BEYOND ISOLATED CLIPS: BENCHMARKING INTER-SEGMENT NARRATIVE CONTINUITY IN MOVIE-LEVEL VIDEO UNDERSTANDING

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent advancements in multimodal large language models (MLLMs) have extended video understanding from short clips to long-form cinematic content. However, existing benchmarks predominantly evaluate isolated video segments, overlooking the narrative continuity essential to full-length movies, such as cross-scene information transfer, character consistency, and plot foreshadowing. To bridge this gap, we introduce **CineScriptBench**, the first movie-level benchmark explicitly designed to evaluate inter-segment historical consistency in cinematic script generation. Comprising 73 full-length movies (153 hours), 5,524 scenes, and 56,993 fine-grained evaluation rubrics, CineScriptBench is constructed via a novel five-stage automated pipeline that injects rolling historical context into scene-level annotations. We propose a comprehensive evaluation protocol measuring both temporal localization and content accuracy, featuring a dedicated dimension for historical consistency via an LLM-as-a-Judge framework. Extensive evaluations of frontier MLLMs reveal severe limitations in long-form narrative understanding. Our findings highlight the critical gap between current MLLM capabilities and true movie-level comprehension, establishing a rigorous baseline for future research in cinematic narrative reasoning.

1 INTRODUCTION

Recent advancements in multimodal large language models (MLLMs) (Bai et al., 2025; Liu et al., 2023; Team, 2026; Xu et al., 2025) have significantly improved video understanding. The field is currently expanding from short duration clips toward long duration cinematic content (Fu et al., 2025; Zhou et al., 2025; Shaar et al., 2026). Accordingly, evaluations of frontier models have shifted from basic captioning and question answering tasks to the more complex challenge of generating detailed, structured scripts for full length movies (Pu et al., 2026; Yao et al., 2026; Wu, 2026; Wu et al., 2025).

For instance, TimeChat-Captioner (Yao et al., 2026) proposes the OmniDenseCaptioning task, which generates fine-grained, temporally grounded descriptions to support comprehensive audiovisual understanding. Building on this, OmniScript (Pu et al., 2026) introduces the V2S task, aimed at producing hierarchical, sequential scene scripts that encompass character actions, dialogues, expressions, and audio cues. Critically, OmniScript extends its evaluation to video clips ranging from 5 to 40 minutes, reflecting a clear transition beyond the short clips typical of earlier benchmarks.

Despite these extensions in duration and the refinement of structured script annotations, current benchmarks still evaluate isolated video segments independently. They fail to consider the informational continuity across an entire movie narrative, including the transmission of information between non-adjacent scenes, character consistency across different parts of the story, plot foreshadowing, and the coherence of historical context. Although current frontier models possess the capacity to process complete movie videos, context window constraints often lead to a significant loss of detailed script information. Such inter-segment dependencies are essential for a comprehensive understanding of cinematic content, yet they remain largely overlooked in existing evaluation frameworks.

To address these limitations, we introduce **CineScriptBench**, a movie-level video understanding benchmark that explicitly evaluates inter-segment information transfer. CineScriptBench provides

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.

timestamp-anchored, scene-level dense captions that capture fine-grained elements including characters, scenes, and events, enabling a comprehensive representation of cinematic narratives. Unlike existing benchmarks that evaluate isolated clips, CineScriptBench performs movie-level holistic assessment, requiring models to generate and maintain accumulated historical information during the annotation of each clip to ensure consistency across the full narrative.

We design a five-stage automated data construction pipeline comprising *Movie Preprocessing*, *Independent Scene Annotation*, *Historical Information Injection*, *Annotation Refinement*, and *Evaluation Protocol Construction*. After the automated pipeline is completed, human annotators perform careful checking and refinement to ensure annotation quality. The resulting benchmark covers 73 full-length movies (totaling 153 hours), with 5524 scenes and 56993 evaluation rubrics.

For evaluation, we propose two categories of metrics: *Temporal Accuracy* and *Content Accuracy*. The former is primarily assessed using Intersection over Union (IoU) to measure temporal localization precision. The latter is divided into four sub-categories, namely *Character*, *Event*, *Scene*, and *History*, which are evaluated via an LLM-as-a-Judge protocol with fine-grained rubrics. In particular, the History sub-category explicitly measures the ability of the model to maintain consistency of historical content across inter-segment annotations.

We conduct comprehensive evaluations of representative state-of-the-art Omni and VL models on CineScriptBench. Our findings reveal a stark contrast between the models' proficiency in localized scene description and their severe deficiencies in long-form narrative comprehension. Even the most capable frontier models exhibit significant degradation in maintaining inter-segment historical consistency, frequently failing to synthesize past events into a coherent narrative trajectory. Furthermore, we observe a pronounced sensitivity to temporal granularity, where performance degrades systematically as the evaluation spans longer narrative arcs. Collectively, these results underscore a critical bottleneck in current MLLMs: the inability to retain and utilize deep narrative structures over extended cinematic contexts, highlighting a substantial gap between isolated clip understanding and true movie-level reasoning.

- We propose CineScriptBench, the first benchmark that explicitly evaluates movie-level script generation with inter-segment historical consistency as a core dimension, providing timestamp-anchored scene-level dense annotations and holistic film-level assessment.
- We design a novel five-stage automated data construction pipeline, together with an evaluation protocol that incorporates rolling historical information injection and progressive script generation, enabling a realistic assessment of model capabilities in long-form cinematic understanding.
- We conduct comprehensive evaluations of state-of-the-art MLLMs on CineScriptBench, revealing significant shortcomings in handling long-range narrative dependencies and providing a rigorous quantitative baseline for future research in cinematic reasoning.

2 Related Work

2.1 Video Understanding Benchmarks

Early video understanding benchmarks primarily targeted short video clips and focused on discriminative tasks such as action recognition and question answering (Caba Heilbron et al., 2015; Xiao et al., 2021). As multimodal large language models (MLLMs) have advanced, the research focus has shifted toward long-form and cinematic content. Numerous benchmarks have been proposed to evaluate temporal reasoning and long-context comprehension, including MVBench (Li et al., 2024) for dynamic temporal tasks, EgoSchema (Mangalam et al., 2023) for very long-form egocentric videos, and recent comprehensive evaluations like Video-MME (Fu et al., 2025), LongVideoBench (Wu et al., 2024), and LVBench (Wang et al., 2025a), which extend assessments to hour-long videos. In the specific cinematic domain, datasets like MovieNet (Huang et al., 2020) provide large-scale multimodal annotations, while benchmarks such as ShotBench (Liu et al., 2025) and CineTech-Bench (Wang et al., 2025b) evaluate the understanding of cinematic language and shot composition. However, these benchmarks predominantly rely on multiple-choice question-answering formats or localized clip retrieval. They do not assess the generation of complete, structured scripts, thereby limiting their ability to evaluate holistic narrative generation capabilities across an entire film.

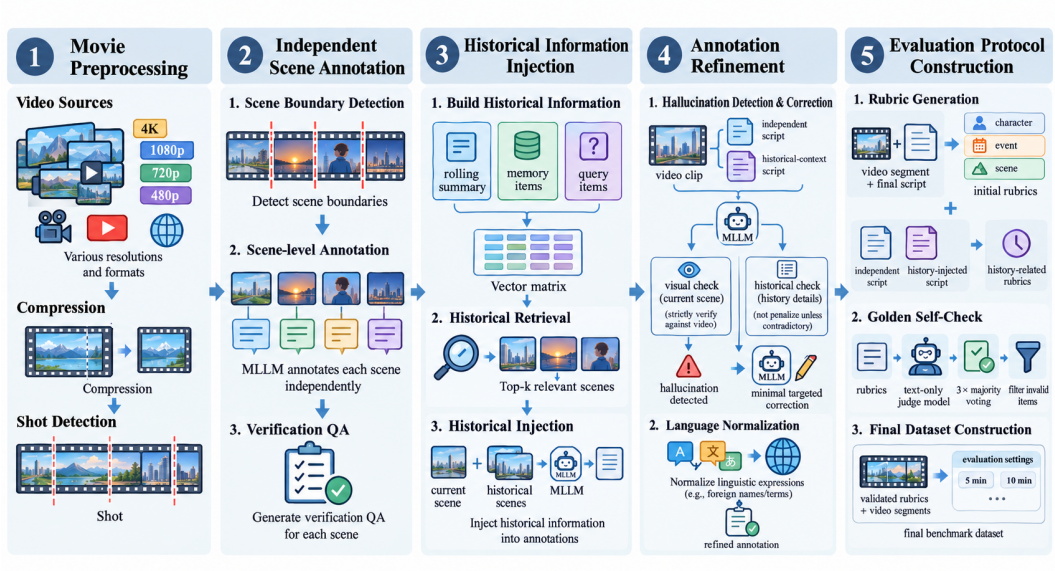

Figure 1: The five-stage automated construction pipeline of CineScriptBench.

2.2 Cinematic Dense Captioning and Script Generation

In contrast to discriminative benchmarks, another line of research explores generative video description, progressing from coarse captions to dense, structured narratives. Recent audiovisual captioning methods, such as AVoCaDO (Chen et al., 2025) and DiaDem (Chen et al., 2026), achieve multimodal fusion by integrating audio and dialogue cues, yet they typically produce global descriptions without explicit, fine-grained temporal grounding. To address this, TimeChat-Captioner (Yao et al., 2026) proposes the Omni Dense Captioning task to generate timestamped, scene-level descriptions. Furthermore, OmniScript (Pu et al., 2026) introduces the Video-to-Script (V2S) task, aiming to generate hierarchical scene scripts encompassing character actions, dialogues, and audio cues. Despite these advances in structured description, all these methods process isolated video segments rather than full-length movies. They inherently fail to account for inter-segment information transfer, character consistency, and plot foreshadowing within the broader historical context of a cinematic narrative.

3 Method

3.1 Task Formulation and Evaluation Protocol

We formulate movie-level evaluation as a progressive script generation task with rolling historical context. Given a full-length movie video V with total duration T, the goal is to generate a temporally grounded, scene-level script for the entire movie while maintaining narrative information accumulated over time.

Movie segmentation. We first apply a scene detection algorithm to V to obtain a shot sequence $\mathcal{L} = \{l_1, l_2, \ldots, l_M\}$, where each shot l_m represents a continuous cinematic unit. To construct evaluation segments, we partition the movie into clips of approximately fixed duration Δ. Instead of cutting the video at rigid uniform timestamps, each candidate boundary is snapped to the nearest shot boundary in $\mathcal{L}$. This snapping strategy prevents a complete shot from being split across segments and preserves the natural cinematic structure, resulting in a final video segment sequence $\mathcal{V} = \{V_1, V_2, \ldots, V_N\}$.

Progressive script generation. The model processes the segments $\mathcal{V}$ sequentially in chronological order. For each segment V_i, the model f_θ generates a scene-based script $\hat{Y}_i$ along with an updated rolling history $\hat{H}_i$:

$$(\hat{Y}_i, \hat{H}_i) = f_\theta(V_i, H_{i-1}), \tag{1}$$

where the initial history $H_0 = \emptyset$, and for $i > 1$, $H_{i-1} = \hat{H}_{i-1}$. Each predicted script $\hat{Y}_i$ consists of a set of scenes $\hat{Y}_i = \{\hat{y}_{i,1}, \ldots, \hat{y}_{i,K}\}$. Each scene $\hat{y}_{i,k}$ is defined by a detailed caption c and its start and end timestamps (t_s, t_e). These timestamps are constrained to the shot boundaries in $\mathcal{L}$ rather than being arbitrary continuous values. The updated history $\hat{H}_i$ summarizes key narrative elements observed up to segment i to condition future generations.

Evaluation metrics. We evaluate predictions from two perspectives: Temporal Accuracy and Content Accuracy. For temporal accuracy, we align predicted scene segments with ground-truth scene segments using a dynamic programming-based matching strategy. Let $\mathcal{M}$ denote the set of successfully matched prediction-ground-truth pairs. For each pair $(\hat{p}, g)$, the temporal intersection-over-union is computed as $\mathrm{IoU}(\hat{p}, g) = |\hat{p} \cap g|/|\hat{p} \cup g|$. We report the movie-level mean IoU by averaging the IoU over all predicted segments. Additionally, we report the matched IoU:

$$\mathrm{mIoU} = \frac{1}{|\mathcal{M}|} \sum_{(\hat{p},g)\in\mathcal{M}} \mathrm{IoU}(\hat{p}, g), \tag{2}$$

which only considers predictions successfully matched to ground-truth segments. For content accuracy, we evaluate four dimensions: character, event, scene, and history. We adopt an LLM-as-a-Judge protocol, denoted as a scoring function S_ϕ, with fine-grained rubrics to assign a score to each dimension.

3.2 Benchmark Construction Pipeline

In this section, we mainly introduce the construction pipeline as shown in Figure 1.

Movie preprocessing. We select candidate movies from the Douban and manually filter the video sources based on visual quality, audio quality, completeness, and accessibility. After this screening process, we obtain 73 raw full-length movies. To ensure consistency, we uniformly compress the raw videos to 480p resolution. For each movie video V, we apply PySceneDetect to obtain its shot sequence $\mathcal{L} = \{l_j\}_{j=1}^{M}$, where each shot is defined by a time interval $l_j = [t_j^s, t_j^e]$. These shot sequences serve as the temporal basis for subsequent segment construction and annotation.

Independent scene annotation. For each movie V, we divide the video into short fixed-duration windows W. For each window, we prompt an MLLM to determine whether a scene transition occurs. If a transition is detected, the model selects the corresponding scene boundary from $\mathcal{L}$; otherwise, the window is extended by a short interval, and the detection process is repeated until a scene boundary is identified. After obtaining all candidate boundaries, we construct a boundary-centered segment using several shots before and after each candidate boundary. The MLLM re-evaluates and refines the boundary based on this narrower context to yield the initial scene segmentation.

Each resulting scene segment is independently annotated with a detailed script. To improve annotation quality, the MLLM generates fine-grained verification questions from the video segment to check the generated script. Scripts that fail to satisfy the verification questions are refined accordingly. Finally, to prevent over-segmentation, we perform a merging pass using a sliding window over consecutive scene segments. The MLLM evaluates each adjacent pair to determine whether they should be merged based on narrative continuity, yielding the final refined scene segmentation and independent scene-level annotations.

Historical information injection. To inject inter-segment historical information into the independently annotated scripts, we construct a hierarchical historical memory for each scene segment. Let the independent script for segment V_i be C_i. The memory comprises three components: (1) a rolling summary R_{i-1}, which is a fixed-length narrative that accumulates key plot developments; (2) a set of memory items $\mathcal{K}_{i-1}$, consisting of fine-grained keywords extracted from previous captions; and (3) a set of query items $\mathcal{Q}_i$, derived by analyzing the relationship between C_i and R_{i-1} to identify missing historical context.

We embed both the memory and query items into a vector space. Processing the segments sequentially, we retrieve the top-k memory items from preceding segments that best match the query items of the current segment:

$$\mathcal{C}_i = \mathrm{TopK}(\mathcal{Q}_i, \mathcal{K}_{i-1}). \tag{3}$$

The video segments corresponding to these retrieved memory items are concatenated with the current segment and fed into an MLLM g_ϕ. The model refines the independent script based on this expanded context:

$$\hat{C}_i = g_\phi(C_i, \mathcal{C}_i, R_{i-1}), \tag{4}$$

effectively injecting the necessary historical information and ensuring inter-segment consistency.

Annotation refinement. To ensure the quality and consistency of the final annotations, we first perform a rigorous hallucination detection and correction process. For each scene segment, an MLLM is provided with the video clip alongside two versions of the script: the independent script and the final script enriched with historical context. To prevent historical information from interfering with the verification of current visual content, the evaluation is divided into two parts. Descriptions of the current scene are strictly verified against the video, while injected historical details are not penalized for being unverifiable within the current clip, unless they explicitly contradict the visual or audio evidence. If any hallucinations are detected, a secondary MLLM call is triggered to perform minimal, targeted corrections, modifying only the conflicting expressions while preserving the rest of the narrative. Subsequently, we employ language models to normalize the linguistic expressions of all generated scripts (e.g., rendering foreign-language names and terms in a unified language). This refinement pipeline yields the final, high-quality scene-level annotations.

Evaluation protocol construction. To construct fine-grained evaluation rubrics, we prompt an MLLM with the video segment and its final script to generate initial rubrics categorized into character, event, and scene dimensions. Additionally, to evaluate inter-segment historical consistency, we provide the model with both the independently annotated script and the final history-injected script to generate history-related rubrics.

To ensure the reliability and validity of these rubrics, we perform a rigorous golden self-check. We employ the same text-only judge model used in the final evaluation to verify each rubric item against the ground-truth final script. Each item is evaluated independently three times using a majority voting mechanism. Items that consistently fail the self-check are discarded as poorly formulated, while passing or undecided items are conservatively retained. This filtering process effectively eliminates invalid rubrics and reduces evaluation noise. Finally, the validated rubrics and video segments are packaged into the final benchmark dataset under various evaluation settings, such as 5-minute or 10-minute segment configurations.

4 Experiments

4.1 Experimental Setup

Evaluated Models. We evaluate a representative set of recent MLLMs, organized into two categories: *proprietary models*, including the Gemini and Qwen-Plus series, and *open-source models*, including the Qwen-Omni series together with several community-released omni models. These models are omni architectures that consume raw visual frames and audio streams as input. To further examine the contribution of the audio modality, we extend our evaluation to visual-language (VL) models, which process only visual frames without audio. Since such models cannot access spoken dialogue, we evaluate them on a dedicated subset of 17 movies that contain burned-in subtitles, ensuring that dialogue information remains available through the visual channel.

Evaluation Settings. Our main results are reported under the 5-minute segment setting: each movie is partitioned into sequential clips of approximately five minutes, with candidate boundaries snapped to the nearest shot boundaries to preserve cinematic integrity. Following the progressive protocol in Section 3.1, models process the segments in chronological order, generating a scene-level script for each segment while maintaining a rolling historical context that conditions subsequent generations. Building on this default configuration, we further conduct a series of diagnostic studies, including varying the segment duration from 5 to 20 minutes to examine the impact of temporal granularity, comparing alternative history organization patterns, and supplying ground-truth scene boundaries to disentangle temporal localization from narrative comprehension.

Table 1: Performance of frontier MLLMs on CineScriptBench under the 5-minute segment setting, grouped into proprietary, open-source, and VL models (VL models are evaluated on the 17-movie subtitle subset). The table reports temporal metrics (IoU, mIoU) and content metrics over the four dimensions together with the overall score; best results in each column are in bold.

Models	Param./B	Omni	Temporal Metrics		Content Metrics				
			IoU	mIoU	Character	Event	Scene	History	Overall
Proprietary Models									
Gemini-3-Flash-Preview	-	✓	46.10	65.19	12.08	15.64	15.77	7.96	13.59
Gemini-3.5-Flash	-	✓	52.34	72.85	15.31	23.61	17.28	8.39	18.23
Gemini-3.1-Pro-Preview	-	✓	62.97	79.41	16.91	27.09	19.37	**9.23**	20.64
Qwen-3.5-Omni-Plus	-	✓	**69.79**	83.62	21.64	31.52	29.74	6.05	24.76
Qwen-3.8-Omni-Flash	-	✓	68.70	**83.90**	**28.22**	**38.40**	**38.13**	8.53	**31.06**
Open-source Models									
Qwen-2.5-Omni	7B	✓	46.75	49.32	1.66	1.85	2.37	0.65	1.69
OmniVinci	9B	✓	53.42	56.18	2.05	2.28	3.03	0.89	2.11
MiniCPM-o-4.5	9B	✓	58.24	61.57	2.41	2.62	3.51	1.14	2.46
Qwen-3-Omni-NT	30B-A3B	✓	61.15	64.69	2.90	2.99	4.08	1.30	2.85
Qwen-3-Omni-T	30B-A3B	✓	**62.83**	**66.21**	**3.28**	**3.26**	**4.53**	**1.57**	**3.18**
VL Models (17 films subset)									
Qwen-3.6-Flash	-	✗	66.34	78.19	18.83	21.86	19.39	5.58	17.88
Qwen-3.6-Plus	-	✗	72.85	82.64	23.81	34.14	27.71	8.73	26.37
Qwen-3.8-27B	27B	✗	**80.87**	**87.42**	20.99	16.71	33.02	3.23	17.67
Qwen-3.8-Flash	125B-A6B	✗	71.19	86.53	29.40	39.45	37.70	8.00	31.37
MiniMax-M3	428B-A23B	✗	71.33	85.69	29.64	40.34	**38.60**	7.70	31.91
GLM-5.3-Flash	320B-A18B	✗	72.25	86.68	**30.46**	**41.47**	37.34	**8.08**	**32.46**

Implementation Details. For benchmark construction, the five-stage pipeline described in Section 3.2 is instantiated with Gemini-3.1-Pro-Preview as the annotation and refinement model. For evaluation, temporal accuracy is computed deterministically via the IoU-based matching protocol, whereas content accuracy is judged by Qwen-3.8-27B with fine-grained rubrics; the reliability of this judge is further verified through a dedicated meta-evaluation study. Prompt templates and detailed configuration for benchmark construction and evaluation are provided in the Appendix A.4 and A.5. Benchmark statistics information are provided in Appendix A.2.

4.2 Main Results

Table 1 reports model performance under the 5-minute segment setting. Movie-level script generation remains far from solved: even the strongest models attain an overall content accuracy of only around 31%, while temporal localization is comparatively stronger, indicating that locating scenes is easier than truly understanding them. Among the four content dimensions, History is uniformly the weakest across all models, confirming that maintaining inter-segment narrative consistency, rather than describing isolated scenes, constitutes the primary bottleneck for current MLLMs.

Across model families, proprietary omni models and large-scale VL models achieve comparable performance, and the best VL model even surpasses the best omni model on overall content accuracy, suggesting that subtitle-accessible visual evidence can largely compensate for the absence of audio. In stark contrast, open-source omni models of small to medium scale (7B–30B) exhibit markedly inferior performance and lag substantially behind proprietary counterparts across all dimensions, as their capacity to process long videos degrades sharply due to severe context window constraints. These results indicate that movie-level narrative understanding is currently accessible only to frontier-scale models, leaving a substantial capability gap for future research.

4.3 Performance Dynamics Across Temporal Quartiles

We partition each movie into four equal-duration quartiles along the temporal axis and analyze model performance trends within each segment. As shown in Figure 2, Event and Scene categories exhibit no consistent trend across Omni or VL models, suggesting that their understanding is rel-

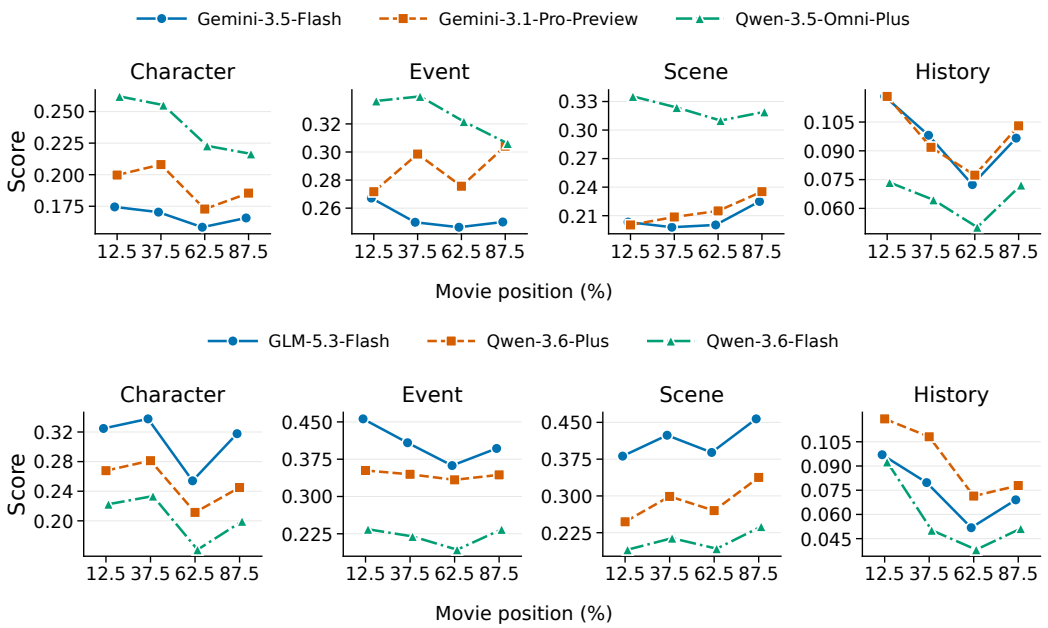

Figure 2: Content accuracy of the four evaluation dimensions across temporal quartiles (movie positions 12.5%–87.5%). Top: omni models; Bottom: VL models on the 17-movie subtitle subset.

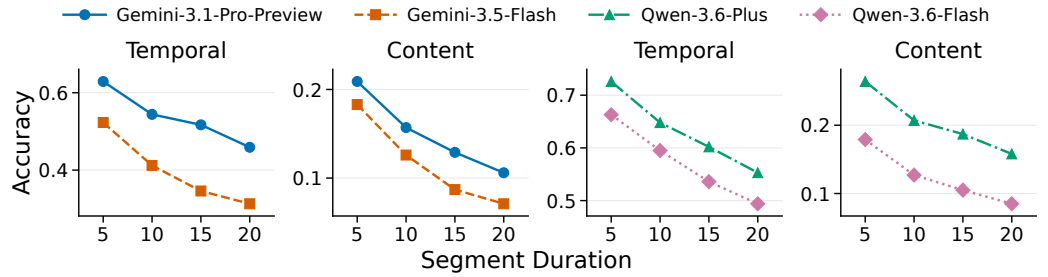

Figure 3: Temporal and content accuracy versus segment duration (5–20 minutes) for representative omni models (left two panels) and VL models (right two panels); both decline near-linearly as segments grow longer.

atively self-contained with limited cross-segment temporal dependency. In contrast, the Character category displays a clear recovery in the second (25%–50%) and fourth (75%–100%) quartiles, which we attribute to the narrative tendency of concentrating principal character appearances in the early-to-mid and late acts, where the rolling accumulation of history context provides increasingly rich character-level information. Most notably, the History category shows the most coherent pattern across models: an initial decline followed by a sharp rebound in the final quartile. We hypothesize that this recovery, rather than a continued degradation over time, is driven by the narrative conventions of film closing acts, where foreshadowing payoffs, character reunions, and plot convergence collectively activate and validate the accumulated historical context, enabling more effective exploitation of long-range dependencies.

4.4 Impact of Temporal Granularity

We further extend the evaluation by varying the duration of video segments from 5 to 20 minutes. As shown in Figure 3, both temporal and content accuracy decline in a near-linear fashion as the segment duration increases, i.e., as the temporal granularity becomes coarser. Qualitative cases in Appendix A.3 reveal the underlying cause: with a fixed context window budget, longer segments introduce substantially more video tokens, forcing increasingly sparse temporal sampling. This

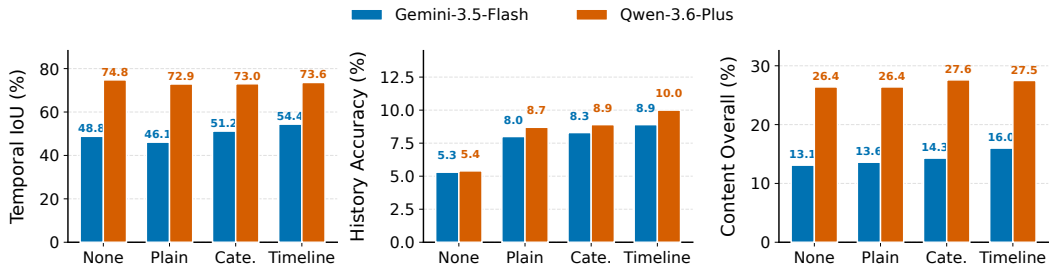

Figure 4: Ablation of rolling-history formats (plain / categorized / timeline) on temporal IoU, history accuracy, and overall content accuracy for two representative models; structured histories consistently outperform the plain baseline.

sparsity in turn leads to progressively fewer fine-grained details in the generated script, accounting for the observed performance degradation.

4.5 Impact of Structured Output

To investigate whether output formatting affects fine-grained comprehension, we compare standard captioning with a *structured output* setting, which explicitly directs the model to organize its predictions into distinct semantic fields (i.e., character, event, and scene). As illustrated in Figure 5, imposing this structural constraint yields substantial accuracy gains in the Character, Scene, and History dimensions, whereas the improvement in Event accuracy remains comparatively marginal. We attribute this disparity to an inherent omission bias in free-form generation. Narrative paragraphs naturally prioritize dynamic actions (events) at the expense of static attributes and cross-segment context. Structured prompting effectively mitigates this bias by forcing explicit allocation of attention.

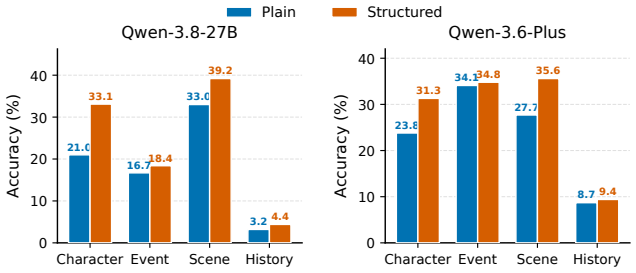

Figure 5: Per-dimension content accuracy of plain free-form captions versus structured output (separate character, event, and scene fields) on two representative models. Structured output yields consistent gains on Character, Scene, and History, while Event improves only marginally.

4.6 Analysis of History Patterns

To probe how the organization of accumulated context affects movie-level script generation, we compare four history patterns: *none*, which processes segments independently without history; *plain*, which stores history as flat prose; *categorized*, which structures it along the four content dimensions; and *timeline*, which appends segment histories in chronological order. As shown in Figure 4, removing history (*none*) leads to a substantial drop in history accuracy, and the performance gap between the two models largely vanishes. This indicates that without accumulated context, models are confined to intra-segment cues and cannot capture inter-segment dependencies. Among the history-enabled variants, both structured patterns consistently outperform *plain*, confirming that explicit organization matters. Intuitively, categorization reduces interference by separating facts into semantic slots, while chronological appending supplies temporal cues for aligning past events; accordingly, *timeline* achieves the best performance, particularly in temporal IoU and history accuracy. The gains are more pronounced for the weaker model, suggesting structured history acts as an external scaffold. Nevertheless, absolute improvements remain modest, indicating that better history representations alleviate but do not resolve the bottleneck in long-form narrative understanding.

4.7 Disentangling Temporal Localization from Narrative Comprehension

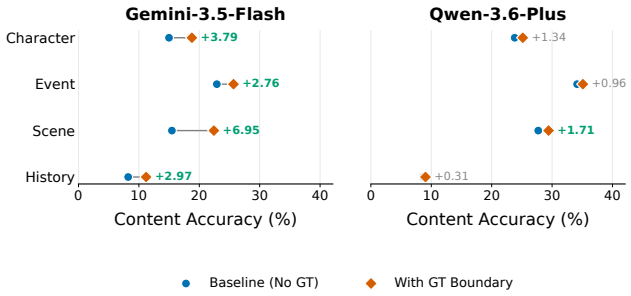

Figure 6: Per-dimension content accuracy with and without ground-truth (GT) scene boundaries for two representative models; supplying GT boundaries yields only marginal gains across all dimensions.

To decouple scene segmentation from content comprehension, we provide the models with ground-truth (GT) scene boundaries while keeping all other inputs and the generation budget identical. As shown in Figure 6, supplying GT boundaries yields only limited gains across all content dimensions: even with perfect temporal localization, content accuracy remains far from satisfactory. The improvements are comparatively larger on Scene and for the weaker model, suggesting that accurate segmentation mainly facilitates the description of local visual content and partly compensates for poorer boundary predictions. In contrast, the History dimension benefits the least, indicating that failures in inter-segment consistency stem not from mis-segmentation but from the inability to accumulate and exploit long-range narrative context. Taken together, these results confirm that the primary bottleneck of movie-level script generation lies in deep narrative comprehension rather than temporal boundary localization, and that correcting segmentation alone cannot close the performance gap.

Table 2: Meta-evaluation of judge models against human annotations. We randomly sample 100 scenes (totaling 1,115 rubrics) from the predictions and compare the automated judgments with human labels.

Judge Model	Content Acc. (%)	Agreement (%)	Precision (%)	Recall (%)	Macro F1 (%)
Qwen-3.8-27B (xhigh)	28.3	90.4	91.5	88.2	89.8
Qwen-3.8-27B (medium)	31.4	**92.5**	**94.2**	**91.5**	**92.8**
Qwen-3.8-27B (low)	31.2	91.9	92.8	90.4	91.6
DeepSeek-V4.1-Flash (max)	30.0	91.7	93.1	89.8	91.4
DeepSeek-V4.1-Flash (high)	30.2	92.2	93.8	90.5	92.1
GPT-OSS-120B (high)	31.4	91.0	90.5	87.2	88.8
GPT-OSS-120B (medium)	30.2	89.9	89.8	86.5	88.1
GPT-OSS-120B (low)	29.2	88.9	88.5	85.1	86.8

4.8 Meta-Evaluation of Judge Models

To ensure the reliability of our automated evaluation pipeline, we conducted a meta-evaluation by randomly sampling 100 scenes (totaling 1,115 rubrics) and comparing the automated judgments against human annotations. As shown in Table 2, the Qwen-3.8-27B (medium) model achieves the highest agreement rate (92.5%) with human labels, alongside superior precision and macro F1 scores. Furthermore, the agreement rates across different models exhibit a reasonable positive correlation with their content accuracy. Based on these results, we select Qwen-3.8-27B (medium) as the default judge model for all experiments.

5 Conclusion

In this work, we introduced **CineScriptBench**, the first movie-level benchmark explicitly designed to evaluate inter-segment narrative continuity and historical consistency in cinematic understanding. We systematically assessed the long-form comprehension capabilities of frontier MLLMs. Our extensive experiments reveal a critical bottleneck in current architectures: while they perform reasonably well in localized scene description and temporal boundary detection, they exhibit severe deficiencies in maintaining long-range narrative dependencies and cross-scene historical consistency. We hope CineScriptBench and our empirical findings serve as a rigorous baseline and a strong catalyst for future research, inspiring the development of next-generation models capable of profound, holistic cinematic understanding.

AI USE STATEMENT

In the preparation of this manuscript, we utilized artificial intelligence (AI) tools exclusively for language refinement, including grammar correction, proofreading, and stylistic polishing to enhance the overall readability of the text. These tools were strictly limited to linguistic assistance and were not employed for autonomous research, data analysis, literature generation, or the formulation of core scientific concepts and conclusions. We confirm that there was no over-reliance on AI throughout the research and writing process, and we take full responsibility for the originality, accuracy, and integrity of the final content.

REPRODUCIBILITY STATEMENT

We are committed to ensuring the full reproducibility of our benchmark. For the benchmark construction, we provide a comprehensive description of the data collection pipeline, filtering criteria, annotation guidelines, and quality control procedures in Section 3 and the supplementary materials (mainly for the codebase), enabling readers to reconstruct the dataset from scratch. For the model evaluation, we detail the evaluation protocol, including prompt templates, decoding hyperparameters, inference configurations, and scoring metrics for all compared models; these details ensure that the reported results can be faithfully reproduced. We believe that the combination of the detailed descriptions in the main text, appendices, and the supplementary materials provides sufficient information to independently reproduce all results presented in this paper.

REFERENCES

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report, 2025. URL https://arxiv.org/abs/2511.21631.

Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

Xinlong Chen, Yue Ding, Weihong Lin, Jingyun Hua, Linli Yao, Yang Shi, Bozhou Li, Yuanxing Zhang, Qiang Liu, Pengfei Wan, Liang Wang, and Tieniu Tan. Avocado: An audiovisual video captioner driven by temporal orchestration, 2025. URL https://arxiv.org/abs/2510.10395.

Xinlong Chen, Weihong Lin, Jingyun Hua, Linli Yao, Yue Ding, Bozhou Li, Bohan Zeng, Yang Shi, Qiang Liu, Yuanxing Zhang, Pengfei Wan, Liang Wang, and Tieniu Tan. Diadem: Advancing dialogue descriptions in audiovisual video captioning for multimodal large language models, 2026. URL https://arxiv.org/abs/2601.19267.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, Peixian Chen, Yanwei Li, Shaohui Lin, Sirui Zhao, Ke Li, Tong Xu, Xiawu Zheng, Enhong Chen, Caifeng Shan, Ran He, and Xing Sun. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis, 2025. URL https://arxiv.org/abs/2405.21075.

Qingqiu Huang, Yu Xiong, Anyi Rao, Jiaze Wang, and Dahua Lin. Movienet: A holistic dataset for movie understanding, 2020. URL https://arxiv.org/abs/2007.10937.

Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, Limin Wang, and Yu Qiao. Mvbench: A comprehensive multi-modal video understanding benchmark, 2024. URL https://arxiv.org/abs/2311.17005.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023. URL https://arxiv.org/abs/2304.08485.

Hongbo Liu, Jingwen He, Yi Jin, Dian Zheng, Yuhao Dong, Fan Zhang, Ziqi Huang, Yinan He, Yangguang Li, Weichao Chen, Yu Qiao, Wanli Ouyang, Shengjie Zhao, and Ziwei Liu. Shotbench: Expert-level cinematic understanding in vision-language models, 2025. URL https://arxiv.org/abs/2506.21356.

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

Junfu Pu, Yuxin Chen, Teng Wang, and Ying Shan. Omniscript: Towards audio-visual script generation for long-form cinematic video, 2026. URL https://arxiv.org/abs/2604.11102.

Shaden Shaar, Bradon Thymes, Sirawut Chaixanien, Claire Cardie, and Bharath Hariharan. Movierecapsqa: A multimodal open-ended video question-answering benchmark, 2026. URL https://arxiv.org/abs/2601.02536.

Qwen Team. Qwen3.5-omni technical report, 2026. URL https://arxiv.org/abs/2604.15804.

Weihan Wang, Zehai He, Wenyi Hong, Yean Cheng, Xiaohan Zhang, Ji Qi, Xiaotao Gu, Shiyu Huang, Bin Xu, Yuxiao Dong, Ming Ding, and Jie Tang. Lvbench: An extreme long video understanding benchmark, 2025a. URL https://arxiv.org/abs/2406.08035.

Xinran Wang, Songyu Xu, Xiangxuan Shan, Yuxuan Zhang, Muxi Diao, Xueyan Duan, Yanhua Huang, Kongming Liang, and Zhanyu Ma. Cinetechbench: A benchmark for cinematographic technique understanding and generation, 2025b. URL https://arxiv.org/abs/2505.15145.

Di Wu. Multimodal large language model-based automatic shot decomposition and narrative structure analysis in films. In *Proceedings of the 5th International Conference on Computer, Artificial Intelligence and Control Engineering*, pp. 566–572, 2026.

Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. Longvideobench: A benchmark for long-context interleaved video-language understanding, 2024. URL https://arxiv.org/abs/2407.15754.

Kangyi Wu, Pengna Li, Jingwen Fu, Yizhe Li, Yang Wu, Yuhan Liu, Jinjun Wang, and Sanping Zhou. Event-equalized dense video captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8417–8427, June 2025.

Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. Next-qa:next phase of question-answering to explaining temporal actions, 2021. URL https://arxiv.org/abs/2105.08276.

Jin Xu, Zhifang Guo, Hangrui Hu, Yunfei Chu, Xiong Wang, Jinzheng He, Yuxuan Wang, Xian Shi, Ting He, Xinfa Zhu, Yuanjun Lv, Yongqi Wang, Dake Guo, He Wang, Linhan Ma, Pei Zhang, Xinyu Zhang, Hongkun Hao, Zishan Guo, Baosong Yang, Bin Zhang, Ziyang Ma, Xipin Wei, Shuai Bai, Keqin Chen, Xuejing Liu, Peng Wang, Mingkun Yang, Dayiheng Liu, Xingzhang Ren, Bo Zheng, Rui Men, Fan Zhou, Bowen Yu, Jianxin Yang, Le Yu, Jingren Zhou, and Junyang Lin. Qwen3-omni technical report, 2025. URL https://arxiv.org/abs/2509.17765.

Linli Yao, Yuancheng Wei, Yaojie Zhang, Lei Li, Xinlong Chen, Feifan Song, Ziyue Wang, Kun Ouyang, Yuanxin Liu, Lingpeng Kong, Qi Liu, Pengfei Wan, Kun Gai, Yuanxing Zhang, and Xu Sun. Timechat-captioner: Scripting multi-scene videos with time-aware and structural audio-visual captions, 2026. URL https://arxiv.org/abs/2602.08711.

Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Zhengyang Liang, Shitao Xiao, Minghao Qin, Xi Yang, Yongping Xiong, Bo Zhang, Tiejun Huang, and Zheng Liu. Mlvu: Benchmarking multi-task long video understanding, 2025. URL https://arxiv.org/abs/2406.04264.

A APPENDIX

Algorithm 1 Progressive Movie-Level Evaluation Protocol

Require: Full-length movie video V, shot boundaries $\mathcal{L}$, target segment duration Δ, evaluator f_θ
Ensure: Movie-level generated script $\hat{\mathcal{Y}}$

Phase 1: Shot-Snapped Partitioning
Initialize current time $t \leftarrow 0$, segment index $i \leftarrow 1$
while $t <$ Duration(V) **do**
  $t_{target} \leftarrow t + \Delta$
  $t_{snap} \leftarrow \arg\min_{t_b \in \mathcal{L}} |t_b - t_{target}|$ ▷ Snap to nearest shot boundary
  Define video segment $V_i \leftarrow V[t : t_{snap}]$
  $t \leftarrow t_{snap}, \quad i \leftarrow i + 1$
end while
Let total segments $N \leftarrow i - 1$
Phase 2: Progressive Generation with Overlap Handling
Initialize rolling history $H_0 \leftarrow \emptyset$, overlap context $S_{overlap} \leftarrow \emptyset$
for $i = 1$ to N **do**
  Extract video segment V_i
  $(\hat{Y}_i, H_i) \leftarrow f_\theta(V_i, H_{i-1}, S_{overlap})$ ▷ Generate script and update history
  $S_{overlap} \leftarrow \emptyset$ ▷ Reset overlap for current segment
  for each predicted scene $\hat{y} \in \hat{Y}_i$ **do**
    if end time of $\hat{y}$ exceeds the boundary of V_i **then**
      $S_{overlap} \leftarrow \hat{y}$ ▷ Retain for the next window
      Truncate $\hat{y}$ to the end boundary of V_i for current evaluation
    end if
  end for
  Append $\hat{Y}_i$ to $\hat{\mathcal{Y}}$
end for
return $\hat{\mathcal{Y}}$

A.1 DETAILED EVALUATION PROTOCOL

To provide a rigorous and reproducible blueprint of our evaluation methodology, we formalize the complete progressive evaluation protocol in Algorithm 1. The protocol consists of two core phases designed to balance temporal granularity with cinematic integrity.

Phase 1: Shot-Snapped Partitioning. Instead of naively slicing the movie into rigid n-minute (e.g., $\Delta = 5$ min) chunks, which often cuts through continuous shots and destroys semantic coherence, we employ a shot-snapping strategy. For each target boundary t_{target}, the algorithm searches the pre-computed shot boundary sequence $\mathcal{L}$ and snaps the cut to the nearest valid shot transition t_{snap}. This ensures that each evaluated segment V_i comprises complete cinematic units, preventing the model from being forced to interpret fragmented visual actions.

Phase 2: Progressive Generation and Overlap Handling. The evaluator f_θ processes the partitioned segments chronologically, conditioned on the rolling history H_{i-1}. Crucially, cinematic scenes frequently transcend arbitrary time windows. To handle this, we introduce a cross-boundary overlap mechanism ($S_{overlap}$). If the model predicts a scene $\hat{y}$ that crosses the boundary of the current segment V_i (i.e., its predicted end time exceeds the snapped boundary t_{snap}), this incomplete scene is recorded as contextual overlap. It is then passed as a conditioning input to the subsequent segment V_{i+1}. This overlap handling serves two critical purposes: (1) it prevents the model from redundantly re-annotating the same continuous event in the next window, and (2) it enforces temporal continuity, ensuring that the progressive script generation accurately reflects the uninterrupted narrative flow across segment boundaries.

A.2 STATISTICS OF CINESCRIPTBENCH

Table 3 and Figure 7 summarize the overall scale and the distributions of CineScriptBench. The benchmark comprises 73 full-length movies totaling 153 hours, with an average duration of ap-

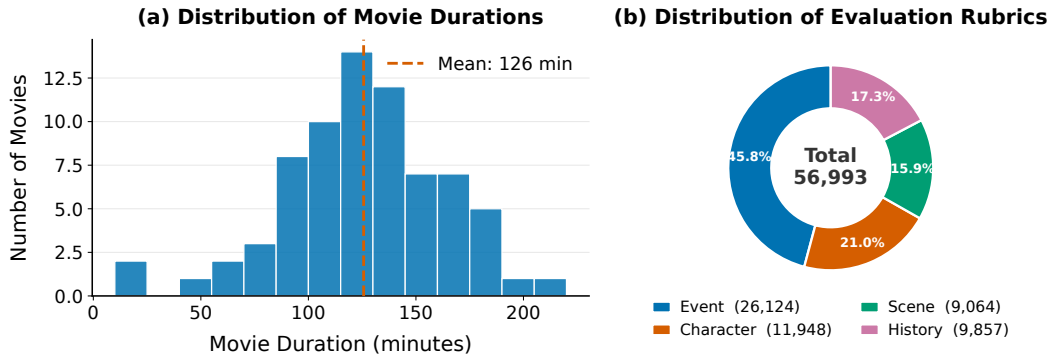

Figure 7: Overall statistics of CineScriptBench. **(a)** Movie durations concentrate in the standard 90–180 minute range, with a mean of 126 minutes. **(b)** The 56,993 rubrics are distributed across the four content dimensions, with History rubrics (17.3%) explicitly targeting inter-segment narrative continuity.

Table 3: Overall Statistics of OMNIMOVIEBENCH.

Statistic	Value	Description
# Full-length Movies	73	Curated high-quality cinematic sources
Total Duration	153 Hours	Approx. 126 mins per movie on average
# Scene Segments	5,524	Snapped to natural shot boundaries
# Evaluation Rubrics	56,993	Fine-grained LLM-as-a-Judge criteria
Avg. Rubrics per Scene	∼10.3	Covering 4 content dimensions
Avg. Scenes per Movie	∼75.7	Capturing narrative and spatial shifts

proximately 126 minutes per movie. As shown in Figure 7 (a), movie durations concentrate in the 90–180 minute range, matching the standard length of feature films, while a small number of shorter and longer titles preserve source diversity. These movies are partitioned into 5,524 scene segments (about 75.7 per movie), each snapped to natural shot boundaries, and are annotated with a total of 56,993 fine-grained evaluation rubrics, i.e., roughly 10.3 rubrics per scene spread over the four content dimensions, ensuring that every scene is examined from multiple fine-grained perspectives under the LLM-as-a-Judge protocol.

Figure 7 (b) further breaks down the rubrics by dimension. Event rubrics form the largest share (26,124, 45.8%), reflecting that character actions and plot developments constitute the core of cinematic narratives; Character (11,948, 21.0%), History (9,857, 17.3%), and Scene (9,064, 15.9%) together account for the remaining 54.2%, providing balanced coverage of character identities, spatial contexts, and inter-segment historical dependencies. In particular, History rubrics exist only for scenes whose final annotations absorb historical information from earlier scenes; their substantial presence confirms that CineScriptBench evaluates inter-segment narrative continuity explicitly and densely, rather than treating it as a by-product of the other dimensions.

A.3 CASE STUDY

To dissect where progressive script generation breaks down, we qualitatively inspect *Interstellar* under all four segment durations and summarize three representative cases in Table 4. All excerpts are English translations of the model's raw outputs, truncated for space; Evt/Hist report the passed event/history rubric items of the corresponding ground-truth scene.

Case I: intact local comprehension, absent historical absorption. In the hospital-reunion scene (02:39:45–02:42:06), where Cooper meets his dying daughter Murph, the model reconstructs the local narrative almost perfectly at *every* duration (event rubrics 4/5 at 5-, 10-, 15- and 20-min): it names both characters, the oxygen tube, Cooper's admission that he was the childhood "ghost," and Murph's urging to go find Brand. Yet the three history rubrics of this scene—Murph's refusal to bid farewell and to watch his transmitted videos when Cooper left home, the bookshelf/Morse-

Table 4: Qualitative cases on *Interstellar* across segment durations. Excerpts are translated and truncated model outputs; Evt/Hist denote passed event/history rubric items of the matched ground-truth scene ("–" means the scene carries no history rubrics).

Setting	Model output excerpt (translated)	Evt / Hist
Case I: Hospital reunion (02:39:45–02:42:06)		
5-min	"Cooper comes to a reception marked with the NASA logo, where a red-haired clerk tells him his family are waiting ... behind the blue door he finds his daughter Murph, now a white-haired old woman on a nasal oxygen tube ... holding her hand, Cooper tearfully admits that he was the 'ghost' of her childhood; Murph says she always knew, because he had promised to return; she urges him to leave and go find Brand."	4/5, 0/3
20-min	"Cooper comes to the NASA-marked reception ... pushes open the door and sees many relatives gathered around a bedside ... his daughter Murph, now a dying white-haired old woman ... Cooper admits that he was the 'ghost' in her study back then; Murph replies that she always knew ... she urges Cooper to leave and find Brand."	4/5, 0/3
Case II: Cruise video messages (00:55:24–00:57:01)		
5-min	"Cooper ... video-calls his son Tom on Earth. Tom tells his father that the school requires him to retake plant pathology but allows him to take advanced agriculture courses early ... the screen then switches to grandfather Donald, who apologizes that Murph stubbornly refuses to greet."	3/6, 2/3
15-min	"After the time jump, Cooper ... watches the latest video messages from his children. Daughter Murph complains about school courses, son Tom reports the family situation ... Cooper's eyes moisten as he realizes he has missed their growing-up."	2/6, 0/3
Case III: Launch to departure (00:42:26–00:50:20)		
5-min	"... robot TARS emits humorous banter; Cooper interacts with it and lowers its humor parameter from 100% to 75% ... Cooper directs Doyle for fine operation, reminding him to 'take it slowly'; as the docking ring closes and locks, Doyle reports 'target locked' ... Professor Brand ... informs them that the journey to Saturn is expected to take two years."	4/7, –
20-min	"... TARS interacts with its characteristic humor setting, and Cooper requests lowering its humor level ... Cooper manually operates the spacecraft to approach the rotating 'Endurance' ... at this point they receive a video message from Earth showing the aged Donald and the grown Tom and Murph, a recording transmitted years later ..."	1/7, –

code origin of the "ghost" contact, and Brand's identity as Cooper's former crewmate—score 0/3 at *all* four durations. Ample context therefore does not rescue inter-segment consistency: historical absorption is a capability gap of current MLLMs rather than an artifact of window size.

Case II: coarse windows corrupt accumulated identity bindings. In the cruise video-message scene (00:55:24–00:57:01), the 5-min run correctly binds each message to its sender through the rolling history: "Tom tells his father that the school requires him to retake plant pathology but allows him to take advanced agriculture courses early ... the screen then switches to grandfather Donald, who apologizes that Murph stubbornly refuses to greet" (history 2/3). At 15-min the same span is absorbed into a coarser window and the bindings collapse: "Daughter Murph complains about school courses, son Tom reports the family situation"—the academic update is re-attributed to Murph and Donald disappears entirely (history 0/3). History rubrics across durations degrade as $2/3 \rightarrow 1/3 \rightarrow 0/3 \rightarrow 1/3$: once the rolling history loses an anchor fact, later segments inherit the corrupted state.

Case III: granularity-driven detail erosion. For the launch-to-departure span (00:42:26–00:50:20), the 5-min run emits eight fine-grained segments that preserve quantitative and causal details—TARS's humor setting lowered from 100% to 75%, its honesty at 90%, the "twelve minutes" docking countdown, Doyle executing the manual docking ("target locked"), and Professor Brand's two-year estimate for Saturn—scoring 4/7 event rubrics. The 20-min run compresses the same span into four segments: the numeric settings vanish, the docking is re-attributed to Cooper,

and the later Earth-message montage (grown Tom and Murph) is conflated into the spin-up video call, leaving only 1/7 event rubrics. Movie-wide, the number of predicted scenes shrinks from 265 (5-min) to 130 (20-min), confirming that a fixed token budget forces sparser temporal sampling as windows grow, which directly produces the near-linear decline in Figure 3.

Taken together, Cases I–III separate two failure regimes of movie-level script generation: a duration-independent *historical-absorption* deficit (Case I), and a duration-dependent *context-erosion* deficit that corrupts or drops accumulated facts (Cases II–III).

Table 5: Hyperparameters of the benchmark construction pipeline (default values used to produce the released benchmark).

Stage	Parameter	Value	Description
Preprocessing	Video resolution	480p	Uniform compression of all sources
	Shot detector	PySceneDetect	Produces the shot sequence $\mathcal{L}$
Segmentation (C1)	Window presets (base / extend)	120 s / 60 s	Initial window and extension step
	Max search window	300 s	Upper bound of window extension
	Min segment length	2 s	Rejects too-short proposals
	Judge context extension	60 s	Extra context beyond proposal end for arbitration
Annotation (C2)	Min caption length	40 chars	Validation floor for draft captions
	Caption length	6–10 sentences	Coherent single-paragraph narrative
	Min verify questions	4	Per-scene verification QA
	Max uncertain ratio	0.3	Verification invalid above this ratio
	Consistency target / max repair loops	1.0 / 2	Trigger and cap of targeted repair
Merging (C3)	Merge policy	Conservative	Keep separate unless same-scene is certain
	Checklist cap after merge	12 items	Four categories combined
History memory (C4)	Rolling summary cap	800 chars	Fixed-length narrative R_{i-1}
	Memory items per segment	4–10	Keyword items $\mathcal{K}_{i-1}$
	Query items per segment	cap 2	Precision-over-recall queries $\mathcal{Q}_i$
	Embedding model	BGE-M3	Embeds memory and query items for retrieval
Refinement (C5)	Top-k reference segments	2	Retrieved per query item
	Length-ratio guard	[0.5, 2.0]	Rejects over- or under-long refined captions
Rubrics (C7)	Rubric cap per segment	12 items	Four categories combined
	History rubrics	Contrastive	Derived by comparing the two caption versions

A.4 Hyperparameters and Configurations

Table 5 lists the construction-pipeline hyperparameters, i.e., the default values used to produce the released benchmark. Across all pipeline stages, failed or schema-invalid LLM responses are re-queried with a retry budget of 2–3 attempts, and all concurrency parameters are infrastructure-level and do not affect results. For evaluation, all models are inferred under a single unified configuration to ensure a fair comparison: the visual sampling rate is fixed at 2 FPS, and the decoding temperature is set to 0.0 for deterministic generation; for models that do not support explicit temperature configuration, we fall back to their default decoding values. All remaining decoding hyperparameters follow each model's official default settings. Especially, for open-source omni models, we set the visual sampling rate to 0.4 FPS due to the limited context windows.

A.5 Prompt Templates

This section provides the complete prompt templates and configuration details required to reproduce CineScriptBench and our evaluations. All prompts follow a unified card format: each card carries an ID (`C*` for construction, `E*` for evaluation), is preceded by a one-sentence purpose description, and marks dynamic inputs with placeholders `{...}`, whose conventions are summarized in Table 6.

A.5.1 Placeholder Conventions

All dynamic inputs in the prompt cards of Appendix A.5.2–A.5.3 are marked with brace-wrapped placeholders, typeset in typewriter face via the `\ph` command (e.g., `{CAPTION}`). At runtime, each placeholder is substituted with the corresponding content: media placeholders denote video

Table 6: Placeholder conventions used across all prompt templates.

Placeholder	Meaning
{VIDEO_SEGMENT}	Current video segment (visual frames + audio)
{SHOT_BOUNDARIES}	Candidate shot boundaries from PySceneDetect
{HISTORY}	Rolling historical context H_{i-1}
{SCRIPT}	Independent or refined scene-level script
{MEMORY_ITEMS}	Retrieved top-k memory items $\mathcal{C}_i$
{PRED_SCRIPT}	Model prediction to be judged
{GT_SCRIPT}	Ground-truth final script
{RUBRICS}	Fine-grained rubric items for judging
{PREV_ANALYSIS}	Judgment record (analysis) from the previous window-extension round
{SCENE_INDEX}	Index of the current scene segment
{TOTAL_SCENES}	Total number of scene segments in the movie
{SHOT_BOUNDARIES_REL}	Shot boundaries relative to the segment start time
{CAPTION}	Generated scene caption to be verified
{QUESTION_LIST}	List of verification questions in JSON format
{PREV_SUMMARY}	Rolling plot summary accumulated up to the previous scene segment
{MAX_SUMMARY_CHARS}	Hard character limit of the fixed-length rolling summary
{MAX_QUERY_ITEMS}	Upper bound of query items per segment (precision-over-recall)
{REF_VIDEO}	Video clip of the retrieved reference (earlier) scene segment
{CUR_VIDEO}	Video clip of the current scene segment
{REF_SCENE_META}	Scene index and absolute time range of the reference segment
{CUR_SCENE_META}	Scene index and absolute time range of the current segment
{REF_CAPTION}	Caption of the reference segment (refined version when available)
{CUR_CAPTION}	Current-segment caption (independent or intermediately refined)
{QUERY_LIST}	Numbered keyword-form queries to be verified in the current round
{CLIP_RANGE}	Absolute time range of the continuous clip covering segments A and B
{A_TIME_RANGE}	Absolute time range of segment A
{B_TIME_RANGE}	Absolute time range of segment B
{A_CAPTION}	Final caption of segment A (reference only)
{B_CAPTION}	Final caption of segment B (reference only)
{A_CHECKLIST}	Four-dimension checklist of segment A (reference only)
{B_CHECKLIST}	Four-dimension checklist of segment B (reference only)
{SCENE_TIME_RANGE}	Absolute time range of the scene segment under review
{INDEPENDENT_CAPTION}	Independent caption describing only current-segment content
{FINAL_CAPTION}	Final caption with historical information injected (object under review / correction / unification)
{DETECT_CONCLUSION}	One-sentence conclusion of the hallucination detection
{DETECT_ANALYSIS}	Point-by-point verification analysis of the hallucination detection
{SCENE_ID}	Identifier of the scene segment
{WINDOW_DURATION}	Duration of the current video window in seconds
{ROLLING_HISTORY}	Rolling plot history accumulated by the model up to the previous window
{GOLDEN_BOUNDARY_LIST}	Given ground-truth scene boundaries

clips attached to the message, textual placeholders denote strings or JSON objects produced by upstream pipeline stages, and scalar placeholders denote pipeline hyperparameters taken from Table 5. Placeholders shared by multiple cards carry identical semantics throughout (e.g., {CAPTION} and {PREV_SUMMARY}), whereas card-specific placeholders are defined locally; some placeholders appear only conditionally (e.g., {PREV_ANALYSIS} is included only when the current window is an extension of a previous one). Media placeholders are either listed explicitly in the user block (e.g., Cards C1, C2.1, C5) or implicitly attached together with a textual description of the covered time range (e.g., Cards C2.3, C3). Table 6 summarizes every placeholder used in this appendix together with the cards in which it appears.

A.5.2 Benchmark Construction Prompts

C1: Scene transition detection. Following the window-based detection described in Section 3.2, the MLLM inspects each candidate window and determines whether it contains a complete and meaningful scene; if no scene is identified, the window is extended by a fixed step and the query is repeated, with the previous-round analysis carried over to stabilize consecutive judgments:

C1: Scene Transition Detection

[System] You are an expert film editor and video-understanding assistant. Given a video segment and a list of candidate shot-boundary timestamps produced by an automatic shot detector, determine whether the segment contains a complete and meaningful scene, and localize the first such scene by selecting its start and end from the candidate timestamps.
[User]
Video segment: {VIDEO_SEGMENT}
Candidate shot boundaries (seconds, relative to segment start): {SHOT_BOUNDARIES}
Task: identify the first complete and meaningful scene in the segment and report its range (start, end).
Notes:
– 0 denotes the starting boundary of the segment; start may be 0.
– The last candidate denotes the ending boundary of the segment; end must not be the segment ending boundary, except when the segment reaches the end of the movie and the scene ends at the video ending.
– Both start and end must be selected from the candidate boundaries.
[Optional block — included only when the current window is an extension of a previous window]
Previous-round judgment record
The current window is obtained by extending the previous window forward and contains all content of the previous window. The previous judgment and analysis are:
{PREV_ANALYSIS}
Refer to the previous analysis and re-judge based on the complete current window; avoid swinging back and forth between consecutive rounds.
[Output format] Strict JSON: `{"found": bool, "start": float, "end": float, "confidence": "high|medium|low", "analysis": string}`

C2: Independent scene annotation and verification. Following the independent annotation protocol in Section 3.2, the MLLM first generates a detailed scene caption, and then performs a self-verification cycle by generating and answering fine-grained checking questions. This process consists of three sub-prompts:

C2.1: Initial Scene Captioning

[System] You are an expert video understanding assistant. Your task is to generate a detailed, fine-grained script/caption for a given video scene segment. The caption should comprehensively describe the scene environment, characters, actions, dialogues, and causal events in a coherent narrative style.
[User]
Video segment: {VIDEO_SEGMENT}

Please write a plot caption for this scene segment (Scene {SCENE_INDEX} out of {TOTAL_SCENES} in the full movie; this position is provided only for grasping the narrative context).
Shot boundaries within the segment (relative time in seconds, for reference only): {SHOT_BOUNDARIES_REL}
Emphasis: This annotation is performed completely independently. You do not have any information about other parts of the movie.

C2.2: Verification Question Generation

[System] You are a quality inspector for annotated data. Given a plot caption for a scene segment, generate accurate and comprehensive verification questions to check whether the caption is consistent with the video content.
Requirements:
– Generate at least 4 questions covering different aspects of the caption (e.g., characters, events, environment, causality, object details), prioritizing the most important claims and those most prone to hallucination.
– Each question must be answerable solely by watching the video and must be a yes/no question.
– Attach the annotated claim being verified and its corresponding aspect to each question.
– When quoting dialogue or specific words in the question and claim, strictly use standard double quotes ("").
Output format (strictly followed): Do not output JSON. Output each question as a small section with 4 lines: the first line is the question number (Q1, Q2, ...), and the next 3 lines start with "aspect:", "question:", and "claim:", respectively. Do not output any other content. Example:
Q1
aspect: Character
question: Does the man in the scene wear a gray suit throughout?
claim: The man wears a gray suit.
[User]
Scene caption: {CAPTION}

C2.3: Verification Answering

[System] Please watch the video carefully and answer the following questions one by one based solely on the video content (yes / no / uncertain), providing brief evidence.
Output format (strictly followed): Output only a single JSON object, without any other text, explanations, or code block markers. The "evidence" must be written in a single line without line breaks. When quoting dialogue or words, strictly use standard double quotes ("").
Expected JSON structure: {"answers": [{"id": 1, "answer": "yes/no/uncertain", "evidence": "brief evidence"}]}
[User]
Question list: {QUESTION_LIST}

C3: Over-segmentation merging (boundary refinement). To prevent over-segmentation, a merging pass slides over consecutive scene segments: for each temporally adjacent pair (A earlier, B later), the MLLM watches the continuous clip spanning from A's start to B's end and judges whether the two segments actually belong to the same scene, i.e., whether the location/set remains unchanged with no scene switch in between (changes of camera position, angle, shot size, shot/reverse-shot patterns, and editing rhythm do not constitute a boundary). The captions and checklists of the two segments serve only as auxiliary references; when uncertain, the model conservatively keeps the pair separate. If a pair is judged mergeable, the model selectively regenerates a unified caption and checklist for the merged segment rather than mechanically concatenating them, and the slide continues over the merged sequence until no adjacent pair remains mergeable, yielding the final refined scene segmentation.

C3: Over-Segmentation Merging (Boundary Refinement)

[System] You are a professional structural audit expert for film scene segments. You are given the annotations of two **temporally adjacent** scene segments from the same movie (segment A earlier, segment B later). Determine whether they are actually the same scene segment and should be merged into one.
Definition of a scene segment
Scene segment = a continuous stretch of content with a unified spatiotemporal context: the location/set remains unchanged and stays within the same scene, from entering that scene until the scene ends (switches to another location/set).
Boundary = a scene change: the location/set switches to another scene (a time jump accompanying the scene change also counts). The following changes do NOT constitute a boundary: camera position, angle, or shot-size changes; shot/reverse-shot; editing rhythm changes; and minor entrances/exits or movements of characters within the same location.
Judgment criteria
– Base the judgment primarily on the continuous video clip covering [A start, B end] that you watch; the two captions and checklists are only references; do not introduce external knowledge about the film;
– If in the video A and B share the same location/set and no scene switch occurs between them (only camera/angle/shot-size changes, shot/reverse-shot, editing rhythm changes, or minor character entrances/exits and movements within the same location), judge mergeable;
– If in the video B switches to another location/set (even if the characters are the same, and even if accompanied by a time jump), judge not mergeable;
– When uncertain, be conservative: if you cannot clearly determine that they are the same scene, do not merge.
Regeneration requirements when mergeable
You are given the final captions and final checklists of the two segments; they are only references—defer to the video you watch, do not mechanically concatenate, and make selective choices:
– caption: write the plot of the merged segment as one coherent narrative paragraph (no bullet points, no titles). Retain content necessary for understanding the plot (scene environment, characters present, who did what, why, and with what result, key dialogue points, turning points, ending state), and remove trivial content unimportant to plot understanding; keep character appellations consistent across the two segments; do not assume attributes that cannot be confirmed from the video and the reference annotations; defer to the video and do not fabricate content absent from the video and the reference annotations; always use standard double quotes (”) for quoted dialogue;
– checklist: combining with the video, select and merge from the two checklists, retaining only items necessary for understanding the merged plot; deduplicate semantically redundant items to one; delete trivial items; each item remains a self-contained single-sentence check containing one atomic fact; character items describe only static character attributes (identity/appearance/clothing), with behaviors and actions assigned to event; the four categories together generally contain no more than 12 items.
Output format (strictly followed)
Output only a single JSON object with no other content; string values written in one line without line breaks.
Not mergeable:
{”merge”: ”no”}
Mergeable:
{”merge”: ”yes”, ”caption”: ”merged caption”, ”checklist”: {”character”: [...], ”event”: [...], ”scene”: [...], ”history”: [...]}}
[User]
The video clip covers {CLIP_RANGE}; within it, segment A spans {A_TIME_RANGE} and segment B spans {B_TIME_RANGE}.
Reference annotation of segment A
caption: {A_CAPTION}
checklist:
{A_CHECKLIST}

Reference annotation of segment B
caption: {B_CAPTION}
checklist:
{B_CHECKLIST}
Please watch the video, determine whether A and B should be merged, and output only the specified JSON.

C4: Hierarchical historical memory construction. To instantiate the hierarchical historical memory described in Section 3.2, three generators are invoked for each scene segment: a rolling-summary updater maintaining the fixed-length narrative R_{i-1}, a memory-item extractor producing retrievable keyword items $\mathcal{K}_{i-1}$, and a query-item generator identifying points in the current caption that require historical context $\mathcal{Q}_i$. The three generators run concurrently within a segment, while segments are processed strictly sequentially so that the rolling summary chains forward. The generated items are not annotation evidence themselves; they serve as retrieval clues consumed by the deterministic Top-k retrieval in Eq. (3).

C4.1: Rolling Summary Update

[System] You are a rolling-summary compressor for movie plots. Given the "rolling summary up to the previous scene segment" (empty for the first segment of the movie) and the "plot caption of the current scene segment," output the updated rolling summary.
The rolling summary is of fixed length: no matter how far the movie progresses, the summary retains only the most critical plot content; its total length remains constant and never grows with the number of scenes.
Requirements:
– Incorporate the new information from the current caption while compressing the old summary to make room for it: prioritize retaining main characters' names and identities, key events and causal relations, and unresolved foreshadowing; secondary details from earlier scenes gradually fade out or are deleted;
– Statements conflicting with the current content are overridden by the current content;
– Write as one coherent narrative paragraph without line breaks, bullet points, or titles, with the total length strictly within the required character limit;
– When quoting dialogue or words, always use standard double quotes ("");
– Synthesize only from the given summary and caption; speculation, invention, or external knowledge about the movie is strictly forbidden.
Output format (strictly followed): output only the updated complete rolling summary text itself; do not output JSON, do not wrap it in code blocks, and do not output any explanation, title, prefix, or suffix.
[User]
Rolling summary up to the previous scene segment:
{PREV_SUMMARY} (empty when the current segment is the first scene of the movie)
Plot caption of the current scene segment:
{CAPTION}
Please output the updated rolling summary. Hard constraint: the total length must not exceed {MAX_SUMMARY_CHARS} characters; since the rolling summary is fixed-length, if the merged content exceeds the limit, preferentially compress older content and retain only the most critical plot.

C4.2: Memory Item Extraction

[System] You are a plot-memory extractor. Given the plot caption of a scene segment, extract memory items for that scene. Each item is a set of keywords (not sentences), which will be embedded and stored in a vector database for later semantic retrieval of "which historical information in earlier scenes is related to a given plot point." The items themselves are not used as evidence for later annotation (the evidence is the video and the full captions); they serve only as retrieval clues, so the keywords should point as precisely as possible to the corresponding plot content.

Item types (choose from the following four):
– character: characters appearing in this segment (name/appellation, appearance features, identity);
– event: key plot events in this segment (agent + action + key object/outcome);
– object: objects or locations with special narrative function (name, features, state);
– causal: motives, preconditions, promises, foreshadowing, and other information that may produce cross-scene associations.
Writing requirements:
– Each item consists of 2–6 keywords or phrases separated by commas, centered on a single memory point (one character, one event, one object, or one causal link);
– Keywords must carry the most discriminative identifying information (character names or appellations, appearance features such as clothing or hairstyle, location names, object names, concrete actions, etc.) to facilitate retrieval hits;
– Do not write complete sentences; do not use connectives or decorative descriptions; verbatim dialogue may be quoted as keywords;
– When quoting dialogue or words, always use standard double quotes (””);
– Extract 4–10 items in total, covering the plot points in this segment that truly have retrieval value;
– Extract only from the given caption; speculation or invention is strictly forbidden.
Output format (strictly followed): output only a single JSON object, without any other text, explanation, or code-block markers:
{”items”: [{”type”: ”character — event — object — causal”, ”keywords”: ”keyword1, keyword2, keyword3”}]}
[User]
The caption of the scene segment is as follows:
{CAPTION}

C4.3: Query Item Generation

[System] You are a retrieval-need identifier for plot annotation. Given the independent caption of a scene segment and the rolling plot summary up to but excluding that segment (the summary is explicit historical information; the caption is the independent annotation of the current segment), identify, based on the relationship between the two, the points in the caption that require retrieving historical information from earlier scenes, and output retrieval items.
Two situations warrant a retrieval item:
– Historical information can further enrich the caption: a character, event, or object in the caption, read together with the rolling summary, turns out to have corresponding historical information in earlier plot (e.g., the summary shows that the character was previously named or introduced, that the event has a precondition or foreshadowing, or that the object appeared before); the rolling summary here serves as a clue for discovering retrieval points;
– Historical information is missing: the caption itself contains content that can only be understood with earlier plot, e.g., a character known only by appearance or pronoun with unknown identity or name, an event whose motive or cause is unstated, or an object whose origin or special meaning is unclear.
Item types (same taxonomy as memory items):
– character: a character in the caption for whom the summary indicates that early identity, name, or background information is available, or whose identity or origin the caption itself leaves unclear;
– event: an event in the caption for which the summary indicates a corresponding precondition, motive, or foreshadowing, or whose motive and cause the caption itself does not state;
– object: an object or location in the caption whose origin or special meaning derives from earlier plot;
– causal: caption content that echoes historical plot shown in the summary via causality or foreshadowing, worth fetching earlier-scene details to confirm.
Strictly follow the precision-over-recall principle:

– The vast majority of captions need no retrieval items: content unrelated to historical plot, content fully understandable from the caption itself, generic common-sense references (passersby, clerks, etc.), and content naturally introduced at first appearance must never be output;
– Output an item only when you are certain that this point in the caption is truly related to earlier plot and that fetching historical information would substantially improve the annotation; at most {MAX_QUERY_ITEMS} items.
Writing requirements (isomorphic to memory items, i.e., a set of keywords rather than sentences):
– Each item consists of 2–6 keywords or phrases separated by commas: the front part gives the identifying features of the point (character appearance/appellation, event content, object name), and the back part gives the type of historical information to fetch (identity, name, motive, precondition, origin, meaning, etc.);
– Example: {"type": "character", "keywords": "man in black coat, scar on left cheek, identity, name"}.
Output format (strictly followed): output only a single JSON object, without any other text, explanation, or code-block markers. When no point requires retrieval: {"items": []}; otherwise: {"items": [{"type": "character — event — object — causal", "keywords": "keyword1, keyword2"}]}
[User]
Rolling plot summary up to but excluding this scene segment:
{PREV_SUMMARY} (empty when the current segment is the first scene of the movie)
Independent caption of the current scene segment:
{CAPTION}

C5: Memory-grounded script refinement. Following Eq. (4), the retrieved historical context is used to refine the independent script. For each current segment, every query item retrieves its top-k best-matching preceding segments via embedding similarity over memory items (Eq. (3)); the retrieved reference segments are then processed in descending order of retrieval score. In each refinement round, g_ϕ jointly watches the reference and the current segment (presented as two ordered videos) together with the reference caption, the current caption, the rolling summary R_{i-1}, and the round's query keywords, and outputs a minimally clarified current-segment caption. Rounds are applied sequentially, each taking the previous round's output as its input, and outputs whose length ratio to the input caption falls outside $[0.5, 2.0]$ are rejected. When available, the reference caption passed to g_ϕ is itself the sequentially refined version, consistent with the sequential processing order of segments.

C5: Memory-Grounded Script Refinement (g_ϕ)

[System] You are an expert in refining film annotations. The videos and plot captions of two scene segments of the same movie are provided in order: the first video and its caption are the **earlier reference segment**; the second video and its caption are the **current segment**. The current-segment caption was annotated independently and may lack historical information that is knowable only from the reference segment.
A set of "queries" extracted from the current segment (in keyword form) is also provided; they are merely retrieval clues that retrieval deems likely answerable from the reference segment.
Refinement goal
The sole goal is to **clarify** the current segment: add a small amount of necessary historical information so that character identities and event motives/causes in the current caption become explicit. It is not to pile in historical information—anything optional must not be added.
Rules
Watch both videos and read both captions carefully, and verify each query one by one:
1. Only add correlations that can be **directly confirmed** from the reference segment's video or caption: e.g., a character referred to by appearance in the current segment is confirmed to be the same person as someone in the reference video (appearance/clothing can be compared across the two videos), and his or her name/appellation/identity directly corresponds in the reference caption; or an event whose motive/cause is unstated in the current segment is exactly an event that occurred in the reference segment. When neither the reference video nor the reference

caption provides explicit evidence, adding is strictly forbidden—no commonsense inference, possibility judgment, or imagination;
2. For queries that the reference segment cannot answer (no relevant content / correlation unconfirmable), make no addition for them;
3. Add only what is necessary to clarify the query: a name, a modifier, or a brief causal link suffices—do **not retell the reference segment's events**, and do not add new sentences or length to summarize historical background; historical information should be embedded into the current segment's existing narrative rather than appended separately;
4. The addition must be natural: the final caption must read like an independently written plot introduction with no trace of fusion:
– Meta-expressions pointing to the annotation process or prior text, such as "reference segment," "preceding text," "previous plot," "previously," "historical information," or "mentioned in the previous segment," are strictly forbidden—the reader does not know the reference segment exists;
– Parenthetical annotations, footnotes, or "supplementary note"-style splicing are strictly forbidden;
– A character whose name is confirmed must be called directly by name (appearance features may naturally remain as in-sentence modifiers when necessary); do not retain long referential descriptions such as "the person wearing such-and-such clothes";
5. Modifications stay minimal: only add historical information related to this reference segment; do not rewrite the current segment's own plot, do not add or remove events of this segment, and do not introduce information beyond the two videos, the two captions, and the rolling summary;
6. Preserve the coherent narrative style and length of the original caption (6–10 sentences, no line breaks, no bullet points, no titles); the post-fusion length should be roughly the same as the original (only a few words or in-sentence components added; noticeable lengthening counts as over-fusion);
7. When quoting dialogue or words, always use standard double quotes ("").
Output format (strictly followed)
Directly output the processed complete caption text of the current segment; do not output JSON, do not wrap it in code blocks, and do not output any explanation, title, prefix, or suffix. If none of the queries can be answered by the reference segment, output the current-segment caption unchanged.
[User]
Two videos are attached in order: Video 1 = {REF_VIDEO}, the reference segment clip (Scene {REF_SCENE_META}, earlier in time); Video 2 = {CUR_VIDEO}, the current segment clip (Scene {CUR_SCENE_META}).
Background: rolling plot summary up to the segment preceding the current one (for grasping the narrative thread only; do not copy it directly into the caption)
{PREV_SUMMARY}
Reference segment caption (corresponds to Video 1, earlier in time)
{REF_CAPTION}
Current segment caption (corresponds to Video 2)
{CUR_CAPTION}
Queries to verify and resolve in this round (derived from the current segment's query items, keyword form, reference clues only)
{QUERY_LIST}
Please watch both videos and read both captions, verify each query one by one, and output the processed current-segment caption as required.

C6: Annotation refinement. This stage comprises two sub-steps executed in pipeline order: hallucination detection and correction (C6.1), followed by language unification (C6.2). In C6.1, an auditor MLLM watches the segment and reviews the final caption in two parts: content describing the current segment is verified strictly against the visuals and audio, whereas injected historical information is not penalized for being unverifiable within the current clip unless it clearly contradicts the visual or audio evidence; when hallucinations are detected, a second MLLM call performs minimal, targeted corrections, modifying only the conflicting expressions while preserving the rest of the narrative. In C6.2, a text-only model translates foreign-language (English) names and terms into the

annotation language (Chinese) under a strict literal-translation policy that forbids any modification of the remaining text.

C6.1a: Hallucination Detection (Two-Part Review)

[System] You are a professional video-annotation quality auditor, performing hallucination review of gold-standard annotations for a long-video plot-understanding benchmark.
Background
The movie has been split into scene segments. You are given the video of one scene segment together with two versions of its caption:
– **Independent caption**: produced by independent annotation, describing only what is visible and audible within this segment;
– **Final caption**: built upon the independent caption with historical information from earlier scenes injected (e.g., character names, identity backgrounds, event causes); this is the object under review.
Task
Watch the segment video and review whether the **final caption** contains hallucinations inconsistent with the actual visuals/audio of this segment. Because the final caption contains historical information, to prevent such information from interfering with the judgment, review in two parts according to the following rules:
1. **Current-segment content part** (content in the final caption that corresponds to the independent caption and describes the current segment's visuals): verify strictly item by item, with extra strictness on details—misattributed character attributes (gender, appearance, identity, clothing), writing character A's action as character B's, incorrect counts, and objects or dialogue lines that appear out of nowhere all count as hallucinations;
2. **Historical information part** (content newly added in the final caption relative to the independent caption, originating from earlier scenes): being unverifiable within the current segment's visuals is normal and **is not judged as a hallucination**; however, if the historical information **clearly conflicts** with the current segment's visuals (e.g., the historical information says "man in black coat" but the person in this segment clearly wears a white coat), it is still judged as a hallucination;
3. Do not relax the verification of current-segment content merely because the historical information "sounds plausible"; and do not judge historical information as a hallucination solely because it cannot be verified;
4. Wording differences, detail-level differences, and reasonable synonymous paraphrases are not hallucinations; report only substantive conflicts with the current segment's visuals/audio.
Output format
Output only the following JSON, with no other content:
{
"has_hallucination": true or false,
"conclusion": "one-sentence conclusion (no hallucination / what hallucination was detected)",
"analysis": "point-by-point verification analysis: which statements match the video, which do not, what exactly the mismatches are, and what the video actually shows"
}
[User]
Scene segment range: {SCENE_TIME_RANGE}
Independent caption (describes only current-segment content): {INDEPENDENT_CAPTION}
Final caption (under review, with historical information injected): {FINAL_CAPTION}
Please review in two parts as required and output the JSON.

C6.1b: Minimal Targeted Correction

[System] You are a professional final-review annotator, producing the final gold-standard annotations for a long-video plot-understanding benchmark.
Background
You are given the video of one scene segment and its plot caption. The caption has undergone independent annotation and global historical-information supplementation; the previous hallucination-detection step found it inconsistent with the video content in some places and

provided a conclusion and analysis. The narrative of the caption itself is already natural and coherent; no stylistic polishing is needed.
Task
Watch the segment video and perform minimal, targeted corrections on the caption:
1. Only modify statements that substantively conflict with the video content (using the previous detection conclusion and analysis as clues, but deferring to what you actually see and hear); keep all other content and wording unchanged, including all true information points and historical supplementary information;
2. If historical supplementary information clearly conflicts with the current segment's visuals (e.g., an appearance/object description from historical information does not match the actual visuals of this segment), correct that statement according to the current segment's visuals; historical information that cannot be verified and does not conflict with the current segment's visuals must be retained without exception;
3. Do not rewrite, do not polish the style, and do not adjust the narrative order or length; the corrected caption must keep the original coherent narrative form (no bullet points, no titles);
4. Only write content actually seen or heard in the video or supported by established historical information; speculation, invention, or external knowledge about the film is strictly forbidden;
5. When quoting dialogue or words, always use standard double quotes (””).
Output format
Output only the following JSON, with no other content:
{
”caption”: ”the corrected complete caption”
}
[User]
Scene segment range: {SCENE_TIME_RANGE}
Current caption: {FINAL_CAPTION}
Hallucination-detection conclusion: {DETECT_CONCLUSION}
Hallucination-detection analysis: {DETECT_ANALYSIS}
Please perform minimal, targeted corrections on the caption and output the JSON.

C6.2: Chinese–English Unification

[System] You are an expert in Chinese–English unification of film annotation texts. You are given the plot captions (final caption and independent caption) of a scene segment. The texts may contain English content (person names, place names, organization names, work titles, terms, etc.). Your sole task is: translate the English parts into Chinese, and keep everything else exactly as is.
Strict requirements
1. Only translate English/foreign-language content into Chinese; do not change a single character of content that is already in Chinese; rewording, polishing, summarizing, compressing, or expanding the Chinese parts is forbidden, and changing punctuation or word order is forbidden;
2. Translate literally only, without expansion: translate according to the literal English source; e.g., if the source only writes ”harry”, translate only to its Chinese transliteration, and do not complete or expand it; abbreviations must not be expanded into full names;
3. Use common, natural Chinese translated names;
4. If a text contains no English content, it must be output exactly as is.
Example
Input final caption: In the early morning, Jack walks into the forest outside Hogwarts with his backpack, looking exhausted.
Output final caption: the identical sentence with only ”Jack” and ”Hogwarts” replaced by their standard Chinese transliterations; every other character unchanged.
Incorrect: rephrasing or compressing the already-Chinese parts—forbidden.
Incorrect: expanding ”Hogwarts” into a longer full name such as ”Hogwarts School of Witchcraft and Wizardry”—forbidden.
Output format (strictly followed)
Output only the following JSON, with no other content:
{

”caption”: ”the final caption after Chinese–English unification”,
”caption_independent”: ”the independent caption after Chinese–English unification”
}
[User]
Texts of Scene {SCENE_ID} to be unified
Final caption: {FINAL_CAPTION}
Independent caption: {INDEPENDENT_CAPTION} (if it is identical to the final caption, this line states so, and the two outputs must stay consistent)
Please translate the English parts of the above texts into Chinese and output only the specified JSON.

C7: Rubric generation. As the final construction step, an MLLM watches the segment video and, conditioned on both caption versions, generates fine-grained evaluation rubrics for the final caption across the four content dimensions. Since the downstream judge sees only the caption text under evaluation and never the video, each rubric item must be a self-contained verification sentence with the concrete factual content spelled out, so that it can be checked against the caption alone; the video serves only to establish the factual basis of what truly occurs and matters in the segment. History items are derived by comparing the final caption with the independent caption, capturing the newly injected historical information points; when the two versions agree or nothing needs to be retained, the history list is empty.

C7: Rubric Generation

[System] You are an expert in constructing quality-evaluation rubrics for video captions. You are given the video of one scene segment of a movie together with two versions of its plot caption: the first is the **independent caption** (produced by independent annotation, describing only what is visible and audible within this segment); the second is the **final caption** (globally refined, possibly absorbing historical information from earlier scenes; note that caption annotations are usually redundant and may contain content of little importance to understanding this segment).
Task
Watch the segment video and, combining the two caption versions, generate evaluation rubrics for the **final caption**. During later evaluation, the judging model **sees only the caption text under evaluation and not the video**; it checks item by item whether the caption correctly and sufficiently covers the content described by each item. Therefore, the video is used only to help you determine what truly occurs and matters in this segment (the factual basis), while each rubric item must be written as a verification sentence about the caption: the item must spell out the concrete factual content (who did what, where, with what result, etc.) so that the judging model can decide, without watching the video and by checking against the caption alone, whether the item is satisfied. By content nature, items fall into four categories:
– **character**: static attributes of characters appearing in this segment: identity, name, appearance, clothing, etc.; character behaviors and actions belong to event and must not be written under character;
– **event**: important plot events occurring in this segment, e.g., cause, course, and outcome;
– **scene**: scene/environment content of this segment, e.g., location, set, time atmosphere, etc.;
– **history**: comparing the two caption versions, historical information points newly added in the final caption relative to the independent caption (supplements from earlier scenes), used specifically to evaluate whether historical information was correctly absorbed; output an empty list when the two versions agree or when nothing needs to be retained.
Selection and writing criteria (important)
1. Retain only reasonable and important evaluation items: content necessary for understanding the plot of this segment; redundant or trivial content in the annotation that does not help plot understanding (e.g., minor details that do not affect plot comprehension) must not appear in the rubrics; the total number of items should be moderately focused, with the four categories together generally no more than 12 items per segment;
2. Items must be grounded in the true content of the video: confirm content against the video; do not fabricate information present in neither the video nor the captions;

3. Each rubric item is a complete, self-contained verification sentence in the annotation language, with the concrete factual content written into the sentence (e.g., ”the caption should mention that the male lead picks up a red suitcase at the station on a rainy night”), so that the judging model, without watching the video and by checking against the caption under evaluation alone, can decide whether the item is satisfied; vague formulations that depend on watching the video, such as ”the video shows...” or ”in the frame...”, are forbidden;
4. Merge semantically duplicate items: different phrasings of the same information point keep only one;
5. Atomicity: each item contains only one atomic fact; bundling multiple facts into one item with ”and / then / simultaneously” is forbidden; information containing multiple facts must be split into multiple items, each carrying a single fact;
6. Fine-grained visual attributes (color, clothing style, posture details, etc.) are written only when they are key clues for identifying a character or advancing the plot; non-critical colors or styles must not serve as verification points (e.g., write ”fighter jet” rather than ”green fighter jet” when the color does not affect the plot).
Output format (strictly followed)
Output only JSON, with no explanation:
{
”character”: [”...”, ...],
”event”: [”...”, ...],
”scene”: [”...”, ...],
”history”: [”...”, ...]
}
[User]
Independent caption (produced by independent annotation)
{INDEPENDENT_CAPTION}
Final caption (after global refinement)
{FINAL_CAPTION}
Please watch the segment video and, combining the two caption versions, generate four-category rubrics for the final caption according to the selection and writing criteria (evaluation will check only the caption, not the video), and output the JSON.

A.5.3 Evaluation Prompts

E1: Progressive script generation (task prompt). This is the task prompt issued to every evaluated model under the progressive protocol of Section 3.1: for each window, the model segments it into scene segments, writes a detailed caption per segment, and updates its rolling plot history, which is fed back to condition the next window.

E1: Progressive Script Generation

[System] You are a professional movie plot understanding and annotation model, watching an entire movie sequentially via sliding windows. You are given one video window; complete three tasks: segment this window according to the scene-segment definition, write a detailed plot caption for each scene segment, and update the rolling plot history (history).
Definition of a scene segment
Scene segment = a continuous stretch of content with a unified spatiotemporal context: the location/set remains unchanged and stays within the same scene, from entering that scene until the scene ends (switches to another location/set).
Boundary = a scene change: the location/set switches to another scene (a time jump accompanying the scene change also counts). The following changes do NOT constitute a boundary: camera position, angle, or shot-size changes; shot/reverse-shot; editing rhythm changes; and minor entrances/exits or movements of characters within the same location.
Meaningless content does not constitute a scene segment and is skipped directly: opening credits (studio logo, film title, opening song, opening credits), ending credits (ending song, credit roll, end card), pure black/white frames, blank transition frames, and interstitial shots carrying no narrative information.
Input description

– Video: one window clip of the entire movie;
– Shot-seconds list: automatically detected shot start points within the clip (seconds relative to the clip start, ascending). Scene boundaries occur only at shot cut points; the start / end of each scene segment must be selected from this list (start may be 0, i.e., the clip beginning is the scene beginning).
Task requirements
1. Segmentation: skip meaningless content and segment the clip into sequentially ordered scene segments. If a scene has not ended at the window end (it will continue in a subsequent window), still output that segment as usual;
2. Caption: write a detailed plot caption for each scene segment (requirements below);
3. History update: based on the input history, supplement the new plot of this window and output the updated complete history. It should be appropriately concise; the content is unrestricted—output the rolling historical information according to your understanding of the video content. When there is no input history, write it directly based on this window's content.
Caption requirements
– Visual anchoring (anti-hallucination): before writing the caption of each scene segment, first identify the characters, environment, and events in the opening (first 2 seconds) and ending (last 2 seconds) frames of that segment; describe only content that actually appears in that segment, unaffected by adjacent scenes' visuals;
– Describe the complete content of the scene segment in one coherent narrative paragraph (no bullet points, no titles);
– Coverage: scene environment (location and atmosphere), characters present, course of events (who did what, why, and with what result, including key dialogue points), turning points (if any), and the ending state;
– Length 6–10 sentences with high information density; empty generalizations are rejected;
– Character appellation: names given in the video via dialogue, subtitles, or other means may be used directly; unnamed characters are referred to by concrete appearance features (e.g., "the middle-aged man in the gray suit"), and the same character's appellation stays consistent throughout;
– Conservative reference: character attributes that cannot be confirmed from the video (gender, occupation, identity relations, etc.) must not be assumed; use neutral references; only information explicitly presented in the visuals or dialogue is written into the caption;
– Write only content actually seen or heard in the video; speculation, invention, or external knowledge about the film is strictly forbidden;
– When quoting dialogue or words, always use standard double quotes ("").
Output format (strictly followed; unparseable replies are judged as failures and retried)
– Output only one legal JSON object, starting directly with { and ending with }; do not output any reasoning process or explanation before or after, and do not wrap it in markdown code blocks;
– start / end must be numbers (seconds relative to the clip start, selectable only from the shot-seconds list or 0), without quotes, units, or other text;
– String values are written in one line without line breaks; quotations inside string values must not be unescaped double quotes;
– The JSON structure must be complete: all double quotes paired, lists and brackets closed, and no trailing comma at the end of the object.
Example structure (replace field values with actual content):
{
"scenes": [{"start": seconds relative to clip start, "end": seconds relative to clip start, "caption": "detailed plot caption"}],
"history": "updated rolling history; content and form unrestricted"
}
[User]
Current window information
– Clip duration: {WINDOW_DURATION} seconds
– Shot-seconds list (relative to clip start, ascending): {SHOT_BOUNDARIES} (if no shot cut point is detected, only 0 and the clip end are selectable)
Previous plot history

{ROLLING_HISTORY}
Please watch the video and complete segmentation, per-segment captioning, and history update; output only the specified JSON.

E2: Variant prompts for diagnostic analyses. The two diagnostic analyses of Sections 4.5 and 4.7 are implemented as orthogonal, exact-string variants of Card E1, so that any performance difference can be attributed to the single manipulated factor: the history organization style (plain / categorized / timeline / none) and the golden-boundary mode. Under the plain baseline with golden boundaries disabled, the prompt is byte-identical to Card E1. When fed back to the next window, a categorized history is rendered with per-dimension section labels and a timeline history as a numbered list; the none style returns an empty history block so that windows are annotated independently.

E2.1: History Organization Variants (Section 4.5)

The history-task instruction (Task 3) and the history output field of Card E1 are replaced as follows; all other text remains byte-identical.
Variant categorized. Task 3 is replaced by: "3. History update: maintain the rolling history as a structured object that summarizes, along the following four dimensions, the accumulated information from the beginning of the movie up to this window; absorb this window's new plot on top of the input history and rewrite the whole object (the four dimensions are consistent with the evaluation criteria): – character: static information of characters who have appeared—identity, name, appearance, clothing, mutual relations; write attributes only, concrete actions belong to event; – event: important plot events that have occurred, summarized along the causal thread (cause/course/outcome); semantically duplicate content may be concisely merged; – scene: key locations, sets, and time atmospheres that have appeared; – history: cross-scene timeline and causal background, i.e., the recap needed to understand subsequent plot. Write each dimension as one appropriately concise paragraph; when there is no input history, write directly based on this window's content." The history output field is replaced by: "history": {"character": "accumulated static character information", "event": "accumulated thread of important events", "scene": "accumulated key scenes/environments", "history": "accumulated prior timeline and causal background"}.
Variant timeline. Task 3 is replaced by: "3. History update: maintain the rolling history as a chronologically ordered scene-beat list; each element summarizes one scene segment or key event in one sentence; on top of the input history list, append this window's new scene entries at the end, keeping existing entries and their order unchanged (do not rewrite or delete earlier entries), and output the updated complete list. When there is no input history, write directly based on this window's content." The history output field is replaced by: "history": ["first chronological scene beat", "second", "..."].
Variant none (no-history ablation). All history-maintenance wording and the history output field are removed: "three tasks" becomes "two tasks", the history-update task and the history clause are deleted, and the JSON example ends at the scenes list. Windows are then annotated independently with no accumulation across windows.

E2.2: Golden-Boundary Variant (Section 4.7)

To disentangle temporal localization from narrative comprehension, all "segment by yourself / select boundaries from the shot list" wording of Card E1 is replaced by "boundaries are given; fill them in verbatim"; all other text (scene-segment definition, caption requirements, output example) remains byte-identical. The four replaced passages are:
1. Task overview: "complete three tasks: process this window segment by segment according to the input given scene boundaries, write a detailed plot caption for each scene segment, and update the rolling plot history."
2. Input description: "Scene boundary list: the start / end of each scene segment already given within this window (seconds relative to the clip start, ascending, see the user message). These boundaries are fixed; you do not need to judge segmentation yourself, and the start / end of each scene segment must use the given values verbatim."

3. Task 1: ”1. Per-segment processing: the scene-segment boundaries of this window are given in the scene boundary list; output one scene segment for each given boundary, filling in start / end verbatim with the given values; do not add, remove, merge, or adjust boundaries yourself.” 4. Output format: ”start / end must be numbers (seconds relative to the clip start) and must use the values given in the scene boundary list verbatim, without quotes, units, or other text.” Correspondingly, the user message provides the scene boundary list {GOLDEN_BOUNDARY_LIST} (numbered start/end pairs) in place of the shot-seconds list, and the closing instruction becomes ”watch the video, write captions segment by segment according to the given scene boundaries, update the history, fill in start/end verbatim with the given values, and output only the specified JSON.”