 minutes, measured from the end of the earlier ground truth interval to the start of the later one. These statistics show that the benchmark emphasizes temporally distant confusable moments rather than local boundary distinctions. Figure 2 summarizes the distributions of ground truth positions, temporal distance of paired movie moments, moment durations, and query lengths.

Ground-truth moments are spread widely across movies, reinforcing the need for full movie search. With interval midpoints normalized by movie duration, each of the first nine deciles contains roughly 9 to 13% of queries, while the final decile, which largely covers end credits, contains 3.5%.

Comparison with Existing Benchmarks. Table 1 compares CoM^2-Bench with existing benchmarks in scale and design. CoM^2-Bench is the only benchmark that deliberately constructs and verifies confusable moments within the same full movie. MAD releases only precomputed CLIP features rather than the videos, and its average target length is only 4.1 s. In contrast, CoM^2-Bench builds on the movie release of ALLVB (Tan et al., 2025), enabling evaluation on raw frames, and its targets are complete moments averaging 60.2 s.

3 Diagnosing Confusable Movie Moment Temporal Grounding

With CoM^2-Bench in place, we ask how current grounding methods handle it and where they break down. We analyze a retrieve-then-localize pipeline, whose separate retrieval and localization stages let us attribute a failure to one stage, and adopt Movie RAG (Kim et al., 2026) as a representative and competitive instance (Table 3). Movie RAG has a chunk retriever that ranks movie chunks by query similarity and a temporal localizer that predicts event boundaries within the supplied window. We refer to these components as the retriever and the localizer.

A chunk is a retrieval unit, while a window is the span supplied to the localizer, initially one retrieved chunk and possibly extended beyond its edges (Section 4.1). Movie RAG reports fixed-length chunks to be competitive with scene segmentation, so we partition movies into fixed 180 s chunks for our diagnosis. We instantiate the retriever with Qwen3-VL-Embedding 2B (Li et al., 2026) and the localizer with TimeLens2 8B (Zhu et al., 2026), and use both throughout our diagnosis unless stated otherwise. The retriever embeds each chunk using all frames sampled at 1 fps together with its subtitles, embeds the text query, and ranks chunks by cosine similarity between the chunk and query embeddings. The upper part of Figure 3 shows this retrieve-then-localize pipeline.

Table 2: **Chunk retrieval, not localization, limits grounding.** Giving the localizer the oracle chunk (row 6) raises mIoU much more than a perfect localizer on the retrieved chunk (row 3, † marks upper bounds assuming predictions stay within the window). Fine-tuning the retriever or reranking (rows 4 and 5) closes only part of that gap. Extending the oracle chunk within the search's expansion budget (row 7) adds a further gain. The gray row 8 combines both oracles.

Row	Window selection	Localizer	R@1@0.3	R@1@0.5	R@1@0.7	mIoU
Temporal grounding on full-movies						
1	✗ (Full-movie)	TimeLens2	10.0	6.9	3.9	7.4
Base pipeline and bounds						
2	Retrieved top-1	TimeLens2	25.8	19.1	12.2	18.5
3	Retrieved top-1	Oracle†	34.8	31.9	28.1	31.0
Retrieval fixes (Section 3.4)						
4	Fine tuned retriever	TimeLens2	32.4	24.1	15.3	23.1
5	Top-5 reranking	TimeLens2	33.4	25.1	16.1	24.3
Oracle retriever reference						
6	Oracle Top-1	TimeLens2	77.3	57.9	34.8	53.8
7	Oracle Top-1 + Oracle expand	TimeLens2	79.9	63.0	43.2	58.3
8	Oracle Top-1	Oracle†	99.9	99.6	85.9	91.6

We hold one stage fixed while replacing the other with an oracle to assess how window selection and boundary prediction limit performance. All analyses in this section evaluate the full CoM^2-Bench.

3.1 Where Does the Pipeline Fail?

To identify the main cause of failure in the Movie RAG, we vary the window selection and the localizer with oracle substitutes and report the results in Table 2.

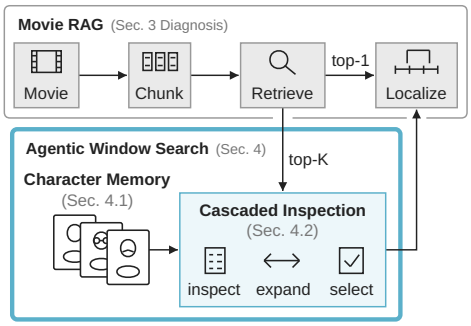


Figure 3: **Movie RAG and Agentic Window Search.** The upper row shows the retrieve-then-localize pipeline diagnosed in Section 3. The lower block replaces the window from the top-ranked chunk with agentic search over the top K candidates described in Section 4.

The retriever matters more than the localizer. To assess how much window selection limits accuracy, we first substitute the retriever's top-1 chunk with an oracle top-1 chunk, the one that overlaps the ground-truth interval most. Supplying this oracle chunk lets TimeLens2 reach 53.8 mIoU (Table 2, row 6), far above the 31.0 mIoU that even an oracle localizer reaches on the retrieved top-1 chunk (row 3). These results indicate frequent retrieval misses, with the retrieved top-1 chunk having no overlap with the ground truth for 62.8% of queries, as well as incomplete coverage when overlap exists. Specifically, for a single continuous ground truth interval, the maximal attainable IoU equals the fraction covered by the retrieved window, assuming the predicted interval lies entirely within that window. The same ceiling limits even the joint oracle in row 8 to 91.6 mIoU. Window selection thus caps what any localizer can achieve, since a localizer can only place boundaries inside the window it receives. A better localizer therefore helps little while the window misses or cuts the event, and its gains grow only once retrieval supplies the right chunk.

Fixed 180 s chunking truncates events. Even with the oracle chunk, localization loses accuracy, partly because fixed chunks ignore event boundaries and can cut an event at their edges. Extending the oracle chunk toward the event within the budget of Section 4 (at most three 30 s steps) raises mIoU from 53.8 to 58.3 (row 7), mostly at the strictest threshold (R@1@0.7 from 34.8 to 43.2), and the remaining gap to the oracle localizer reflects the localizer itself. Growing a window past a fixed chunk edge thus recovers part of the loss, which motivates the bounded expansion in Section 4.1.

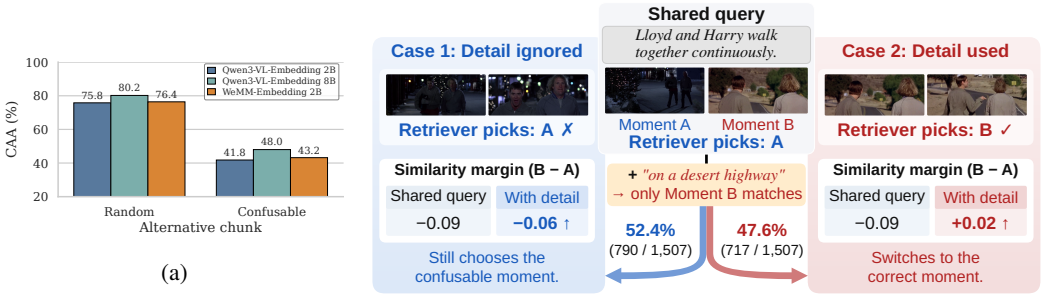


Figure 4: **The retriever confuses paired moments because it follows shared content.** (a) Chance-adjusted accuracy (CAA) at picking the correct chunk over one alternative, where 0% is chance. Every retriever loses much of its discrimination against the paired confusable moment. (b) For each query, we drop the distinguishing detail to form a shared query and compare the retriever's margin between the two paired oracle chunks with and without the detail. Among the queries from CoM^2-Bench, adding the detail switches only 47.6% to the correct one.

3.2 Do confusable moments cause the failure?

Since retrieval remains the dominant bottleneck, we next ask whether the retriever can tell oracle chunks from their paired confusable alternatives. To isolate this, we compare each query's oracle chunk against one alternative and select the chunk with higher query similarity under the retriever. In the Confusable condition, the alternative is the paired confusable query's oracle chunk. In the Random condition, accuracy is the exact expectation over uniformly sampled chunks from the same movie after excluding the oracle chunks of both paired queries and any chunk overlapping either ground truth interval. Reporting chance-adjusted accuracy (CAA), where 0% is chance and 100% is always correct (Figure 4a), the default retriever keeps 75.8% of its discrimination against a random chunk but only 41.8% against the confusable one, about half. A larger Qwen3-VL-Embedding 8B and a different embedding model, WeMM-Embedding (Zhou et al., 2026), show the same drop, so the confusion reflects neither model scale nor one particular embedding model. This raises the question of whether the retriever uses the detail that separates a pair at all.

3.3 Does the Retriever Use the Distinguishing Detail?

The retriever's success on confusable pairs comes mostly from the content the two moments share, and the distinguishing detail is decisive in fewer than half of the cases where it is needed. For each pair, we prompt an LLM to write a shared query from the attributes common to the two queries, dropping the detail, and we compare it with each original query under the same retriever and pairs as above. For example, the shared query "Lloyd and Harry walk together continuously." matches both moments of Figure 1 (Figure 4b), the snowy street and the desert highway, and only restoring the setting should send each query to its correct moment. A query prefers its correct moment when its similarity to its own oracle chunk exceeds that to the confusable moment's oracle chunk, and we call this difference the margin. Since the shared query fits both moments, its preference says nothing about the correct one, and an ideal retriever would let the detail move every original query to its correct moment. Instead, when the shared query prefers the confusable moment, the detail moves only 47.6% of queries to the correct one (Figure 4b), and when it already prefers the correct moment, the detail leaves 94.2% of choices unchanged, with the full distribution in Appendix C.2. The drop in Section 3.2 thus comes from similarity following the shared content, so recovering the detail calls for checking it against the video rather than reweighting similarity.

3.4 Can Simple Retrieval Fixes Close the Gap?

Fine-tuning the retriever with auto-generated training data. The first fix fine-tunes Qwen3-VL-Embedding 2B on 5,730 queries in 2,865 confusable pairs, auto-generated without human review by our benchmark pipeline (Appendix B.1) from ALLVB 64 movies outside the CoM^2-Bench. Each pair supplies a hard negative, so for every query we pair its target moment's oracle chunk as

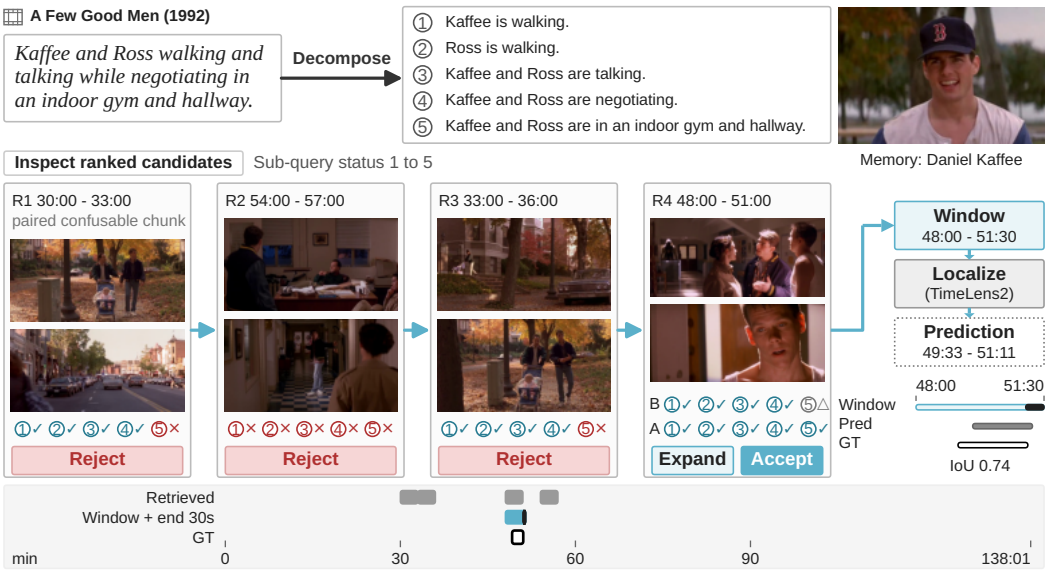


Figure 5: **Example of Agentic Window Search on the movie "A Few Good Men".** DECOMPOSE forms numbered sub-queries, and INSPECT checks each against candidate windows with a character reference, rejecting candidates that miss a sub-query, and EXPAND adjusts a window whose evidence is cut off at its edge before accepting it.

the positive with the confusable moment's oracle chunk as the negative, and a contrastive InfoNCE objective (Oord et al., 2018) with in-batch negatives trains the retriever to prefer the target chunk.

Reranking. The second fix leaves the retriever alone and adds the Qwen3-VL Reranker (Li et al., 2026). Instead of scoring a chunk and query from separate precomputed vectors, it reads the two together and returns one relevance score for each of the top five chunks, the same candidates our search later inspects. These fixes, rows 4 and 5 of Table 2, lift mIoU from 18.5 to 23.1 and 24.3, still far below the 53.8 the same localizer reaches on the oracle chunk. Both fixes still score the query as a whole, so a candidate matching the shared content can outrank one matching the detail. This calls for checking each part of the query against the candidate, which we take up in Section 4. Section 5.2 then evaluates whether combining either fix with this verification yields further gains.

4 DIRECT EVIDENCE VERIFICATION BY AGENTIC WINDOW SEARCH

The retriever folds every attribute of a query into one score, so its choice follows what a pair shares (Section 3.3). Agentic Window Search (Figure 5) instead splits the query into sub-queries and checks each against candidate windows. It inspects candidates in retrieval order, extending windows within a fixed budget (Section 4.1), and links named characters to visual references (Section 4.2). Unlike in Movie RAG, the returned window W may come from any rank and extend past chunk edges, and the unchanged localizer predicts $\hat{\tau} = [s, e]$ from q and the video restricted to W, keeping the retriever and localizer fixed with no task-specific training.

4.1 CASCADED INSPECTION WITH BOUNDED EXPANSION

Inspection checks every part of the query, not its overall similarity. A single MLLM performs DECOMPOSE, INSPECT, and SELECT through zero-shot prompting without task-specific training. Given query q, ranked chunks $C_{1:K}$, and character memory (Section 4.2), MATCHCHARACTERS produces candidates P by name lookup in memory without a model call. A single DECOMPOSE call splits q into a checklist D of sub-queries, so that the distinguishing detail is checked on its own rather than absorbed into the shared content, and selects characters from P with one reference each to form R, keeping both fixed across candidates. Because fixed chunks can cut an event at their edges (Section 3.1), candidate inspection distinguishes missing evidence from evidence cut off at a window edge. INSPECT evaluates the sampled frames and subtitles of the candidate window with the references in R and assigns each sub-query a status of true, false, or cutoff. It returns

ACCEPT when every sub-query holds somewhere in the window, REJECT for missing evidence, or EXPAND for evidence cut off at an edge. The same call chooses the expansion side and duration, and EXPANDWINDOW applies it without a model call, clamps the window to movie boundaries, and appends frames and subtitles from the added span. The budget permits at most M expansions per candidate, each up to $\Delta_{\max}$ seconds. Once the budget is exhausted, EXPAND is disabled, and any residual EXPAND response is converted to REJECT. The full pseudocode is in Appendix D.1.

Candidates are inspected in retrieval order, and the loop stops at the first ACCEPT, so a high-ranked candidate that satisfies the query ends the search early. If all candidates are rejected, a single SELECT call compares their final summaries, rejection confidences, and midpoint frames and returns the selected window. The procedure thus returns a window W for the localizer with bounded model calls. Excluding retries of malformed outputs, the worst case requires $K(M+1)$ INSPECT calls, one DECOMPOSE call, and at most one SELECT call.

4.2 CHARACTER MEMORY CONSTRUCTION

Sub-queries often name characters, such as Lloyd and Harry in Figure 1, and INSPECT can verify them only if it knows each character's appearance, which neither the query nor the window states. We therefore build character memory from movie video, subtitles, and the movie identifier, without evaluation queries or ground truth. Dialogue anchors identify names, nicknames, and forms of address through introductions, speaker identification, and direct address to participants, excluding possessive references and third-party mentions. Anchor timestamps select nearby frames for reference clips where the model links names to visible people. Comparisons use stable physical cues to merge names for the same character while preserving distinct characters and appearance variants. Memory stores names and reference frames checked for identity and quality across multiple clips. INSPECT (Section 4.1) uses these references to match a named character to a person on screen, without adding appearance constraints absent from the query or proving presence in a candidate window. Appendix D.2 gives construction details.

5 EXPERIMENTS

Setup. We instantiate the pipeline introduced in Section 3 with the `Qwen3-VL-Embedding 2B` and TimeLens2. Retrieval and inspection sample frames at 1 fps. Retrieval uses all sampled frames per chunk, together with subtitles. We use an MLLM in AWS with `Qwen3.5-4b` and `Qwen3.6-27b`, using $K = 5$ ranked candidates. Verification permits $M = 3$ expansions per candidate with $\Delta_{\max} = 30$ s per expansion. Expansion uses the same sampling rate, with at most 32 frames per expansion. Decompose and Select use temperature 0.2, while Inspect uses temperature 0.4 and top probability 0.9. Thinking is disabled for all three roles. Memory variants use the same stored character memory, while variants without memory give the verifier no character references.

Evaluation. AWS comparison keeps retrieval and localization fixed across verifier configurations. Temporal IoU is $|\hat{\tau} \cap \tau| / |\hat{\tau} \cup \tau|$, where $\hat{\tau}$ and τ are the predicted and ground truth intervals and $|\cdot|$ denotes temporal duration. We report mIoU as the mean temporal IoU over queries and R@1@θ as the fraction of queries whose single prediction achieves an IoU of at least θ, with $\theta \in \{0.3, 0.5, 0.7\}$. Both metrics are reported as percentages. Queries with missing or invalid predictions receive an IoU of zero. We also report PairSuccess@θ defined in Section 2 at the same thresholds.

5.1 MAIN RESULTS

Public models that claim long-video temporal grounding struggle on CoM^2-Bench, with UniversalVTG (An et al., 2026) highest at 16.3 mIoU and several others near zero, indicating that full-movie confusable grounding is hard rather than that these models are weak in general. Separating retrieval and localization is stronger, and Movie RAG reaches 18.5 mIoU, the best baseline here.

Holding this baseline's retriever and localizer fixed and adding no task-specific training, Agentic Window Search raises mIoU to 26.8 with the 27B verifier and to 22.5 with the 4B verifier, and R@1@0.5 from 19.1 to 28.2. AWS also improves pair success, raising PairSuccess@0.5 from 4.1% to 8.8%, so it more often grounds both confusable moments of a pair, though the low value shows that telling confusable moments apart stays hard. It closes 24% of the gap from Movie RAG to the oracle retriever reference of 53.8 (Table 2, row 6). These gains come entirely from checking

Table 3: **Direct evidence verification improves grounding and pair success without training.** Public grounding models score low on CoM^2-Bench, while Movie RAG is the strongest baseline. Checking candidate evidence raises every metric. P@θ is PairSuccess@θ, the percentage of confusable pairs whose two queries are both grounded with IoU at least θ.

Model	P@0.3 ↑	P@0.5 ↑	P@0.7 ↑	R@1@0.3 ↑	R@1@0.5 ↑	R@1@0.7 ↑	mIoU ↑
VideoChat-T (Zeng et al., 2025)	0.0	0.0	0.0	0.1	0.0	0.0	0.0
Time-R1 (Wang et al., 2025b)	0.0	0.0	0.0	0.1	0.0	0.0	0.0
UniTime (Li et al., 2025)	0.5	0.1	0.0	6.0	3.9	2.1	4.8
UniversalVTG (An et al., 2026)	5.8	2.7	1.2	22.1	14.6	8.7	16.3
TimeLens (Zhang et al., 2026b)	0.0	0.0	0.0	0.4	0.3	0.1	0.3
TimeLens2 (Zhu et al., 2026)	1.1	0.8	0.3	10.0	6.9	3.9	7.4
Movie RAG (Kim et al., 2026)	7.1	4.1	2.0	25.8	19.1	12.2	18.5
Movie RAG + AWS (Ours, 4B)	10.6	6.4	3.0	31.3	23.6	15.1	22.5
Movie RAG + AWS (Ours, 27B)	15.0	8.8	3.7	37.0	28.2	17.9	26.8

candidate evidence, with the retriever, the localizer, and their weights unchanged, so the retrieval bottleneck identified in Section 3 can be relieved without retraining any component. A single 27B pass over the top-five windows already lifts mIoU to 25.1, direct evidence checking working as the diagnosis predicts, and cascaded inspection raises it further to 26.8 (Section 5.2).

5.2 Analysis of Agentic Window Search

Table 4 ablates the Full 27B configuration one change at a time, with the retriever and localizer fixed. Removing character memory lowers mIoU from 26.8 to 26.1, and removing the cascaded inspection of Section 4.1, so that the verifier instead selects among the five candidate windows and their subtitles in a single inference, lowers it to 25.1, so both components contribute. A single inference already reads the candidate video but judges each window against the whole query, whereas decomposing the query and checking each sub-query against one window at a time keeps the distinguishing detail from being absorbed into the shared content, which accounts for the gain from 25.1 to 26.8. Removing both returns the pipeline to Movie RAG at 18.5 mIoU, so character memory and inspection together account for the gain. The retrieval fixes of Section 3.4 also help when placed before AWS, feeding the same verifier reranked or fine-tuned candidates while keeping the memory, localizer, and expansion budget fixed. Reranking reaches 28.2 mIoU (Table 4), and fine-tuning the retriever reaches 30.6, each above the fix on its own in Table 2 and above AWS alone. Reranking reorders the candidates and fine-tuning rescores them, while AWS instead checks their evidence, so the retrieval fixes and AWS are complementary. Additional retriever and localizer combinations, MomentSeeker results, and qualitative examples are in Appendices E and F.

Table 4: **Each AWS component helps, and reranking or fine-tuning the retriever adds further gains.** mIoU for a single change from the Full 27B run.

Method	mIoU
Full (27B)	26.8
Remove a component	
w/o memory	26.1
w/o inspection	25.1
w/o both (Movie RAG)	18.5
Add a component	
w/ Reranker	28.2
w/ Retriever FT	30.6

6 Conclusion

Grounding confusable movie moments requires checking the evidence that tells them apart rather than the content they share. We introduced CoM^2-Bench, a benchmark that pairs confusable movie moments and grounds each query independently across the full movie. Diagnosing a retrieve-then-localize pipeline, we found window retrieval to be the main bottleneck, and that the retriever separates a moment from its confusable partner far less reliably than from an unrelated one, because its choice follows the content the two share rather than the detail that sets them apart. Together, these results suggest the principal failure is not insufficient retrieval capacity but insufficient use of the discriminative evidence that separates otherwise similar moments. Direct verification (Agentic Window Search) substantially reduces this failure even with the retriever and localizer fixed, showing the diagnosis is actionable. Gains remain bounded, as localization stays imperfect even with the best overlapping chunk, and the low PairSuccess shows grounding both queries in a pair is still hard. We hope CoM^2-Bench guides future work that improves both window selection and boundary prediction and generalizes beyond a single retrieve-then-localize pipeline.

AI USE STATEMENT

Large language and multimodal models are components of the studied system. We use Gemini 3.1 Pro for benchmark query generation as described in Section 2 and Appendix B, Qwen3.6 27B for character memory construction in Appendix D.2, and Qwen verifiers for Agentic Window Search in Sections 4 and 5. The verifier prompts are provided in Appendix G.

ETHICS STATEMENT

Movies and subtitles are sourced from ALLVB, and LLM generated queries are human verified by three of the authors. The benchmark concerns movie content and does not target private individuals. Character memory stores reference frames and names from the movies themselves. Our base pipeline uses pretrained models without fine tuning. We release queries, timestamps, and code in the supplementary material, but do not redistribute the videos.

REPRODUCIBILITY STATEMENT

Section 2 and Appendix B describe benchmark construction and human verification. The Setup and Evaluation paragraphs in Section 5 specify models, sampling, thresholds, and metrics. Appendices C.1, D.2, and E.1 describe the input configurations of public baselines, character memory construction, and pipeline configurations. Appendix G provides the verifier prompts. The supplementary material contains the Agentic Window Search code, an evaluation script, the CoM^2-Bench annotations, and the training data of Section 3.4.

REFERENCES

Joungbin An, Agrim Jain, and Kristen Grauman. Universalvtg: A universal and lightweight foundation model for video temporal grounding. *arXiv preprint arXiv:2604.08522*, 2026.

Kirolos Ataallah, Xiaoqian Shen, Eslam Abdelrahman, Essam Sleiman, Mingchen Zhuge, Jian Ding, Deyao Zhu, Jürgen Schmidhuber, and Mohamed Elhoseiny. Goldfish: Vision-language understanding of arbitrarily long videos. In *European Conference on Computer Vision*, pp. 251–267. Springer, 2024.

Wayner Barrios, Mattia Soldan, Alberto Mario Ceballos-Arroyo, Fabian Caba Heilbron, and Bernard Ghanem. Localizing moments in long video via multimodal guidance. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 13621–13632. IEEE, 2023.

Jianghan Chao, Wenhui Tan, Yuchong Sun, Ruihua Song, Liyun Ru, et al. Jointavbench: A benchmark for joint audio-visual reasoning evaluation. In *International Conference on Learning Representations*, volume 2026, pp. 84035–84062, 2026.

Shixing Chen, Chun-Hao Liu, Xiang Hao, Xiaohan Nie, Maxim Arap, and Raffay Hamid. Movies2scenes: Using movie metadata to learn scene representation. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6535–6544. IEEE, 2023.

Zixu Cheng, Yujiang Pu, Shaogang Gong, Parisa Kordjamshidi, and Yu Kong. Shine: Saliency-aware hierarchical negative ranking for compositional temporal grounding. In *European Conference on Computer Vision*, pp. 398–416. Springer, 2024.

Andong Deng, Zhongpai Gao, Anwesa Choudhuri, Benjamin Planche, Meng Zheng, Bin Wang, Terrence Chen, Chen Chen, and Ziyan Wu. Seq2time: Sequential knowledge transfer for video llm temporal grounding. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13766–13775. IEEE, 2025.

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

Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. TALL: Temporal activity localization via language query. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017. URL https://arxiv.org/abs/1705.02101v2.

Tiantian Geng, Jinrui Zhang, Qingni Wang, Teng Wang, Jinming Duan, and Feng Zheng. Longvale: Vision-audio-language-event benchmark towards time-aware omni-modal perception of long videos. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18959–18969. IEEE, 2025.

Google DeepMind. Gemini 3.1 pro model card, 2026. URL https://deepmind.google/models/model-cards/gemini-3-1-pro/.

Weiyu Guo, Ziyang Chen, Shaoguang Wang, Jianxiang He, Yijie Xu, Jinhui Ye, Ying Sun, and Hui Xiong. Logic-in-frames: Dynamic keyframe search via visual semantic-logical verification for long video understanding. *Advances in Neural Information Processing Systems*, 38:124389–124422, 2025a.

Yongxin Guo, Jingyu Liu, Mingda Li, Qingbin Liu, Xi Chen, and Xiaoying Tang. Trace: Temporal grounding video llm via causal event modeling. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 89594–89618, 2025b. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/df027cf11469e746ef94d583f9f5537f-Paper-Conference.pdf.

Seung Hyun Hahm, Minh T Dinh, and SouYoung Jin. Storyteller: Training-free narrative grounding for long-form audio description. *arXiv preprint arXiv:2607.11798*, 2026.

Tengda Han, Max Bain, Arsha Nagrani, Gül Varol, Weidi Xie, and Andrew Zisserman. Autoad ii: The sequel–who, when, and what in movie audio description. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 13599–13609. IEEE, 2023.

Tanveer Hannan, Md Mohaiminul Islam, Jindong Gu, Thomas Seidl, and Gedas Bertasius. Revisionllm: Recursive vision-language model for temporal grounding in hour-long videos. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19012–19022. IEEE, 2025.

Qingqiu Huang, Yu Xiong, Anyi Rao, Jiaze Wang, and Dahua Lin. Movienet: A holistic dataset for movie understanding. In *European conference on computer vision*, pp. 709–727. Springer, 2020.

Yatai Ji, Shilong Zhang, Jie Wu, Peize Sun, Weifeng Chen, Xuefeng Xiao, Sidi Yang, Yujiu Yang, and Ping Luo. Ida-vlm: towards movie understanding via id-aware large vision-language model. In *International Conference on Learning Representations*, volume 2025, pp. 52639–52652, 2025.

Dong-Hee Kim, Seonwoo Choi, Changbeen Kim, Jungmyung Wi, Juyeon Ko, Youngju Choi, Il Hyeon Mun, Hyunwoo J Kim, and Donghyun Kim. Beyond visual boundaries: Rethinking scene segmentation for movie rag. *arXiv preprint arXiv:2608.28699*, 2026.

Donghoon Lee, Jun Yeop Shim, Sun-Jae Yoon, and Chang D Yoo. Character identifying video language alignment network for weakly-supervised video-subtitle moment retrieval. In *International Conference on Pattern Recognition and Artificial Intelligence*, pp. 108–123. Springer, 2024.

Jie Lei, Licheng Yu, Mohit Bansal, and Tamara Berg. Tvqa: Localized, compositional video question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pp. 1369–1379, 2018.

Jie Lei, Licheng Yu, Tamara Berg, and Mohit Bansal. Tvqa+: Spatio-temporal grounding for video question answering. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pp. 8211–8225, 2020a.

Jie Lei, Licheng Yu, Tamara L Berg, and Mohit Bansal. Tvr: A large-scale dataset for video-subtitle moment retrieval. In *European Conference on Computer Vision*, pp. 447–463. Springer, 2020b.

Jie Lei, Tamara L. Berg, and Mohit Bansal. QVHighlights: Detecting moments and highlights in videos via natural language queries. In *Advances in Neural Information Processing Systems*, 2021. URL https://arxiv.org/abs/2107.09609v2.

Juncheng Li, Junlin Xie, Long Qian, Linchao Zhu, Siliang Tang, Fei Wu, Yi Yang, Yueting Zhuang, and Xin Eric Wang. Compositional temporal grounding with structured variational cross-graph correspondence learning. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3022–3031. IEEE, 2022.

Linjie Li, Yen-Chun Chen, Yu Cheng, Zhe Gan, Licheng Yu, and Jingjing Liu. Hero: Hierarchical encoder for video+ language omni-representation pre-training. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pp. 2046–2065, 2020.

Mingxin Li, Yanzhao Zhang, Dingkun Long, Keqin Chen, Sibo Song, Shuai Bai, Zhibo Yang, Pengjun Xie, An Yang, Dayiheng Liu, et al. Qwen3-vl-embedding and qwen3-vl-reranker: A unified framework for state-of-the-art multimodal retrieval and ranking. *arXiv preprint arXiv:2601.04720*, 2026.

Zeqian Li, Shangzhe Di, Zhonghua Zhai, Weilin Huang, Yanfeng Wang, and Weidi Xie. Universal video temporal grounding with generative multi-modal large language models. volume 38, pp. 64426–64455, 2025.

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

Ye Liu, Kevin Qinghong Lin, Chang-Wen Chen, and Mike Zheng Shou. Videomind: A chain-of-lora agent for temporal-grounded video reasoning. In *International Conference on Learning Representations*, volume 2026, pp. 57481–57506, 2026.

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

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Liangyang Ouyang, Yifei Huang, Mingfang Zhang, Caixin Kang, Ryosuke Furuta, and Yoichi Sato. Multi-speaker attention alignment for multimodal social interaction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24608–24619, 2026.

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

Anyi Rao, Linning Xu, Yu Xiong, Guodong Xu, Qingqiu Huang, Bolei Zhou, and Dahua Lin. A local-to-global approach to multi-modal movie scene segmentation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10143–10152. IEEE, 2020.

Darshana Saravanan, Varun Gupta, Darshan Singh, Zeeshan Khan, Vineet Gandhi, and Makarand Tapaswi. Velociti: Benchmarking video-language compositional reasoning with strict entailment. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18914–18924. IEEE, 2025.

Sukmin Seo and Geewook Kim. Natural-language temporal grounding in hour-long videos is a search problem: A benchmark and empirical decomposition. *arXiv preprint arXiv:2606.12300*, 2026.

Xiaoqian Shen, Wenxuan Zhang, Jun Chen, and Mohamed Elhoseiny. Vgent: Graph-based retrieval-reasoning-augmented generation for long video understanding. volume 38, pp. 100805–100830, 2025.

Mattia Soldan, Alejandro Pardo, Juan León Alcázar, Fabian Caba, Chen Zhao, Silvio Giancola, and Bernard Ghanem. Mad: A scalable dataset for language grounding in videos from movie audio descriptions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5026–5035, 2022.

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

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.

Makarand Tapaswi, Yukun Zhu, Rainer Stiefelhagen, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. Movieqa: Understanding stories in movies through question-answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4631–4640, 2016.

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

Xiaohan Wang, Yuhui Zhang, Orr Zohar, and S Videoagent Yeung-Levy. Long-form video understanding with large language model as agent. In *European Conference on Computer Vision (ECCV)*, volume 15138, pp. 58–76, 2024.

Ye Wang, Ziheng Wang, Boshen Xu, Yang Du, Kejun Lin, Zihan Xiao, Zihao Yue, Jianzhong Ju, Liang Zhang, Dingyi Yang, et al. Time-r1: Post-training large vision language model for temporal video grounding. *Advances in Neural Information Processing Systems*, 38:83330–83364, 2025b.

Ziyang Wang, Shoubin Yu, Elias Stengel-Eskin, Jaehong Yoon, Feng Cheng, Gedas Bertasius, and Mohit Bansal. Videotree: Adaptive tree-based video representation for llm reasoning on long videos. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3272–3282. IEEE, 2025c.

Junyu Xie, Tengda Han, Max Bain, Arsha Nagrani, Gül Varol, Weidi Xie, and Andrew Zisserman. Autoad-zero: A training-free framework for zero-shot audio description. In *Asian Conference on Computer Vision*, pp. 81–97. Springer, 2024.

Zhenghao Xing, Ruiyang Xu, Yuxuan Wang, Jinzheng He, Ziyang Ma, Qize Yang, Yunfei Chu, Jin Xu, Junyang Lin, Chi-Wing Fu, and Pheng-Ann Heng. Native active perception as reasoning for omni-modal understanding. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=3dLPVNhFZh.

Xinyi Xu, Hongsong Wang, Guo-Sen Xie, Caifeng Shan, and Fang Zhao. Hitea: Hierarchical temporal alignment for training-free long-video temporal grounding. In *The Fourteenth International Conference on Learning Representations*, 2026.

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

Sunjae Yoon, Ji Woo Hong, Eunseop Yoon, Dahyun Kim, Junyeong Kim, Hee Suk Yoon, and Chang D Yoo. Selective query-guided debiasing for video corpus moment retrieval. In *European Conference on Computer Vision*, pp. 185–200. Springer, 2022.

Huaying Yuan, Jian Ni, Zheng Liu, Yueze Wang, Junjie Zhou, Zhengyang Liang, Bo Zhao, Zhao Cao, Ji-Rong Wen, and Zhicheng Dou. Momentseeker: A task-oriented benchmark for long-video moment retrieval. *Advances in Neural Information Processing Systems*, 38, 2025.

Abhay Zala, Jaemin Cho, Satwik Kottur, Xilun Chen, Barlas Oguz, Yashar Mehdad, and Mohit Bansal. Hierarchical video-moment retrieval and step-captioning. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 23056–23065. IEEE, 2023.

Xiangyu Zeng, Kunchang Li, Chenting Wang, Xinhao Li, Tianxiang Jiang, Ziang Yan, Songze Li, Yansong Shi, Zhengrong Yue, Yi Wang, et al. Timesuite: Improving mllms for long video understanding via grounded tuning. In *International Conference on Learning Representations*, volume 2025, pp. 38057–38081, 2025.

Ce Zhang, Jinxi He, Katia Sycara, and Yaqi Xie. Lens: Adaptive spatio-temporal zooming for keyframe sampling in long-form videos. *arXiv preprint arXiv:2607.25125*, 2026a.

Hao Zhang, Aixin Sun, Wei Jing, Guoshun Nan, Liangli Zhen, Joey Tianyi Zhou, and Rick Siow Mong Goh. Video corpus moment retrieval with contrastive learning. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 685–695, 2021.

Jun Zhang, Teng Wang, Yuying Ge, Yixiao Ge, Xinhao Li, and Limin Wang. Timelens: Rethinking video temporal grounding with multimodal llms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10419–10429, 2026b.

Junjie Zhou, Ke Mei, Lei Li, Tianyi Wang, Fengyun Rao, and Jing Lyu. Wemm-embedding: Wechat multi-modal embedding technical report. 2026. URL https://arxiv.org/abs/2608.24053.

Yuhan Zhu, Changlian Ma, Xiangyu Zeng, Xinhao Li, Zhiqiu Zhang, Songze Li, Jun Zhang, Tianxiang Jiang, Yuandong Yang, Ziang Yan, et al. Timelens2: Generalist video temporal grounding with multimodal llms. *arXiv preprint arXiv:2607.17423*, 2026.

Yuanhao Zou, Arthad Kulkarni, Lucas Tonanez, Lincoln Spencer, Guangyu Sun, Tianxingjian Ding, Andong Deng, Yi Li, Shuangjun Liu, Yuan Li, et al. Conditional multi-event temporal grounding in long-form video. *arXiv preprint arXiv:2606.15320*, 2026.

APPENDIX CONTENTS

A Related Work **15**

B Benchmark Construction and Verification **16**

B.1 Query Pair Generation 16

B.2 Human Verification 17

C Baseline and Retriever Details **19**

C.1 Input Configurations of Public Grounding Models 19

C.2 Retriever Analysis 19

D Agentic Window Search Details **19**

D.1 Cascaded Inspection 19

D.2 Character Memory Construction 20

E Additional Experiments **21**

E.1 Robustness to the Retriever and Localizer 21

E.2 Generalization to MomentSeeker 22

F Additional Qualitative Examples **23**

G Prompts **23**

A RELATED WORK

Temporal Grounding Benchmarks. Benchmarks cover movie grounding, localized television question answering, and long video comprehension (Soldan et al., 2022; Lei et al., 2018; 2020a;b; Tapaswi et al., 2016; Wang et al., 2025a), alongside audiovisual temporal understanding (Geng et al., 2025; Chao et al., 2026). Evaluation also tests binding within videos and grounding under compositional event conditions (Saravanan et al., 2025; Zou et al., 2026). CoMET (Zou et al., 2026) builds queries with an MLLM from sampled temporal and spatial attribute combinations, then grounds all matching events in long videos across multiple domains. Some benchmark construction pipelines remove repeated events and separate search from localization in their analysis (Seo & Kim, 2026). To complement binding and event condition tests, our benchmark pairs confusable queries for full movie search, grounding each independently without the alternative query as input, so that success requires telling the paired moments apart.

Video Grounding and Agentic Search. Grounding methods develop hierarchical retrieval, compositional alignment, and negative ranking (Zala et al., 2023; Li et al., 2022; Cheng et al., 2024), as well as temporal representations and grounding supervision (Guo et al., 2025b; Zeng et al., 2025; Deng et al., 2025; Wang et al., 2025b; Zhang et al., 2026b; Li et al., 2025; Zhu et al., 2026; An et al., 2026). Search can narrow candidate windows through recursive localization, hierarchical alignment, or audiovisual guidance (Hannan et al., 2025; Xu et al., 2026; Barrios et al., 2023). Movie segmentation and scene representations organize retrieval units (Huang et al., 2020; Rao et al., 2020; Chen et al., 2023). Movie RAG (Kim et al., 2026) partitions movies into chunks, retrieves scenes by query similarity using the Qwen3-VL-Embedding (Li et al., 2026) 2B model, and passes candidates to a temporal localizer. We retain this structure and insert window search between retrieval and localization. Agents gather evidence through tools and retrieval (Wang et al., 2024; Fan et al., 2024; Ma et al., 2025; Ataallah et al., 2024; Shen et al., 2025), adapt observations to queries (Wang et al.,

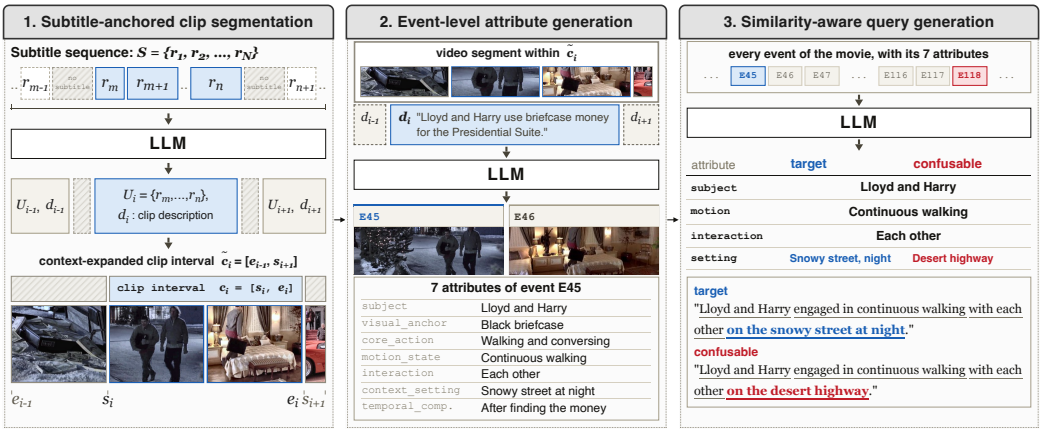


Figure A: **Dataset construction pipeline.** Subtitle-anchored clip segmentation and description generation are followed by context expansion, event and attribute generation, and confusable query pair generation. The figure uses R and C_i for the subtitle sequence and cue groups, denoted S and U_i in the text.

2025c; Yang et al., 2025; Lin et al., 2026; Zhang et al., 2026a; Xing et al., 2026), and verify temporal candidates or decomposed cues (Liu et al., 2026; Guo et al., 2025a). Complementary methods align subtitles and video for retrieval (Lei et al., 2020b; Li et al., 2020; Zhang et al., 2021; Yoon et al., 2022), link character references and speakers to visual evidence (Lei et al., 2020a; Lee et al., 2024; Ouyang et al., 2026), and preserve identity or narrative context through memory and character prompting (Song et al., 2024; Han et al., 2023; Xie et al., 2024; Ji et al., 2025; Hahm et al., 2026). We combine dialogue-anchored character references with candidate inspection and bounded expansion to distinguish confusable moments without task-specific training. The agent checks evidence and adjusts windows before passing its selection to a fixed localizer.

B Benchmark Construction and Verification

B.1 Query Pair Generation

We source the 46 full-length movies from ALLVB (Tan et al., 2025), whose official release provides movie files for research use.

We construct CoM^2-Bench using Gemini 3.1 Pro (Google DeepMind, 2026) for query generation through three stages, namely subtitle-anchored clip segmentation, event-level attribute generation, confusable query pair generation.

Subtitle-anchored clip segmentation. We first represent each movie subtitle as an ordered sequence of subtitle cues. A subtitle cue is the smallest temporal unit provided by the subtitle file and consists of an index, a timestamp interval, and its corresponding text. We denote the k-th subtitle cue as r_k. For example, a subtitle entry is represented as a subtitle cue r_{169} containing its corresponding timestamp interval and text. Given an ordered subtitle sequence $S = \{r_1, r_2, \ldots, r_N\}$, we employ an LLM-based scene segmentation process to group consecutive subtitle cues into semantically coherent clips. The LLM analyzes the subtitle content and temporal continuity of adjacent cues, merging cues that belong to the same scene, conversation, or continuous narrative episode into a single clip. Specifically, the i-th subtitle-derived clip is defined as a contiguous group of cues $U_i = \{r_m, r_{m+1}, \ldots, r_n\}$, where m and n denote the first and last cue indices assigned to the clip. The temporal interval of the clip is determined by the start time of the first cue and the end time of the last cue, i.e., $c_i = [t^s_m, t^e_n]$. Along with the clip boundaries, the LLM generates a concise natural-language description d_i for each clip, summarizing the main action or narrative content expressed by its grouped subtitle cues. Thus, each subtitle-derived clip c_i is paired with a corresponding clip description d_i, which is subsequently used as semantic context during event-level processing. After obtaining subtitle-derived clips, we construct a context-expanded clip video for each clip. Given

an original clip $c_i = [s_i, e_i]$, we define its expanded interval as $\tilde{c}_i = [e_{i-1}, s_{i+1}]$, where the preceding and following clip boundaries provide additional visual context. For the first and last clips, the movie boundaries are used when a preceding or following clip is unavailable. The original clip interval c_i is preserved inside the expanded interval $\tilde{c}_i$, together with its relative temporal position. Therefore, the expanded video contains the subtitle-free visual context immediately before and after the original clip without absorbing the neighboring clips themselves.

Event-level attribute generation. For each clip c_i, the multimodal model is provided with the context-expanded clip video $\tilde{c}_i$, the location of the original clip interval c_i within it, and the subtitle-derived descriptions of the preceding, current, and following clips, d_{i-1}, d_i, and d_{i+1}, respectively. The clip descriptions provide high-level narrative context for interpreting the visual content and its relation to neighboring clips, while the event boundaries and attributes are grounded in the video. Events are primarily generated from the visual content within the original clip interval. If the original clip contains multiple visually separable events, it is decomposed into event-level temporal segments. Since subtitle-derived clip boundaries may not exactly align with actual visual event boundaries, the surrounding visual context is used to refine the temporal extent of events and to identify additional visually meaningful events near clip boundaries. For the subtitle-free gap preceding the original clip, the preceding clip description d_{i-1} helps determine whether the observed content belongs to the preceding clip. Content associated with the preceding clip is excluded to avoid duplication, whereas a visually meaningful event that is neither associated with the preceding clip nor part of the current clip event is generated as an additional event for the current clip. In contrast, the visual context after the original clip interval is utilized as additional information for understanding the original clip, while newly occurring visually separable events within this context are not added to the current clip. These events are assigned to the corresponding following clip to prevent duplicate event generation across adjacent context-expanded clip videos.

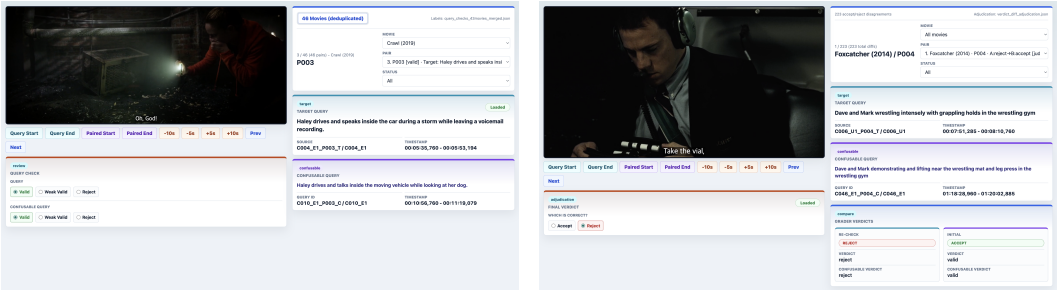
For each event, we generate attributes belonging to three general semantic families. Visual information represents object-level and scene-level visual cues associated with the event. Action information describes the observable behaviors and motions within the event. Temporal-relation information captures the dependencies and temporal relationships between the event and its surrounding contexts. In implementation, these semantic families are represented using seven structured slots, namely subject, visual anchor, core action, motion state, interaction cause, context setting, and temporal composition. Specifically, subject, visual anchor, and context setting constitute Visual information, while core action and motion state constitute Action information. The interaction/cause and temporal composition slots represent Temporal-relation information, capturing the cause of the event and its temporal relationship with nearby events. The resulting representation provides a structured basis for comparing events across the entire movie.

Confusable query pair generation. We next identify confusable event pairs based on the event-level attributes. We select event attributes from temporally distant clips to construct confusable event pairs. For every retained pair, we select exactly three shared attributes and one discriminative attribute. Shared attributes must be semantically compatible for both events and appear independently in each event. The discriminative attribute captures the principal difference needed to distinguish the target from its confusable counterpart. We then generate two parallel descriptive queries using the same four attribute roles. The queries are written as direct event descriptions rather than retrieval instructions or questions. This parallel construction ensures that the principal linguistic difference between the two queries reflects the intended event-level contrast rather than incidental variation in sentence structure. Finally, each pair is converted into temporal-grounding instances. Each instance consists of a target query q_t, a paired confusable query q_c, and their corresponding temporal intervals τ_t and τ_c.

B.2 Human Verification

Reviewers used a web interface that plays the movie with subtitles, shows both queries with their proposed intervals and structured attributes, and allows jumping to interval boundaries before recording a decision (Figure B).

The acceptance criteria are shown in Figure C. Reviewers judge the two members of each pair independently, and a pair is retained only when both queries are accepted, otherwise the whole pair



(a) Initial verification and second-pass review. (b) Adjudication of disagreements.

Figure B: **Verification interface.** (a) The target query and its confusable counterpart each receive a separate verdict, both in the initial verification and in the later review by another annotator. (b) For pairs whose acceptance outcomes disagree, the two preceding verdicts are displayed together for the resolving decision.

Annotator guideline

Accept an annotation only if all of the following are satisfied.

1. **Query validity.** The event or content described by the query clearly occurs in the video and is unambiguous.
2. **Temporal relevance.** The content described by the query is clearly visible or can be confidently inferred within the annotated GT interval.
3. **Temporal accuracy.** The GT interval tightly covers the relevant event without substantial irrelevant content or missing important portions.
4. **Uniqueness.** There is no other substantially different temporal interval that is equally valid for the same query.
5. **Evaluation validity.** A temporal grounding prediction that exactly matches the GT interval ($\mathrm{IoU} = 1$) should clearly represent a more accurate localization of the query than predictions outside or substantially shifted from the GT interval.

Reject the annotation if any of these conditions are not clearly satisfied. When uncertain or when multiple intervals are equally plausible, choose `reject`.

Figure C: **Annotator guideline.** Acceptance criteria shown to reviewers during human verification.

is removed. Reviewers also confirm that the two queries form a meaningful confusable pair, sharing the intended attributes while remaining distinguishable by the discriminative one. Of the 4,080 queries automatically generated across the 46 movies, 3,014 were retained after verification. Three annotators divided the 46 movies among themselves for the initial verification. Each annotator then reviewed six movies assigned to another annotator without access to the initial decisions, with B reviewing A's movies, C reviewing B's movies, and A reviewing C's movies, covering 18 movies in total. For binary agreement, `valid` is treated as acceptance. Inter-annotator agreement on pair acceptance before adjudication, measured by Cohen's κ, was 0.771, indicating substantial agreement on pair acceptance rather than independently assigned temporal boundaries. Disagreements were resolved by the third annotator who had not provided either preceding judgment. Annotators judged each query against the acceptance criteria (Figure C) rather than assigning new boundaries. Table A reports agreement before adjudication.

Table A: **Reviewers agree on pair acceptance before adjudication.** Agreement before adjudication by annotator assignment, where arrows mark the initial reviewer followed by the reviewer performing the blind check.

Reviewer assignment	Movies	Compared pairs	Agreement (%)	κ	Disagreements
A $\rightarrow$ B	6	282	91.1	0.775	25
B $\rightarrow$ C	6	269	90.0	0.778	27
C $\rightarrow$ A	6	259	90.3	0.757	25
Overall	18	810	90.5	0.771	77

Table B: **Confusable gold selection by contrast attribute.** Accuracy is reported in percent. Both retrievers use the same full 3,014 query set sampled at one frame per second. Each N entry lists the Qwen3-VL-Embedding and WeMM-Embedding query counts in that order, separated by a slash. All attribute groups are available for both retrievers.

Contrast attribute	N	Qwen3-VL-Embedding	WeMM-Embedding
setting	1,310 / 1,310	77.3	77.8
subject	382 / 382	66.8	67.5
core action	566 / 566	66.1	65.9
motion	212 / 212	56.6	61.3
interaction	196 / 196	67.9	69.4
visual anchor	336 / 336	69.6	70.2
temporal composition	12 / 12	66.7	58.3

C Baseline and Retriever Details

C.1 Input Configurations of Public Grounding Models

Every baseline receives the complete movie. The released code of Time-R1 and TimeLens-8B cap a video at 768 frames, and that of TimeLens2-8B at 2,048 frames, which would sample a median-length movie only once every 9 or 3.4 seconds. We therefore raise the frame limit of these three models to 12,000 frames, which no movie in CoM^2-Bench reaches, so that each of them samples the full movie at 1 fps. VideoChat-T cannot be extended in this way, because its released checkpoint fixes the length of its visual input; it therefore samples uniformly over the full movie within that fixed length. UniTime extracts features from 2,048 uniformly sampled frames and runs its released coarse-to-fine procedure, selecting a temporal window from the full movie before localizing within it. UniversalVTG operates on cached full-movie features extracted at 1 fps.

C.2 Retriever Analysis

We break down confusable gold selection by contrast attribute for Qwen3-VL-Embedding and WeMM-Embedding (Zhou et al., 2026) on the same full set of 3,014 queries, sampled at one frame per second under the Random and Confusable definitions of Section 3.2, in Table B.

Attribute groups differ in sample size and movie composition, so these descriptive results do not isolate the effects of individual attributes.

Figure D extends the shared-query analysis of Section 3.3, which Figure 4b illustrates with one pair, by showing the margin of every query under the shared and original queries.

D Agentic Window Search Details

D.1 Cascaded Inspection

Algorithm 1 states the full cascaded inspection procedure that Section 4.1 describes, inspecting ranked candidates in order and expanding a window within a fixed budget until one is accepted or a fallback selection is made.

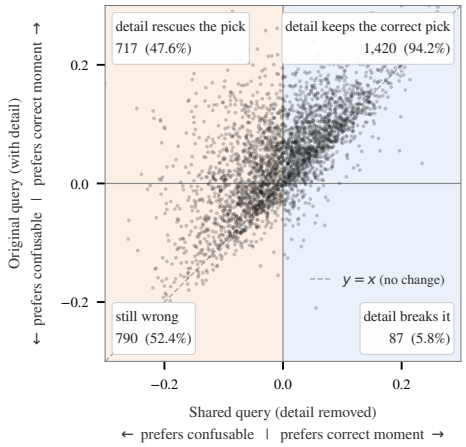


Figure D: **Shared content decides most retrieval picks.** Each point is one query, placed by its margin under the shared query (x) and the original query (y). Points stay near $y = x$, so the detail rarely changes the pick.

Algorithm 1 Cascaded Inspection with Bounded Expansion for one query. INSPECT checks each sub-query in D using fixed references R. ACCEPT returns W, REJECT advances, and EXPAND reveals adjacent evidence within budget.

Require: Query q, ranked $C_{1:K}$ ($K \geq 1$), movie, character memory
Require: At most M expansions per candidate, each at most $\Delta_{\max}$
Ensure: A window W for the downstream localizer

$P \leftarrow$ MATCHCHARACTERS(q) ▷ name lookup in memory
$(D, R) \leftarrow$ DECOMPOSE(q, P) ▷ single model call
$H \leftarrow [\,]$ ▷ rejected windows and histories
for each C_r in retrieval order **do** ▷ stop at first acceptance
  $W \leftarrow C_r$.window
  Initialize h with q, D, R and observations from C_r
  for $n = 0, \ldots, M$ **do** ▷ verify with bounded expansion
    $(o, h) \leftarrow$ INSPECT$(h, \text{allow_expand} = (n < M))$
    if $n = M$ and o.decision $=$ EXPAND **then**
      o.decision $\leftarrow$ REJECT
    end if
    if o.decision $=$ ACCEPT **then**
      return W
    else if o.decision $=$ REJECT **then**
      Append (W, h) to H
      break
    else
      $(W, h) \leftarrow$ EXPANDWINDOW$(W, h, o, \Delta_{\max})$ ▷ new band only
    end if
  end for
end for
$W \leftarrow$ SELECT(q, D, H) ▷ VLM selects among rejected
return W

D.2 CHARACTER MEMORY CONSTRUCTION

We construct character memory separately for each movie from its video, subtitles, and identifier without retrieval queries or ground truth timestamps. Qwen3.6 27B (Qwen Team, 2026), used throughout construction, extracts names such as “Alex Smith” and “Alex” from subtitle blocks overlapping by eight entries. Duplicate extractions sharing a normalized proposed name are combined, counting each supporting subtitle entry once. A text only consolidation call uses these counts and sampled contexts to select recurring characters and propose groups of names that refer to the same person. Each group’s longest name serves as its provisional display name, while visual evidence is

Table C: **Pipeline configuration comparisons.** Baseline and Ours use the same retriever and localizer, and Gain denotes the change in mIoU. Ours uses the 27B verifier with character memory and the corresponding retrieval cache.

Retriever	Localizer	Baseline mIoU	Ours mIoU	Gain	Baseline R@1@0.5	Ours R@1@0.5
Qwen3-VL-Embedding	TimeLens2	18.5	26.8	+8.3	19.1	28.2
WeMM-Embedding	TimeLens2	17.9	26.8	+8.9	18.8	28.3
Qwen3-VL-Embedding	UniversalVTG	15.8	23.4	+7.6	15.3	22.7

collected separately for every name. Candidate name pairs come from these groups, unique matches between short and full names, or another text only call suggesting equivalence from dialogue. For each name, matching subtitle entries are candidates for video inspection. We shortlist up to 40 entries, combining lexical heuristic scores with temporal coverage. Using surrounding dialogue, the model excludes mentions classified as references to third parties or possessive references and prioritizes statements identifying the speaker, introductions, and direct address suggesting participation. We choose up to ten entries per name, at least 45 s apart, adding eligible weaker mentions if fewer than two strong candidates qualify. Their timestamps determine which video intervals to inspect.

Each selected timestamp centers a 30 s reference clip with overlapping subtitles, uniformly sampled at 1 fps. One independent call per reference clip links the name to a visible person and selects at most two frames. When at least two reference clips yield candidates, one call jointly compares all selected frames using stable physical cues and identifies reference clips consistent with one character. References must pass confidence checks for the name and identity and quality gates for each frame. Frame identity confidence is the minimum of grounding scores and consistency scores across reference clips. Pairwise comparison requires accepted references for both names. Comparison decisions distinguish the same person in the same or different visual forms from different people, or indicate uncertainty.

Identity resolution uses two stages of constrained union find. With a confidence threshold of 0.80, comparison decisions identifying different people first establish constraints that prohibit merging their names. The first stage merges name groups supported by comparison decisions identifying the same person unless a prohibitive constraint connects their members. Within each character group, the second stage starts with each name separate. Accepted comparison decisions identifying different visual forms prohibit variant merges, while accepted comparison decisions identifying the same form join groups only when these constraints are respected. Remaining groups define variants. Conflicting merges of the same form are skipped, and results depend on processing order, which is not fixed. Each variant pools accepted references and is retained only with support from at least two reference clips, storing up to four references in $\mathcal{M}$. Figure E shows character memory examples from two movies.

E Additional Experiments

This appendix reports experiments beyond the main setup of Section 5, first varying the retriever and localizer and then testing transfer to another dataset.

E.1 Robustness to the Retriever and Localizer

The main results fix Qwen3-VL-Embedding and TimeLens2. Table C instead varies these two components, and AWS keeps a comparable gain when the retriever is changed to WeMM-Embedding.

Both AWS configurations use the same 3,014 queries and their respective retrieval rankings. A retriever change on the same benchmark is not external generalization.

The fine tuned retriever, reranking, and their combination with AWS are covered in Sections 3.4 and 5.2. Table D additionally reports the base pipeline with a larger Qwen3-VL-Embedding 8B retriever in place of the 2B used by Movie RAG, both with TimeLens2 and without AWS, under the same evaluation protocol as the main results. The public grounding models appear in Table 3.

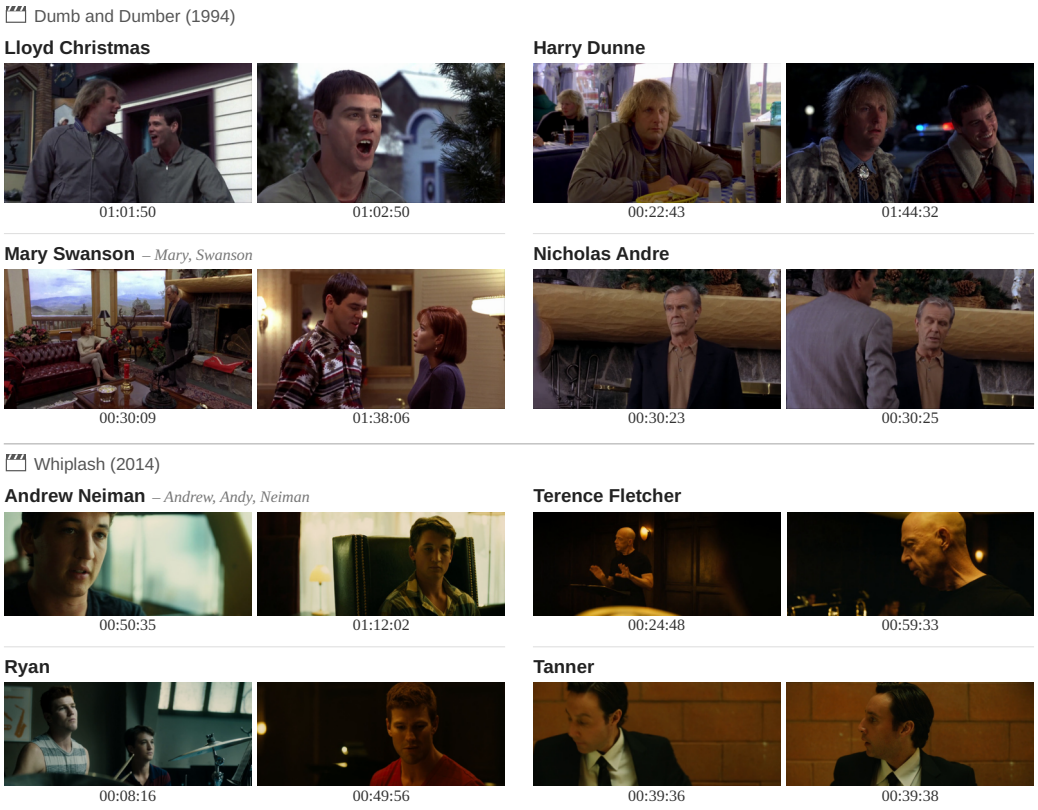


Figure E: **Stored character memory examples.** Entries from two movies with two reference frames each and their timestamps. Display names use credited character names and the smaller italic names reproduce the other stored name records.

Table D: **A larger retriever gives only a small gain.** Replacing the 2B retriever of Movie RAG with a larger Qwen3-VL-Embedding 8B, both with TimeLens2 and without AWS, raises mIoU only from 18.5 to 20.1, so retriever scale alone does not close the gap. R denotes R@1 and P denotes PairSuccess.

Retriever (with TimeLens2)	mIoU	R@0.3	R@0.5	R@0.7	P@0.5
Qwen3-VL-Embedding 2B (Movie RAG)	18.5	25.8	19.1	12.2	4.1
Qwen3-VL-Embedding 8B	20.1	28.4	20.9	13.0	4.9

E.2 Generalization to MomentSeeker

We evaluate whether AWS transfers to text only queries on MomentSeeker (Yuan et al., 2025) without task specific training. The Movie RAG style baseline and Ours use the same retriever and TimeLens2 localizer without character memory. We use our pipeline setting of 180 s chunks, which differs from MomentSeeker's 10 s chunk protocol. We report the same grounding metrics as the main results, R@1 at 0.3, 0.5, and 0.7 and mIoU, counting a prediction as correct when it matches any ground truth interval at the specified IoU threshold following MomentSeeker. PairSuccess does not apply because MomentSeeker has no confusable pairs. AWS transfers to this benchmark, raising mIoU from 19.9 to 24.8 with the 27B verifier (Table E).

Table E: **Generalization to MomentSeeker (Yuan et al., 2025).** Evaluation uses text only queries. Movie RAG and Ours use 180 s chunks with the retriever and TimeLens2 fixed and no character memory, which differs from MomentSeeker's 10 s chunk protocol. Grounding metrics follow the main results, and PairSuccess is omitted because MomentSeeker has no confusable pairs.

Method	mIoU ↑	R@1@0.3 ↑	R@1@0.5 ↑	R@1@0.7 ↑
Movie RAG	19.9	29.6	15.7	6.9
Movie RAG + AWS (4B)	23.4	34.8	19.5	8.3
Movie RAG + AWS (27B)	24.8	37.2	20.8	8.8

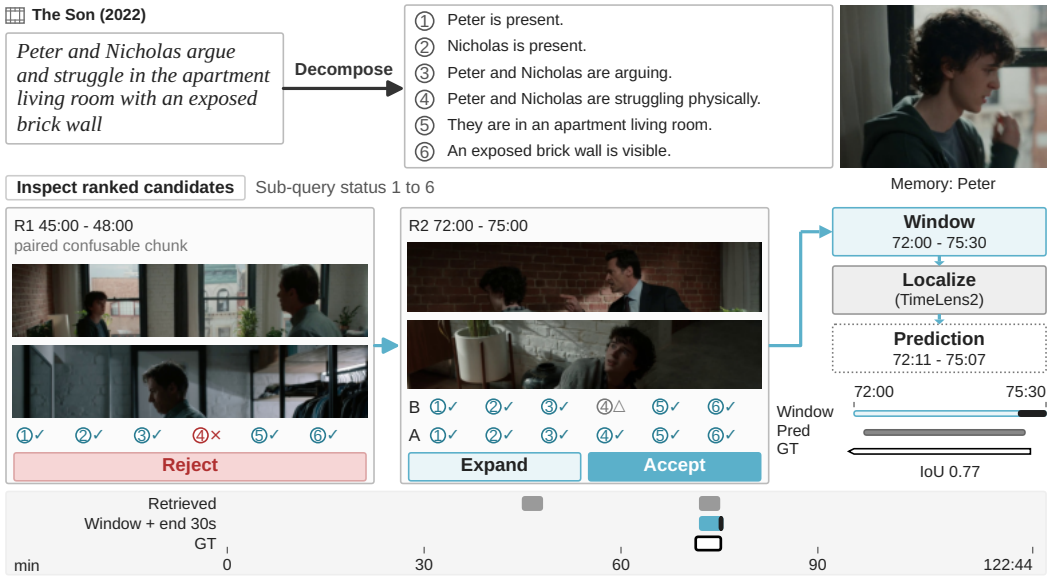


Figure F: **Qualitative example with bounded expansion.** The paired confusable chunk is rejected because the physical struggle sub-query is false, while a 30 s rightward expansion of the next candidate changes that sub-query from cutoff to true and leads to acceptance. Numbered chips show the recorded model status for each sub-query. Check marks, crosses, and triangles denote sub-queries judged true, false, or cut off at the window boundary, and B and A rows show statuses before and after expansion.

F Additional Qualitative Examples

Figure F illustrates a model-generated status change after bounded expansion in The Son. It explains the procedure without establishing a quantitative expansion effect.

G Prompts

These system prompts and user templates are used by the zero shot verifier for Decompose, Inspect, and Select. Inspect is named Look in the implementation, and EMIT corresponds to acceptance. Braced fields in user templates denote runtime substitutions, while JSON braces in system prompts are literal. Image content is attached separately at the positions described below. The serialized sub-queries exclude the internal character reference image field. Optional subtitle blocks are empty when no subtitles are available. Inspect retains prior model replies and expansion observations in its conversation history.

Decompose splits the query and selects character references using `query`, `cid`, `display_name`, and `j`, repeating candidate names and thumbnail labels with each image following its label.

Decompose

```
System prompt
You split a scene description into atomic statements.

Each statement will be checked against a stretch of video by asking:
"is this true at any moment in here?" So each statement must describe
 something observable at a moment -- a person, an action, a place
that is on screen, a line that is spoken.

Rules:
- Split anything that joins two claims, one claim per statement.
Examples: "A and B are talking in the kitchen" becomes "A appears" /
"B is visible" / "Two people are talking in a kitchen". "A runs down
a dark hallway" becomes "A is running" / "A is in a dark hallway".
Vary the wording; do not reuse one phrasing for every statement.
- Write about what is on screen at a moment, NOT about "the scene" as
 a whole. Write "Two people are in an underground lab", not "The
scene takes place in an underground lab". Never begin a statement
with "The scene", "The setting", "The segment", or "The video".
- Keep proper nouns exactly as written: if the description names a
person or place, use that name in the statement rather than a generic
 description (write "Kaffee is shouting", not "A man is shouting").
- State only what the description states. Do not add conditions.
- Each atomic statement must be entailed by the original description.
 Decomposing and checking the statements one by one must not weaken,
strengthen, or change the original meaning. Preserve temporal order,
co-occurrence, causality, negation, quantities, and relationships; do
 not split apart dependencies that carry meaning.
- Together the statements must cover the whole description: a moment
where every statement is TRUE should be the described moment, and a
different moment should make at least one FALSE.
- Write each statement so a similar-but-wrong moment makes it FALSE.
- Plain declarative sentences. No questions, no "should", no hedging.
- 3 to 7 statements.

The CAST below is a SEPARATE input, used only to identify people --
it must NOT change the statements. Decompose the scene description
ALONE into statements, exactly as you would with no cast. Never copy
clothing, colour, hair, or any appearance detail you see in a
thumbnail into a statement: if the scene description does not mention
 what someone is wearing, no statement may mention it either. The
thumbnails are for recognising who is who, not for describing them.

Each person has an id like [C1], a name, and one or more numbered
reference thumbnails like [C1:r1] (identity references from elsewhere
 in the movie, NOT part of any video segment). From the scene
description decide which cast members it actually refers to (ignore
the rest), and for each chosen person pick the single thumbnail that
best fits how they would look in the described moment (for example an
 armored one when the wording implies armor). Do not invent people
who are not in the cast; if the description names no one in the cast,
 return an empty cast. Keep each chosen person's name in the
statements so it lines up with their [REF] identity image at Look
time.

Reply with exactly one JSON object:
{"statements": [{"id": "s1", "text": "..."}], "cast": [{"char_id": "
C1", "name": "...", "ref": "C1:r2"}]}

User template
Scene description: "{query}"

Character candidates appended to the user template
```

```
CAST (choose only the people this description refers to, and the best
-matching thumbnail for each):
[{cid}] {display_name}
[{cid}:r{j + 1}]
```

Inspect checks each candidate using `query`, `decomposition_json`, `chunk_desc`, and `subtitle_block`, followed by video frames, with optional subtitle lines using `timestamp` and `text` and budget messages using `next_expansion`, `used`, and `remaining` for up to three expansions of at most 30 s each.

Inspect

```
System prompt
You are a temporal video-retrieval VERIFIER. You are given a
checklist of atomic statements describing a target moment, and one
candidate video segment sampled at 1 fps -- each frame is preceded by
 its absolute movie-time marker "<T.T seconds>". Decide whether that
moment occurs in the currently visible window.

Work in three steps, in this exact order:

STEP 1 -- observe each statement. For EVERY statement, FIRST write a
short "note" (under 20 words) describing what you actually saw in the
 frames, THEN set "holds" -- the verdict must follow from the note
you just wrote:
- "true": it holds at some moment within the visible frames.
- "false": nothing in the visible frames shows it.
- "cutoff": you cannot see it here, but the evidence runs off an EDGE
 of the window -- the relevant action/shot is already under way in
the FIRST frames, or still unfinished in the LAST frames -- so it
likely holds just outside this window.
Keep the ids.

STEP 2 -- coverage. In "cooccurrence", note roughly WHERE in the
window the "true" statements appear and whether the moment is cut off
 at an edge (evidence for a statement still running in the first or
last frames). You do NOT need the statements to line up at a single
instant -- a downstream temporal-grounding localizer pins the exact
time; here you only judge whether this segment CONTAINS the moment.

STEP 3 -- decide. Set "decision":
- "EMIT": every statement is "true" -- each holds at some moment in
the visible window. They need NOT coincide at one instant: this
segment contains the moment, and a downstream temporal-grounding
localizer refines the exact boundaries, so the visible window is
returned as-is.
- "EXPAND": one or more statements are "cutoff" -- their evidence
runs off an edge of the window. Reveal more frames on that side: set
"expand_direction" to "left" (earlier) or "right" (later) -- the side
 it runs off -- and "expand_seconds" (<= 30).
- "REJECT": a required statement is "false" -- nothing in the visible
 window shows it -- and extending the window will not fix it.

EXPANSION BUDGET:
- Each EXPAND reveals AT MOST 30 seconds of extra footage on ONE side
 (larger requests are truncated to 30s).
- You may EXPAND AT MOST 3 times in total for this segment. Once the
budget is spent you can no longer EXPAND and must EMIT or REJECT.
- A [Budget] line before your turn tells you how many expansions
remain. Expand only when a side clearly cuts off the moment.

Reply with ONLY a JSON object, keys in this order:
```

```
{
  "statements": [{"id": "s1", "note": "<=20 words, what you saw", "
holds": "true|false|cutoff"}],
  "cooccurrence": "<do the true statements coincide at one moment?
when / which edge cuts off>",
  "decision": "EMIT" | "EXPAND" | "REJECT",
  "confidence": <0.0-1.0>,
  "expand_direction": "left" | "right" | null,
  "expand_seconds": <seconds or null>
}

Rules:
- Judge every statement in the checklist. Within each statement write
 "note" before "holds"; commit to "decision" only after all
statements and cooccurrence.
- confidence is required (how sure you are of the decision).
- expand_direction ("left"/"right") and expand_seconds (<= 30) are
REQUIRED only when decision is EXPAND. Choose the side the moment is
cut off on.
- Ground all timestamps in the "<T.T seconds>" markers that precede
each frame.

You may also be given REFERENCE images of named people, each preceded
 by a line "[REF: <name>]" and followed by that person's photo. These
 are identity references from elsewhere in the movie; they are NOT
part of the candidate window and carry NO timestamp. Use them only to
 recognise who is who. Never treat a reference as evidence that a
statement holds or that the moment occurs -- judge solely from the
timestamped candidate frames. Being named does not mean the person is
 present; confirm it in the frames. A person may look different from
their reference (clothing, angle, lighting, or costume such as armor)
; judge identity from stable facial features, not from clothing or
outfit.

You may also be given the segment's SUBTITLES with timestamps -- the
dialogue spoken in the visible window. Use them to tell who is named
or speaking, but a statement is "true" only if the frames confirm it:
 a name in the subtitles is not proof the person is on screen.

User template
Query: "{query}"

Statements the target moment must all satisfy:
{decomposition_json}

Candidate segment: {chunk_desc}
{subtitle_block}The following is that segment as a video sampled at 1
 fps (frames are in order, one per second of movie time). Locate
moments by their absolute movie time using the segment's start time
stated above. Decide EMIT / EXPAND / REJECT.

Optional subtitle template
Subtitles spoken in this window (absolute timestamps):
[t={timestamp:.1f}s] {text}


Available budget message
[Budget] Expansion {next_expansion} of 3 available ({used} used, {
remaining} remaining). Each EXPAND reveals up to 30s on one side. You
 may EXPAND, EMIT, or REJECT.

Exhausted budget message
```

```
[Budget] Expansion budget exhausted ({used} of 3 used, 0 remaining).
You may no longer EXPAND -- choose EMIT or REJECT.

Message when expansion is masked
Expansion budget exhausted. You may no longer EXPAND. Judge the
statements and choose EMIT or REJECT based on what you have seen.
```

Select chooses a fallback when no candidate is accepted using `query`, `decomposition_json`, and `n`, with each candidate entry using `index`, `chunk_id`, `win_start`, `win_end`, `confidence`, and `reasoning` before its representative frame.

Select

```
System prompt
You are choosing which candidate video segment most likely contains a
 target moment. A verifier already inspected each segment and
REJECTED all of them, but one is still the closest match. For each
candidate you are given a representative frame (timestamp labelled)
and a short summary: the time window that was examined, the verifier'
s confidence, and its reasoning.

Pick the SINGLE candidate index most likely to contain -- or be
closest to -- the target moment. Weigh the summaries and the frames
together; a higher verifier confidence means the rejection was more
certain (i.e. that segment is LESS likely to be the answer).

Return ONLY a JSON object:
{
  "selected_index": <integer index of the chosen candidate>,
  "reasoning": "<why this candidate is the closest>"
}

User template
Target query: "{query}"

Statements the target moment must all satisfy:
{decomposition_json}

There are {n} candidate segments below, none confirmed. Each is
followed by its representative frame.

Repeated candidate template
[{index}] chunk {chunk_id}: examined window [{win_start:.0f}s, {
win_end:.0f}s], verdict=REJECT, confidence={confidence:.2f}
    verifier reasoning: {reasoning}

Closing instruction
Pick the single index most likely to contain (or be closest to) the
target moment.
```