AGENTDIRECTOR: A MOVIE RECAP BENCHMARK FOR LONG-HORIZON MULTIMODAL AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Existing multimodal benchmarks probe perception, reasoning, and generation in isolation, yet provide limited diagnostic insight into how these capabilities remain coordinated across long sequences of planning, tool use, and verification. We introduce AgentDirector, a 70-task benchmark that evaluates this capability through movie recap production from full-length films. The workflow couples narrative design and footage localization with narration synthesis, audiovisual editing, and inspection of intermediate results. We decompose this workflow into five settings spanning Planning, Execution, and End-to-End Production, varying supplied narration and plans, temporal requirements, and deliverables to examine these capabilities and their coordination. Our evaluation decomposes performance into Completeness, Verification, and Quality, with criteria aligned to each setting's responsibilities. These complementary components distinguish delivery failures, constraint violations, and weaknesses in narrative or audiovisual quality. The benchmark further supports systematic comparisons across agents and harnesses in task performance, editing quality, and computational cost. Component scores and execution traces help identify capability profiles and failure patterns throughout recap production.

1 INTRODUCTION

Multimodal models are increasingly integrated with external tools to build agents for long-horizon, interactive workflows (Xie et al., 2024; Li et al., 2026; Shen et al., 2023; Yang et al., 2023; Koh et al., 2024). Within these workflows, models use observations to formulate plans, invoke tools to produce intermediate artifacts, and incorporate feedback into subsequent actions (Yao et al., 2022). However, much of multimodal evaluation still examines perception (Li et al., 2023; Liu et al., 2024b; Tong et al., 2024), reasoning (Johnson et al., 2017; Lu et al., 2024; Yue et al., 2024), and generation (Ghosh et al., 2023; Huang et al., 2024; Liu et al., 2024a) as separate capabilities. Evaluating how agents integrate these capabilities across long sequences of interdependent actions therefore remains an important open problem (Liu et al., 2025).

Movie recap production provides a concrete setting for examining these capabilities in combination: agents must use their understanding of a feature-length film to create a coherent audiovisual narrative. The task combines five demands: long-context understanding, narrative planning and footage localization, coordination across production stages, inspection of intermediate outputs to guide revision, and delivery of technically valid, semantically faithful results. The challenge is to keep these capabilities coordinated throughout production so that the narrative, selected footage, and final audiovisual output remain consistent.

To evaluate this coordination under different task responsibilities and guidance conditions, we introduce AgentDirector, a 70-task benchmark with five settings (Figure 1). It complements existing production benchmarks such as AgenticVBench (Cao et al., 2026) by varying the editorial responsibilities assigned to agents within a shared workflow. The settings span three complementary task families. Planning requires agents to generate narration and an editing plan from the film, without producing a video, assessing film understanding and narrative planning. Execution supplies an editing plan and requires agents to realize it as a video excerpt, assessing footage retrieval and audiovisual synchronization. End-to-End Production requires agents to develop their own editing plan and produce a complete recap, assessing coordination across the full production workflow.

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



Figure 1: AgentDirector organizes movie recap production into three task families and five settings. Planning authors narration and an editing plan; Execution realizes a supplied plan as a short recap; and End-to-End Production coordinates planning and execution under a prescribed orchestrator. Settings vary source-timestamp requirements and whether narration is supplied.

The responsibilities and deliverables in each family also determine what constitutes successful completion. A technically valid video may contain footage that does not support its narration, whereas a well-written script may never be realized as the required video. An aggregate score alone can obscure these distinctions. We therefore assess three components: Completeness for delivery of required artifacts, Verification for compliance with verifiable constraints, and Quality for content and audiovisual quality. Within Quality, Content assesses narrative correctness, Sourcing assesses footage localization and alignment with narration, and Production assesses narrative effectiveness and audiovisual presentation. Planning tasks use only Content within Quality, while video tasks use all three dimensions.

Building on these task settings and evaluation criteria, we compare agent–harness configurations across planning, execution, and end-to-end production, and use score decomposition to examine delivery, constraint compliance, and output quality. In From-Scratch Production, we further study performance–resource tradeoffs and conduct pairwise comparisons of completed recaps. These experiments reveal that system strengths vary across production responsibilities, while autonomous full production remains challenging. A score decomposition further shows that high completeness and constraint compliance can coexist with limited narrative accuracy and footage support. Efficiency analysis reveals runtime differences at similar scores, while editing-quality evaluations expose differences between composite rankings and preferences for completed videos.

Our contributions are threefold:

- **A benchmark for long-horizon multimodal coordination.** We introduce AgentDirector, with 70 movie recap tasks across three families and five settings. Varying editorial responsibilities, localization requirements, and deliverables enables assessment of understanding, planning, audiovisual execution, and their integration.
- **Score decomposition aligned with task responsibilities.** We develop a protocol that separates Completeness, Verification, and Quality, with quality further assessed through Content, Sourcing, and Production where applicable. Aligning criteria with each setting's deliverables distinguishes delivery failures, constraint violations, and weaknesses in content, footage use, and presentation.
- **A protocol for system comparison and failure analysis.** We provide a protocol for comparing agent–harness configurations through component scores and execution records to characterize capability profiles and production failures. For From-Scratch Production, the protocol combines resource measurements and pairwise recap judgments to examine editing quality in relation to computational cost.

2 Related Work

2.1 Multimodal and Agent Benchmarks

Output requirements shape what multimodal benchmarks reveal. Video question-answering benchmarks assess event understanding (Tapaswi et al., 2016; Fu et al., 2025), temporal reasoning (Xiao et al., 2021), and information integration across video moments (Wu et al., 2024). Video-generation evaluation measures visual quality, temporal consistency, and alignment with text prompts (Huang et al., 2024; Liu et al., 2024a; He et al., 2024). These tasks characterize understanding or output quality. Interactive evaluation further examines how understanding informs subsequent decisions. OSWorld and VisualWebArena evaluate agents acting on observations in interactive environments (Xie et al., 2024; Koh et al., 2024); VisualAgentBench also examines high-level planning and feedback-driven revision (Liu et al., 2025). AgentDirector extends this direction to consistency across source films, editing plans, and finished recaps. Plans and videos provide complementary evidence of whether understanding and planning are realized in production. Relating intermediate decisions to final artifacts offers a diagnostic perspective on multimodal coordination: whether capabilities remain effective across dependent stages and where their integration breaks down.

2.2 Agentic Video Editing and Production

Video production tasks differ in the guidance supplied, the decisions left to the system, and the required deliverables. MCSC-Bench evaluates shot-level scripts specifying material selection and narration (Hu et al., 2026), while TeaserGen combines narration generation with documentary footage selection for teasers (Xu et al., 2025). MVEBench, introduced with GLANCE, varies prompt specificity and distinguishes on-beat from story-driven editing (Lin et al., 2026). Production capability must therefore be interpreted relative to each task's guidance and responsibilities. These responsibilities also determine the evidence needed for evaluation. Extending delivery from plans to finished videos requires assessing both the realization of editorial requirements and the resulting audiovisual quality. AgenticVBench includes narrated recaps and combines programmatic format checks with expert visual, narrative, and sound rubrics (Cao et al., 2026); Crayotter separates verifiable execution feedback from final-product preferences (Yan et al., 2026). Within a shared movie recap workflow, AgentDirector varies supplied narration and plans, source-film localization requirements, and deliverables, and aligns evaluation with the resulting responsibilities. Linking task decomposition to evaluation evidence makes performance interpretable in terms of planning and execution demands, supporting analysis of capability requirements, coordination, and failure modes across the workflow.

3 Task Formulation and Benchmark Construction

3.1 Task Setup and Production Workflow

Given a source film V of duration T_V, the agent must produce an editing plan P, a rendered recap, or both, as specified by the task. Every setting provides the full film, while supplied narration and editing plans determine which decisions remain for the agent. Across settings, production follows the shared workflow

$$\textit{perceive} \rightarrow \textit{plan/localize} \rightarrow \textit{act} \rightarrow \textit{verify} \rightarrow \textit{deliver}.$$

The agent first inspects V to identify characters, plot events, and candidate scenes, then creates or follows a segment-level plan while preserving any supplied narration. We represent the plan as an ordered sequence of n segments,

$$P = \big((\nu_i, v_i, a_i, d_i)\big)_{i=1}^{n}, \tag{1}$$

where ν_i specifies the narration, v_i the intended visuals, a_i the treatment of voiceover, original sound, music, and effects, and d_i the target clip length. For film-footage entries, the visual description and story context guide retrieval from V. When required, the agent records the selected source interval

$$\tau_i^V = [x_i, y_i), \qquad 0 \le x_i < y_i \le T_V. \tag{2}$$

where x_i and y_i are the start and end times in V. Thus, P specifies the intended recap, while τ_i^V identifies the source material used to realize it. Added material, such as title cards, is specified separately. The plan schema is detailed in Appendix C.2.

The agent then records P for planning tasks or realizes it as video by extracting footage, synthesizing voiceover, aligning subtitles, applying the audio instructions, and assembling the segments in order. Before delivery, it checks the plan and intermediate artifacts against the film and task requirements; any revision preserves supplied narration and plan constraints. The agent finally submits the plan, the rendered recap, or both, together with the supporting records listed in Appendix C.4.

3.2 Benchmark Construction

We construct AgentDirector from 14 full-length source films at 360p paired with human-edited Chinese recaps annotated by ten experts. Detailed dataset statistics, including temporal coverage and cross-setting film reuse, are reported in Appendix C.1. These annotated recaps are used only during construction. Speech recognition first divides each recap into narration segments with text and recap-timeline boundaries (Radford et al., 2023). A multimodal model then analyzes the recap and transcript to reconstruct an editing plan in the representation of Equation 1, recording the narration, visual description, audio instructions, and target clip length d_i of each segment. Recap-timeline boundaries determine d_i; the corresponding source-film intervals τ_i^V remain to be located in V.

We validate each reconstructed plan for required fields, segment structure, duration consistency, repeated annotations, end-credit content, and narration language. After validation, we remove the recap-timeline boundaries while retaining the information needed for planning and footage retrieval. The sanitized plan, source film, and setting-specific guidance form the task instances described in Section 3.3; each instance fixes the source-film version so that τ_i^V uses a consistent timeline. Appendix C.2 specifies the representation, and Appendix C.3 details the construction checks.

3.3 Task Families and Settings

AgentDirector defines five settings in three families (Table 1). The settings specify which narration and plan information is supplied, whether source-film intervals must be reported, and whether the agent submits a plan, a video, or both.

Planning. Given only the film, the agent selects events, writes narration, and produces an editing plan without rendering a video. Required-Timing Planning requires source-film start and end times as in Equation 2, whereas Flexible-Timing Planning makes these timestamps optional. Both require specific shot descriptions and story context supported by the film.

Execution. In Plan-Conditioned Execution, the agent receives the film and a supplied editing plan with narration for an excerpt, typically ten consecutive segments covering 60–90 seconds. The plan describes the desired shots, clip lengths, and audio treatment but omits their source-film locations. The agent must retrieve matching footage, report its source intervals, and produce the excerpt with voiceover, subtitles, and audio as specified. The full source film remains available for retrieval.

End-to-End Production. Both settings require an editing plan and a complete rendered recap. In Narration-Conditioned Production, the agent receives the film and narration but no editing plan; it must preserve the supplied text, plan the visuals and audio, locate footage, and complete the edit. In From-Scratch Production, only the film is supplied, without a title, script, or plan, so the agent also chooses the narrative and writes the narration. In From-Scratch Production and both planning settings, a coherent narrative may differ from the human recap used during benchmark construction.

4 Evaluation Protocol

We evaluate agent submissions against each setting's delivery requirements through three complementary components. Completeness (C) measures delivery of required outputs and supporting artifacts; Verification (V) measures compliance with verifiable task constraints; and Quality (Q) assesses content correctness and, where applicable, footage use and overall presentation. Evaluation uses submitted plans, videos, and supporting records, together with fixed instance-specific references and scoring rubrics. Scoring combines programmatic checks, independent media measurements, and model judgments. Applicable criteria follow each setting's requirements and deliverables. We report both component and composite scores to distinguish incomplete delivery, constraint violations, and weaknesses in content or editing quality. Detailed evaluation procedures are provided in Appendix D.

Table 1: **Input and output requirements for the five settings in AgentDirector.** ✓: supplied input or required output; (✓): optional; —: not supplied or required. Supporting artifacts are listed in Appendix C.4.

Setting	Input			Output				Tasks
	Film	Narration	Plan	New narration	New plan	Source times	Video	
Planning								
Required-Timing Planning	✓	—	—	✓	✓	✓	—	10
Flexible-Timing Planning	✓	—	—	✓	✓	(✓)	—	10
Execution								
Plan-Conditioned Execution	✓	✓	✓	—	—	✓	✓	20
End-to-End Production								
Narration-Conditioned Production	✓	✓	—	—	✓	✓	✓	20
From-Scratch Production	✓	—	—	✓	✓	✓	✓	10
Total								**70**

4.1 Completeness

For the component definitions below, we consider a single trial and omit setting, instance, and trial indices until Section 4.4. All three scores lie in $[0, 1]$.

Completeness measures delivery of the required result and supporting materials. The primary deliverable is a parseable editing plan for Planning, or a rendered recap for Execution and End-to-End Production. Let $h \in \{0, 1\}$ indicate whether this primary requirement is met. For the setting's checklist $\mathcal{D}$, let $z_k \in \{0, 1\}$ indicate satisfaction of requirement k with positive weight w_k^{C}. The completeness score is

$$\mathsf{C} = h \frac{\sum_{k \in \mathcal{D}} w_k^{\mathsf{C}} z_k}{\sum_{k \in \mathcal{D}} w_k^{\mathsf{C}}}. \tag{3}$$

An absent or invalid primary deliverable gives the trial zero, while other missing materials or structural inconsistencies receive weighted deductions. This is the sole explicit hard gate, making valid delivery a prerequisite for receiving credit for quality.

4.2 Verification

Verification measures compliance through deterministic checks on submitted records and media. Structural compliance v_{str} is a weighted mean of applicable checks on structure, timing, and obligations such as preserving supplied narration. For video tasks, media compliance measures decodability and duration, subtitle constraints, integrated loudness, cut density, and speech rate. Available media operators $\mathcal{M}$ contribute equally:

$$m = \frac{1}{|\mathcal{M}|} \sum_{\ell \in \mathcal{M}} m_\ell, \qquad \mathsf{V} = \begin{cases} 0.8\, v_{\text{str}} + 0.2\, m, & \text{video settings,} \\ v_{\text{str}}, & \text{planning settings.} \end{cases} \tag{4}$$

Here $m_\ell \in [0, 1]$ are the media scores; unavailable checks are omitted. Keeping measurable requirements in this layer allows quality judgments to focus on meaning and editorial effect without re-scoring the same technical conditions. Structural checks, media target ranges, and applicability are detailed in Appendix D.3.

4.3 Quality

Quality assesses narrative correctness, footage support, and overall editorial effect through three complementary dimensions: Content, Sourcing, and Production. Their normalized scores are q_{content}, q_{sourcing}, and $q_{\text{production}}$, respectively, each in $[0, 1]$.

Content. The content score evaluates the submitted narration and editing plan against binary criteria derived from human-revised film annotations. These criteria cover plot events, character identities and relationships, and causal connections. A model judge assigns pass, fail, or unverifiable decisions, which are aggregated as detailed in Appendix D.4.1.

Sourcing. The sourcing protocol combines temporal localization with footage–narration consistency. Reference annotations identify key events and their source-film time ranges. A model judge matches selected segments to events; deterministic overlap and coverage measures then score their recorded source intervals. Consistency evaluates rendered segments paired with narration for semantic support and temporal agreement. The two assessments receive equal weight (Appendix D.4.2).

Production. The production score captures the viewing experience of the complete video through six aspects: interpretive insight, narrative engagement, organization and allocation of detail, writing quality, audiovisual presentation, and voiceover expressiveness. Each aspect has four scoring levels, and a multimodal judge selects the level that best describes the finished work. These ratings characterize overall editorial quality; technical compliance is assessed in Verification.

For video settings with all three dimensions available, the quality factor is

$$\mathsf{Q} = 0.50\, q_{\text{content}} + 0.30\, q_{\text{sourcing}} + 0.20\, q_{\text{production}}. \tag{5}$$

Unavailable dimensions are omitted and the remaining weights renormalized. These weights prioritize content correctness and supporting footage while retaining an assessment of the finished work. Both planning settings use $\mathsf{Q} = q_{\text{content}}$, aligning assessment with their required deliverable (Table 3). Appendix D.4 describes the dimension-level scoring and normalization procedures.

4.4 Composite Score and Repeated Trials

For trial r of instance j in setting s, we restore these indices on the three component scores and denote the overall score by $\mathsf{S}_{jr}^{(s)} \in [0, 1]$:

$$\mathsf{S}_{jr}^{(s)} = \mathsf{C}_{jr}^{(s)} \mathsf{V}_{jr}^{(s)} \mathsf{Q}_{jr}^{(s)}. \tag{6}$$

The product makes quality credit depend jointly on delivery completeness and constraint satisfaction. Each instance has up to three independent trials. Let $R_j^{(s)} \in \{1, 2, 3\}$ count attempted trials, including failed deliveries. With N_s instances in setting s, we report

$$\text{Score@3}(s) = \frac{1}{N_s} \sum_{j=1}^{N_s} \frac{1}{R_j^{(s)}} \sum_{r=1}^{R_j^{(s)}} \mathsf{S}_{jr}^{(s)}. \tag{7}$$

The component product is computed within each trial, then averaged across trials and instances. Undelivered trials remain in the mean with zero scores, so the aggregate reflects both output quality and delivery across repeated runs. Comparisons are made within each setting, whose requirements and applicable quality dimensions are held fixed. Component scores and execution records support analysis of the weaknesses behind the aggregate.

5 Experiments and Analysis

5.1 System Setup and Evaluation Procedure

Within each setting, all model–harness configurations share task instances, source-film versions, supplied materials, output requirements, and scoring criteria (Section 4). Each configuration runs three independent trials per instance. We average scores first within instances, then equally across instances to obtain Score@3 (Equation 7). Execution settings and scoring procedures are detailed in Appendices B and D, respectively.

5.2 Overall Performance

Table 2 reports Score@3 for each setting and the unweighted mean across all five. **(1) Aggregate rankings conceal task-specific weaknesses.** Claude Opus 5 leads four settings and averages

Table 2: Overall performance of the evaluated model–harness configurations across five settings. RTP: Required-Timing Planning; FTP: Flexible-Timing Planning; PCE: Plan-Conditioned Execution; NCP: Narration-Conditioned Production; FSP: From-Scratch Production. Full setting definitions are in Table 1. Bold indicates the highest score in each column.

Model	Harness	Planning		Execution	End-to-End Production		Mean
		RTP	FTP	PCE	NCP	FSP	
Claude Opus 5 (Anthropic, 2026b)	Claude Code	**0.377**	0.366	**0.723**	**0.708**	**0.424**	**0.520**
Qwen3.8-Max (Qwen Team, 2026)	Claude Code	0.363	**0.402**	0.709	0.706	0.297	0.496
GPT-5.6 Sol (OpenAI, 2026)	Codex	0.295	0.298	0.710	0.689	0.351	0.469
Qwen3.8-Omni-Flash (Alibaba Cloud, 2026)	Claude Code	0.243	0.319	0.720	0.687	0.367	0.467
Claude Opus 4.8 (Anthropic, 2026a)	Claude Code	0.301	0.335	0.677	0.695	0.324	0.466
GLM-5.3 (Z.ai, 2026)	Claude Code	0.333	0.389	0.526	0.620	0.344	0.442
Kimi K3 (Moonshot AI, 2026)	Claude Code	0.302	0.329	0.486	0.409	0.308	0.367
Qwen3.8-27B (Qwen Team, 2026)	Claude Code	0.135	0.188	0.654	0.607	0.249	0.366

Table 3: Score decomposition of Claude Opus 5 by setting. Completeness (C) measures artifact coverage and Verification (V) measures compliance with verifiable constraints. Quality includes its total score and applicable dimensions. A dash denotes an inapplicable dimension.

Setting	Completeness	Verification	Quality				Overall
			Content	Sourcing	Production	Total	
Required-Timing Planning	0.933	0.920	0.409	—	—	0.409	0.377
Flexible-Timing Planning	0.933	0.929	0.394	—	—	0.394	0.366
Plan-Conditioned Execution	0.967	0.956	0.944	0.579	0.552	0.756	0.723
Narration-Conditioned Production	1.000	0.971	0.942	0.489	0.558	0.729	0.708
From-Scratch Production	0.962	0.854	0.466	0.420	0.655	0.489	0.424

0.520. Qwen3.8-Max ranks second overall (0.496) and leads Flexible-Timing Planning (0.402), yet ranks seventh in From-Scratch Production (0.297). **(2) Planning strength does not ensure execution strength.** GLM-5.3 outperforms Qwen3.8-Omni-Flash in Flexible-Timing Planning (0.389 versus 0.319) but trails in Plan-Conditioned Execution (0.526 versus 0.720), distinguishing narrative planning from the retrieval and audiovisual execution needed to realize a plan. **(3) Autonomous full production remains a shared challenge.** All eight systems score lower in From-Scratch Production than in Narration-Conditioned Production under their respective criteria; the best from-scratch score is 0.424, versus 0.708 for the same configuration in Narration-Conditioned Production.

Low from-scratch scores highlight the difficulty of coordinating understanding, planning, localization, and execution. Varying guidance, temporal requirements, and deliverables separates these responsibilities and reveals model strengths and weaknesses obscured by aggregate scores.

5.3 Score Decomposition

Table 3 shows that Claude Opus 5's high Completeness (0.933–1.000) and Verification (0.854–0.971) coexist with distinct quality profiles across settings.

(1) Narrative accuracy remains weak despite compliance. Both planning settings exceed 0.9 in Completeness and Verification, yet Content scores only 0.409 and 0.394. These factual, character, and causal criteria expose limitations in autonomous narration and planning before rendering. Settings preserving supplied narration score 0.942–0.944 in Content, versus 0.466 in From-Scratch Production; these higher scores do not establish stronger autonomous understanding.

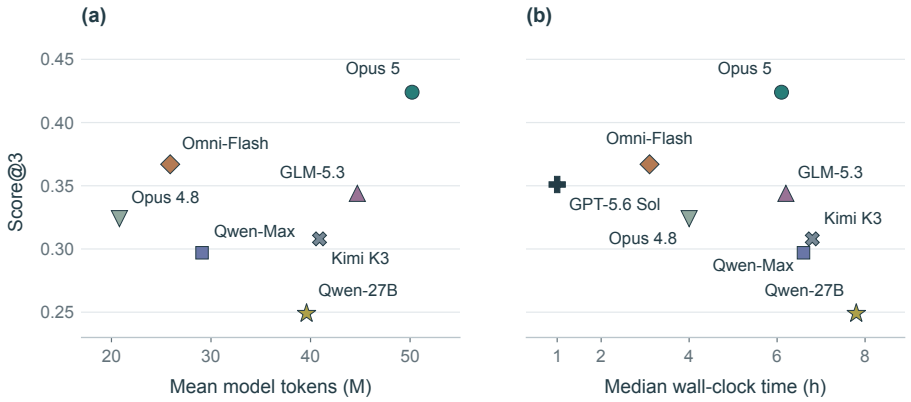


Figure 2: **Efficiency and performance in From-Scratch Production.** Score@3 versus (a) mean tokens and (b) median wall-clock time of successful runs. Qwen abbreviates Qwen3.8.

(2) Supplied narration does not resolve footage grounding. Plan-Conditioned Execution and Narration-Conditioned Production score 0.579 and 0.489 in Sourcing, which assesses event coverage, temporal localization, and footage–narration agreement. The latter requires autonomous visual planning across a full recap. From-Scratch Production scores 0.420, another 0.069 lower. Sourcing thus remains limited as responsibilities broaden, although different instances and requirements prevent causal attribution.

(3) Stronger presentation can coexist with weaker fidelity. From-Scratch Production scores 0.655 in Production, above Plan-Conditioned Execution (0.552) and Narration-Conditioned Production (0.558), despite lower Content and Sourcing. Between the two full-production settings, the content gap (0.476) dominates the quality difference under its 50% weight: the production advantage (0.097) cannot offset it, leaving quality totals of 0.489 versus 0.729. Verification is also lower (0.854 versus 0.971), alongside overall scores of 0.424 versus 0.708. Narrative accuracy and compliance remain concerns alongside footage support.

5.4 Efficiency and Performance

Figure 2 compares performance, mean token consumption, and median runtime in From-Scratch Production. **(1) Similar scores conceal runtime differences.** GPT-5.6 Sol and GLM-5.3 score 0.351 and 0.344, but their median runtimes are 1.0 and 6.2 h, respectively. Claude Opus 5 scores 0.424, exceeding GPT-5.6 Sol by 0.073 at 6.1 times its median runtime. **(2) Fewer tokens do not imply faster completion.** Qwen3.8-Max uses 42.0% fewer tokens than Claude Opus 5 (29.1 M versus 50.2 M) and scores 0.127 lower, yet takes longer (6.6 h versus 6.1 h). **(3) Greater resource use need not yield higher scores.** Qwen3.8-27B uses more tokens and time than Qwen3.8-Omni-Flash but scores lower (0.249 versus 0.367). Runtime medians include successful runs only; timeout rates are 3% for Claude Opus 5 and GLM-5.3 and 0% for others. GPT-5.6 Sol lacks token and call data, restricting its comparison to runtime.

5.5 Pairwise and Human Editing-Quality Evaluation

Experimental setup. In From-Scratch Production, `gemini-3.8-flash` evaluates all 28 pairs of eight models using every available cross-model trial combination within each instance. Videos are anonymous and presented in randomized A/B order; the judge selects a winner without ties. Missing videos are skipped, and rankings use mean win rates against seven equally weighted opponents. Human evaluators rank anonymous videos from all ten instances over three rounds. Each instance–model combination's trial slots are randomly assigned without replacement across rounds. Failed or absent slots receive bottom ranks, averaged for ties. Mean ranks across 30 groups determine the human ordering (lower is better). Details are in Appendices D.6 and D.7.

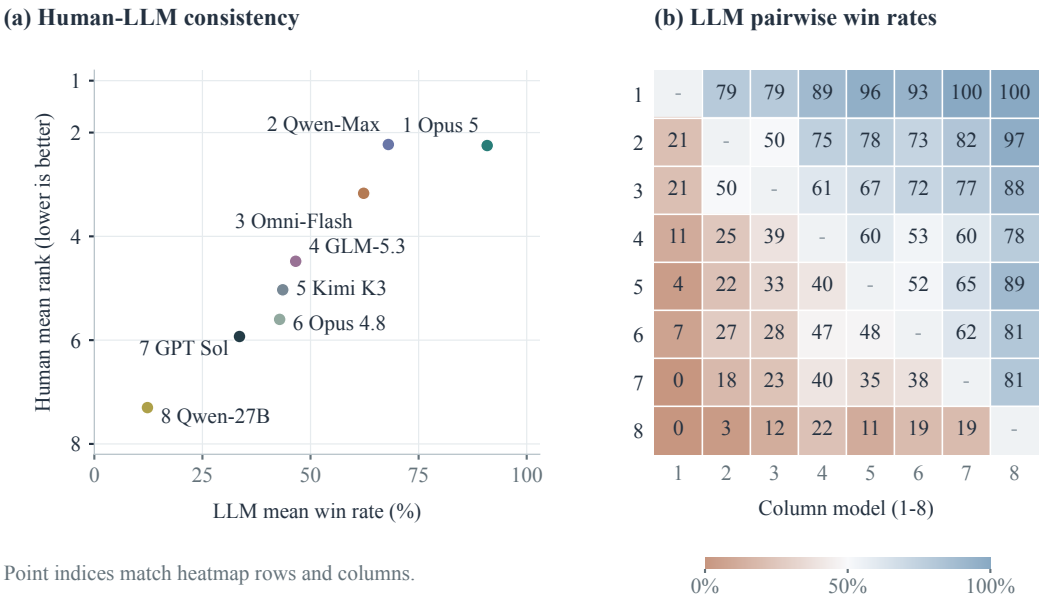


Figure 3: **LLM and human editing-quality evaluation in From-Scratch Production.** (a) LLM mean pairwise win rate versus mean human rank over 30 groups (rank axis reversed). Point indices identify models in (b). (b) Row-model win percentages against column models across 28 pairs; self-comparisons are excluded. Missing videos are skipped by LLM comparisons and assigned bottom ranks (averaged for ties) by humans.

Experimental results and analysis. Figure 3 compares LLM pairwise preferences and human rankings. **(1) Claude Opus 5 is preferred across opponents.** Its mean win rate (0.91) exceeds those of Qwen3.8-Max (0.68) and Qwen3.8-Omni-Flash (0.62), with 79%–100% against every opponent. The Qwen pair splits direct comparisons 50%–50%, so their mean gap reflects other opponents. **(2) Composite rankings differ from video preferences.** Qwen3.8-Max ranks seventh by from-scratch score (0.297) but second in pairwise preferences (0.68); GPT-5.6 Sol ranks third by score (0.351) but seventh in pairwise preferences (0.34). Video preferences complement scores incorporating delivery, compliance, and reference-based content and footage assessments. **(3) Human rankings closely agree with the LLM ordering.** The final rankings have Spearman correlation $\rho = 0.976$, differing only in the leading pair. Qwen3.8-Max and Claude Opus 5 have mean ranks of 2.23 and 2.25, yet Claude Opus 5 receives 22 of 30 first-place rankings versus seven for Qwen3.8-Max, distinguishing average standing from first-choice frequency. Agreement concerns aggregate rankings under different missing-output rules.

6 Conclusion

We introduced AgentDirector to address the limited evaluation of how multimodal agents coordinate understanding, planning, and execution throughout long production workflows. The benchmark uses movie recap production from full-length films, with 70 tasks across five settings spanning Planning, Execution, and End-to-End Production. By varying supplied guidance and production responsibilities, these settings examine individual capabilities and their integration. Our evaluation protocol separates delivery completeness, constraint compliance, and content and audiovisual quality, enabling diagnosis beyond aggregate scores. Experiments with eight model–harness configurations show that system strengths vary across production stages and that autonomous full production remains difficult. Score decomposition of the leading configuration reveals that high completeness and compliance can coexist with limited narrative accuracy and footage support. Resource analysis reveals runtime differences at similar scores, while editing-quality evaluations distinguish composite rankings from video preferences. These findings highlight maintaining fidelity to the source film across narrative planning, footage selection, and final production as a central challenge for current multimodal agents.

AI USE STATEMENT

We used generative AI to reconstruct segment-level editing plans from human-edited movie recaps during benchmark construction (Section 3.2). We also evaluated generative AI agents and used model judges to assess narrative correctness, footage-narration consistency, and production quality (Sections 4.3 and 5.1). We additionally used generative AI to assist in drafting and refining parts of the manuscript. We validate reconstructed plans through structural, timing, and language checks (Appendix C.3). We take responsibility for the final content of this work, including AI-assisted text, annotations, and artifacts.

REPRODUCIBILITY STATEMENT

Experimental details are provided in Section 5 and Appendix B. The evaluation protocol and scoring procedures are described in Section 4 and Appendix D. We plan to publicly release the AgentDirector benchmark and code upon acceptance.

REFERENCES

Alibaba Cloud. Alibaba Cloud Model Studio: Qwen3.8-Omni-Flash, September 2026. URL https://www.alibabacloud.com/help/en/model-studio/qwen3-8-omni-flash. Model documentation. Updated September 18, 2026; accessed September 25, 2026.

Anthropic. Claude Opus 4.8 system card. Technical report, Anthropic, May 2026a. URL https://www.anthropic.com/claude-opus-4-8-system-card.

Anthropic. System card: Claude Opus 5. Technical report, Anthropic, July 2026b. URL https://www.anthropic.com/claude-opus-5-system-card.

Zongheng Cao, Yi Zheng, Rui Song, and Xinyu Hu. Agenticvbench: Can ai agents complete real-world post-production tasks? *arXiv preprint arXiv:2605.27705*, 2026.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24108–24118. IEEE, 2025.

Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. *Advances in Neural Information Processing Systems*, 36: 52132–52152, 2023.

Xuan He, Dongfu Jiang, Ge Zhang, Max Ku, Achint Soni, Sherman Siu, Haonan Chen, Abhranil Chandra, Ziyan Jiang, Aaran Arulraj, et al. Videoscore: Building automatic metrics to simulate fine-grained human feedback for video generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 2105–2123, 2024.

Huanran Hu, Zihui Ren, Dingyi Yang, Liangyu Chen, Qixiang Gao, Tiezheng Ge, and Qin Jin. Mcsc-bench: Multimodal context-to-script creation for realistic video production. *arXiv preprint arXiv:2604.15127*, 2026.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. Vbench: Comprehensive benchmark suite for video generative models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21807–21818. IEEE, 2024.

Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2901–2910, 2017.

Jing Yu Koh, Robert Lo, Lawrence Jang, Vikram Duvvur, Ming Lim, Po-Yu Huang, Graham Neubig, Shuyan Zhou, Russ Salakhutdinov, and Daniel Fried. Visualwebarena: Evaluating multimodal agents on realistic visual web tasks. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 881–905, 2024.

Ke Li, Maoliang Li, Jialiang Chen, Jiayu Chen, Zihao Zheng, Shaoqi Wang, and Xiang Chen. Direct: Video mashup creation via hierarchical multi-agent planning and intent-guided editing. *arXiv preprint arXiv:2604.04875*, 2026.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 292–305, 2023.

Zihao Lin, Haibo Wang, Zhiyang Xu, Siyao Dai, Huanjie Dong, Xiaohan Wang, Yolo Y Tang, Yixin Wang, Qifan Wang, and Lifu Huang. Glance: A global-local coordination multi-agent framework for music-grounded non-linear video editing. *arXiv preprint arXiv:2604.05076*, 2026.

Xiao Liu, Tianjie Zhang, Yu Gu, Iat Long Iong, Song XiXuan, Yifan Xu, Shudan Zhang, Hanyu Lai, Jiadai Sun, Xinyue Yang, et al. Visualagentbench: Towards large multimodal models as visual foundation agents. In *International Conference on Learning Representations*, volume 2025, pp. 95650–95707, 2025.

Yaofang Liu, Xiaodong Cun, Xuebo Liu, Xintao Wang, Yong Zhang, Haoxin Chen, Yang Liu, Tieyong Zeng, Raymond Chan, and Ying Shan. Evalcrafter: Benchmarking and evaluating large video generation models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22139–22149. IEEE, 2024a.

Yuliang Liu, Zhang Li, Mingxin Huang, Biao Yang, Wenwen Yu, Chunyuan Li, Xu-Cheng Yin, Cheng-Lin Liu, Lianwen Jin, and Xiang Bai. Ocrbench: on the hidden mystery of ocr in large multimodal models. *Science China Information Sciences*, 67(12):220102, 2024b.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. In *International Conference on Learning Representations*, volume 2024, pp. 23439–23554, 2024.

Moonshot AI. Kimi K3: Open frontier intelligence, July 2026. URL https://www.kimi.com/en/blog/kimi-k3. Official technical blog, July 16, 2026.

OpenAI. GPT-5.6, 2026. URL https://openai.com/index/gpt-5-6/.

Qwen Team. Qwen3.8-Max: A new bar for coding and cowork, August 2026. URL https://qwen.ai/blog?id=qwen3.8. Official release blog.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *International conference on machine learning*, pp. 28492–28518. PMLR, 2023.

Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face. *Advances in Neural Information Processing Systems*, 36:38154–38180, 2023.

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

Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. Eyes wide shut? exploring the visual shortcomings of multimodal llms. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9568–9578. IEEE, 2024.

Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. Longvideobench: A benchmark for long-context interleaved video-language understanding. *Advances in Neural Information Processing Systems*, 37:28828–28857, 2024.

Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. Next-qa: Next phase of question-answering to explaining temporal actions. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9772–9781. IEEE, 2021.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh J Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, et al. Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments. *Advances in Neural Information Processing Systems*, 37:52040–52094, 2024.

Weihan Xu, Paul Pu Liang, Haven Kim, Julian McAuley, Taylor Berg-Kirkpatrick, and Hao-Wen Herman Dong. Teasergen: Generating teasers for long documentaries. In *International Conference on Learning Representations*, volume 2025, pp. 75558–75578, 2025.

Lecheng Yan, Jianze Lin, Yichong Zhang, Ben Pan, Wenxi Li, Chenyang Lyu, Liting Zhou, and Cathal Gurrin. Crayotter: Learning long-horizon video editing agents via group-relative preference backpropagation. *arXiv preprint arXiv:2608.02694*, 2026.

Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Ehsan Azarnasab, Faisal Ahmed, Zicheng Liu, Ce Liu, Michael Zeng, and Lijuan Wang. Mm-react: Prompting chatgpt for multimodal reasoning and action. *arXiv preprint arXiv:2303.11381*, 2023.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9556–9567, 2024.

Z.ai. GLM-5.3: Frontier coding with emergent cyber capabilities, August 2026. URL https://z.ai/blog/glm-5.3. Official release blog, August 14, 2026.

A Additional Experiments

A.1 Effect of the Harness on the Same Model

We compare Claude Code and Qwen Code while keeping the underlying model fixed. Table 4 reports results for Qwen3.8-27B and Qwen3.8-Omni-Flash across the two planning settings and Plan-Conditioned Execution, including overall scores, score components, and delivery rates. Qwen Code results for the two full-production settings are unavailable, so the comparison is limited to these three settings.

Table 4: **Within-model comparison of Claude Code and Qwen Code.** Score components follow Table 3. Claude Code overall scores are from Table 2. RTP: Required-Timing Planning; FTP: Flexible-Timing Planning; PCE: Plan-Conditioned Execution. Only settings with Qwen Code results are shown.

Setting	Harness	Completeness	Verification	Quality				Overall	Delivery (%)
				Content	Sourcing	Production	Total		
Qwen3.8-27B									
RTP	Claude Code	0.567	0.554	0.244	—	—	0.244	0.135	57
	Qwen Code	0.933	0.897	0.299	—	—	0.299	0.268	93
FTP	Claude Code	0.767	0.757	0.247	—	—	0.248†	0.188	77
	Qwen Code	0.995	0.990	0.334	—	—	0.334	0.328	100
PCE	Claude Code	0.914	0.898	0.933	0.523	0.545	0.732	0.654	92
	Qwen Code	0.989	0.978	0.937	0.505	0.547	0.729	0.705	100
Qwen3.8-Omni-Flash									
RTP	Claude Code	0.800	0.769	0.318	—	—	0.318	0.243	80
	Qwen Code	0.900	0.890	0.327	—	—	0.328†	0.292	90
FTP	Claude Code	0.867	0.860	0.371	—	—	0.371	0.319	87
	Qwen Code	0.867	0.866	0.333	—	—	0.333	0.288	87
PCE	Claude Code	0.981	0.965	0.937	0.568	0.548	0.748	0.720	98
	Qwen Code	0.990	0.983	0.921	0.570	0.548	0.741	0.724	100

A dash denotes an inapplicable dimension. Delivery measures primary-output completion, whereas Completeness also considers supporting artifacts. †Reported values retained: planning uses $\mathsf{Q} = q_{\text{content}}$, but Content and Total differ by 0.001 for 27B/Claude Code in FTP and Omni-Flash/Qwen Code in RTP; these entries require verification.

Qwen3.8-27B benefits most in planning and delivery. With Qwen Code, the overall score rises from 0.135 to 0.268 in RTP and from 0.188 to 0.328 in FTP, gains of 0.133 and 0.140, respectively. Delivery rates increase from 57% to 93% and from 77% to 100%, gains of 36 and 23 percentage points; both planning content scores also rise. In execution, the overall score increases from 0.654 to 0.705 (+0.051), while delivery rises from 92% to 100%. This gain accompanies higher completeness and verification, despite a slightly lower quality total (0.732 to 0.729). The largest score and delivery gains occur in planning; the execution improvement does not imply better quality across all dimensions.

Qwen3.8-Omni-Flash shows mixed changes. Qwen Code raises the overall score in RTP from 0.243 to 0.292 (+0.049), leaves execution nearly unchanged (0.720 to 0.724, +0.004), and lowers FTP from 0.319 to 0.288 (-0.031). Delivery increases from 98% to 100% in execution and from 80% to 90% in RTP, while remaining at 87% in FTP. In flexible planning, content drops from 0.371 to 0.333 despite unchanged completeness (0.867) and slightly higher verification (0.860 to 0.866). In execution, higher completeness and verification coexist with lower quality (0.748 to 0.741), yielding little overall change. Harness benefits therefore depend on the model and task; these descriptive comparisons do not establish benefits for full production.

A.2 TOOL-USE PATTERNS

Figures 4 and 5 summarize tool-use frequency and composition for the seven models with available proxy logs. Counts are reported per run. GPT-5.6 Sol (Codex) is excluded because its proxy logs are unavailable.

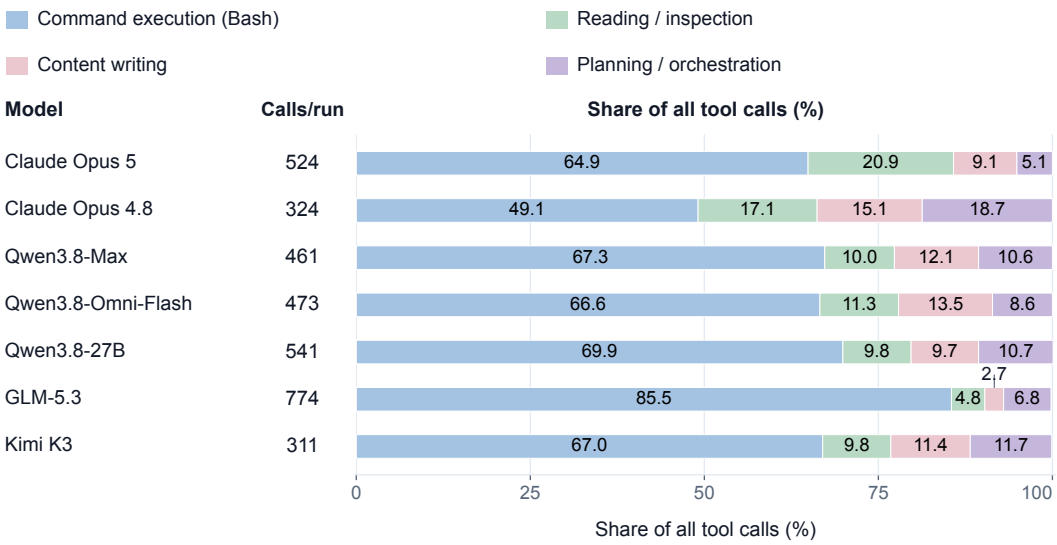


Figure 4: **Distribution of tool calls across four main categories.** Bar segments show percentages of all tool calls; the count column reports total calls per run. Percentages retain their original denominator and supplied precision without renormalization.

GLM-5.3 has the highest tool-call frequency (774 calls per run) and command-execution share (85.5%). Claude Opus 4.8 has the lowest command-execution share (49.1%) and the highest planning/orchestration share (18.7%). Python execution (29.0–36.5%) and frame extraction (20.6–24.9%) are the two largest displayed Bash categories for every model.

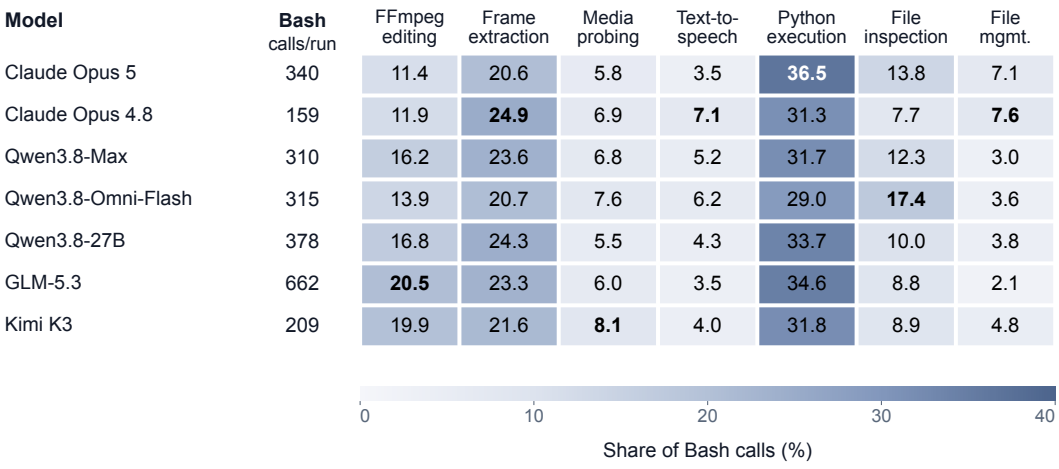


Figure 5: **Breakdown of Bash calls.** Cells show percentages of all Bash calls, and bold marks the highest share in each column. Whisper, environment/dependency operations, and other commands are omitted, each accounting for less than 1.5% of Bash calls. Displayed percentages retain their original denominator without renormalization.

B EXPERIMENTAL DETAILS

This appendix reports the public, reproducibility-relevant execution settings used for Table 2. We treat each row as a *model–harness configuration*, rather than attributing differences to the model alone, because Claude Code and Codex use different agent runtimes and model interfaces. The settings below were verified against the run manifests associated with the evaluated groups. Private service endpoints, credentials, internal revision identifiers, group identifiers, and machine-specific paths are intentionally omitted.

All experiments were executed with Harbor, which provided isolated containers, fixed resource limits, and independent per-trial workspaces. Claude Code or Codex served as the agent runtime within each Harbor job.

B.1 FROZEN TASKS AND REPEATED TRIALS

We launched three independent trials for every fixed task instance. Thus, each configuration submitted 120 jobs for the 40 instances spanning the two Planning settings and Plan-Conditioned Execution, and 90 jobs for the 30 instances spanning Narration-Conditioned Production and From-Scratch Production. Trials use separate workspaces and do not share intermediate plans, selected clips, or generated media.

All experiments use frozen, harness-specific packages of the five evaluation settings. Each package fixes its instance identifiers, source media, instructions, reference evidence, and evaluator implementation. The packages preserve the same task and evaluator contract while adapting agent configuration and orchestration to each harness. Results are audited against the frozen package recorded for each run; pilot runs and runs using other task versions are not substituted.

All reported full runs used the same container image and requested four CPU cores, 12,288 MB of memory, a 46,800-second wall-clock limit, and at most 500 agent iterations per trial. Streaming and complete message logging were enabled. These limits bound the agent process; calls made by media-analysis tools and the evaluation procedure remain separate from the tested model's generation trace.

B.2 MODEL–HARNESS CONFIGURATIONS

Table 5 summarizes the public interfaces and generation settings that differ across evaluated systems. It reports only the official upstream model APIs; deployment-specific gateways and endpoint details are omitted.

Table 5: **Public execution configurations used in Table 2.**

Model(s)	Agent harness	Official model API	Generation settings
Claude Opus 5 Claude Opus 4.8	**Claude Code** 2.1.153	Anthropic Messages API	Thinking: adaptive Effort: maximum Temperature: 1.0 Context: 1,000,000 Max. output: 128,000
Qwen3.8-Max Qwen3.8-Omni-Flash Qwen3.8-27B GLM-5.3 Kimi K3	**Claude Code** 2.1.153	Alibaba Cloud Model Studio API	Thinking: passthrough Temperature: 1.0 Max. output: 32,768
GPT-5.6 Sol	**Codex** 0.154.0	OpenAI Responses API	Reasoning effort: high Temperature: not set Max. output: not set

Context and output limits are harness-declared token limits. "Not set" means that the run manifest did not specify an explicit value.

Claude Code. All Claude Code runs use version 2.1.153. Claude Opus 5 and Claude Opus 4.8 use the Anthropic Messages API with adaptive thinking at maximum effort, a declared 1,000,000-token context limit, a 128,000-token output limit, and temperature 1.0. The Qwen, GLM, and Kimi models are connected to Claude Code through an API-compatibility layer backed by the public Alibaba Cloud Model Studio API. These configurations use passthrough thinking, a 32,768-token output limit, and temperature 1.0. The declared limits characterize the harness configuration, not necessarily the effective usable context.

For Narration-Conditioned Production and From-Scratch Production, Claude Code exposes the planner and executor as named agents: the orchestrator first invokes a planning agent and then one or more execution agents. The two Planning settings and Plan-Conditioned Execution use one agent process and do not require this planner–executor orchestration.

Codex. GPT-5.6 Sol uses Codex version 0.154.0, the `gpt-5.6-sol` model identifier, and the OpenAI Responses API. Its run manifests set reasoning effort to `high`. They do not set an explicit temperature or output token limit, so we do not infer either value from example configuration files.

For Narration-Conditioned Production and From-Scratch Production, the same planner and executor roles are expressed as Codex agent configurations and invoked through the Codex-native agent interface. This changes capability discovery and invocation relative to Claude Code, but not the required task artifacts or evaluation criteria. The two harnesses did not expose directly comparable token and API-call accounting in these runs. Consequently, Figure 2 reports runtime for GPT-5.6 Sol but omits its token and call totals.

B.3 Run Validation and Reporting Policy

For every reported configuration, we validated the model identifier, agent version, dataset split, trial count, and expected job count against the corresponding run manifest. Each submitted group contained all expected jobs with no submission failures. Subsequent execution or delivery failures remain attached to their original trial identities and are not replaced by pilot runs or trials from another group. The benchmark release will include sanitized configurations containing the public parameters needed to reproduce these runs.

C Benchmark Details

C.1 Dataset Composition and Scope

Task allocation and reuse. The 70 task instances derive from 14 distinct films. Plan-Conditioned Execution and Narration-Conditioned Production each contain 20 tasks from the same seven films, reusing the selected excerpt windows across the two settings. Both planning settings and From-Scratch Production each contain ten tasks, one per film, from the same ten-film pool. The two pools share three films, which therefore appear in all five settings. The instances thus form 14 film-level clusters rather than 70 independent film samples; statistical uncertainty should account for this reuse. Repeated trials vary agent execution but do not increase film coverage. Cross-setting score differences also reflect partially different film pools and task scopes.

Film coverage and language. The films were released between 1986 and 2018, with most titles from the 2000s, and span drama, thriller, animation, action, fantasy, and romance. All task instructions and recap narration are in Chinese. Source-film dialogue is predominantly non-Chinese: eight films are primarily in English, three in Japanese, and one each in Chinese, Italian, and Arabic. The benchmark therefore evaluates Chinese recap production from multilingual source material.

Source-film temporal statistics. We count each film once and summarize its latest source-film timestamp in the reference annotations. Across the 14 films, these endpoints range from 86.4 to 239.9 minutes, with a mean of 122.6 minutes and a median of 118.9 minutes. Seven films (50%) have reference timestamps beyond 120 minutes, and the longest annotated timeline reaches nearly four hours. Table 6 gives the distribution. Each task fixes the source-film version (Section 3.2).

Table 6: Distribution of the latest reference timestamp per film. Each of the 14 distinct films is counted once, irrespective of task reuse.

Time range (min)	$[85, 100)$	$[100, 120)$	$[120, 140)$	$[140, 160)$	≥ 160
Films	4	3	5	1	1

C.2 Editing Plan Schema

Editing plans describe a recap as an ordered sequence of segments, each linking narration with visual and audio instructions. The annotations obtained through reverse planning also identify the project and the human recap being analyzed. Agent-generated plans use the same representation, with timing requirements and deliverables determined by the task setting. Table 7 summarizes the segment fields. Planning tasks export the plan, while production packages retain it together with execution records; the required outputs are listed in Appendix C.4.

Table 7: Core fields in the shared editing representation. Timing metadata has a different role from semantic retrieval cues.

Field or group	Meaning
`segment_id`	Stable segment identifier within the ordered plan
`narration`	Narration text and estimated voiceover duration
`content_source`	Movie clip, title card, advertisement, commentary, transition, or graphic
`target_shot_description`	Concrete characters, actions, and locations visible in the intended shot
`shot_type`, `camera_movement`	Shot scale and camera motion
`clip_length_hint`	Agent-visible description of the intended clip length
`movie_locator`	Relative story position, scene anchor, visual cues, nearby event, and confidence
`audio_plan`	Voiceover requirement, original-audio policy, music, effects, and guidance
`source_audio`	Audio observed in the human recap, including type, transcript, language, and confidence
`source_segment_timing`	Input-segment timing in reverse planning; removed from execution inputs
`target_duration_sec`	Numeric target length in the construction record, rendered as a hint during sanitization

Visual retrieval cues. For film footage, the locator combines a relative narrative position, a scene anchor, a nearby story event, and three to six observable visual attributes. Narrative position is recorded as opening, early, middle, late, ending, or unknown, and confidence as high, medium, or low. The scene and event descriptions provide narrative context for retrieval. No locator is assigned to non-film content. Reverse-planned locators exclude generated source-film timestamps.

Audio semantics. Audio instructions specify whether voiceover is required and whether to preserve ambience, retain dialogue, or mute the original soundtrack. The annotations separately describe the audio heard in the human recap: commentary narration, movie dialogue, mixed audio, music or ambience, no audio, or an unknown type. The observed transcript is derived from that recap and may differ from the corresponding source-film soundtrack.

Illustrative segment. As a hypothetical example, a segment pairs the narration (translated into English) “She enters the cabin and discovers that it is empty” with a static medium shot of a woman in a red coat opening a wooden cabin door and looking inside. The estimated voiceover duration is 3.5 seconds, and the target clip length is about 4.2 seconds. The locator places the scene early in the story, during the first cabin visit after arrival in town and following the walk from the village. It lists the red coat, wooden door, and empty cabin as visual cues, with medium confidence. The audio instructions require voiceover, preserve ambient sound, add light background music, and use no sound effects. The target duration is a task requirement; no source-film timestamp is supplied.

Timing provenance. Reverse-planning records use coordinates on the human recap. In Required-Timing Planning and both production settings, output timestamps instead specify the source-film intervals selected by the agent.

C.3 Construction Checks

We use six heuristic checks to identify structurally inconsistent or formulaic plans during benchmark construction. Each check applies only when the necessary information is available; temporal checks therefore exclude segments without optional timestamps.

Table 8: Heuristic checks used in benchmark construction. The conditions and thresholds are defined below.

ID	Target	Condition inspected
H1	Constant confidence	Identical locator-confidence labels across a sufficiently long plan
H2	Formulaic duration	Nearly constant differences between clip and voiceover durations
H3	Credits intrusion	Use of the end-credit region, explicit credit flags, or credit references
H4	Duration outliers	Excessively long source spans relative to their associated voiceover
H5	Metadata consistency	Segment counts and total-duration declarations inconsistent with the plan
H6	Language compliance	Insufficient Chinese-character coverage in recap narration

H1: constant confidence. For a plan with at least five segments, identical confidence labels for all visual locators trigger rejection. This is a heuristic against repeated defaults; a varied label distribution does not establish calibrated confidence.

H2: constant offset. Using the target clip length d_i from Equation 1 and the estimated voiceover duration $\widehat{d}_i^{\mathrm{vo}}$, we define $\Delta_i = d_i - \widehat{d}_i^{\mathrm{vo}}$. For at least five segments, a range $\max_i \Delta_i - \min_i \Delta_i \leq 0.05$ seconds triggers rejection. This targets formulaic duration assignment.

H3: credits intrusion. The check estimates the start of end credits from chapter markers and black-frame analysis. It rejects more than ten seconds of overlap with the credits region beyond a 30-second grace window. An explicit end-credit annotation or a narration reference to the credits also triggers rejection, since the task excludes end credits.

H4: span-to-voiceover outliers. For segments with an applicable source span and voiceover duration, a span exceeding eight times the voiceover duration triggers rejection. Ratios greater than three and at most eight produce warnings. The ratio check excludes segments without voiceover.

H5: inconsistent metadata. The declared segment count must equal the number of segments in the plan. The declared total output duration must be within 20% of the sum of per-clip durations. These checks detect mismatches between project metadata and the plan it describes.

H6: language compliance. All instances require Chinese narration. The Han-character ratio must be at least 0.8 both per segment and over the full narration. Quoted Latin spans are excluded before measurement. This check enforces the benchmark's narration language requirement; it does not measure fluency or narrative quality.

C.4 Required Task Artifacts

C.4.1 Planning

Both Planning settings require the following outputs:

- `movie_editing_plan.json`: the editing plan.
- `project_meta.json`: the movie title, segment count, estimated output duration, and creative rationale.
- `narration_script.md`: the full narration in segment order.

- `watch_notes/`: exactly five non-empty viewing-note files.
- `drafts/`: exactly two non-empty intermediate drafts.
- A self-check receipt, accepted at `output/evaluation/report.json` or as `report.json` in the exported artifact root.

Plans must follow the schema, contain all required content, and include at least ten segments. In Required-Timing Planning, segments must provide numeric start and end times with the end later than the start. These timestamps are optional in Flexible-Timing Planning. Viewing notes and drafts document the recorded process.

C.4.2 EXECUTION

In addition to the final recap, Plan-Conditioned Execution requires six structured files:

- `execution_results.json`: segment-level localization outcomes.
- `audio_mix_plan.json`: the intended treatment of the audio layers.
- `adapted_execution_plan.json`: the plan as adapted during execution.
- `unresolved_segments.json`: segments not successfully resolved.
- `segment_timeline.json`: placement of segments in the output.
- `concat_manifest.txt`: inputs to the concatenation operation.

Six workspace directories retain the intermediate media: `selected_clips/`, `candidates/`, `voiceovers/`, `bgm/`, `audio_mix/`, and `temp/`. The exported package excludes `selected_clips/`, `candidates/`, `temp/`, and `bgm/`. The rough-cut video is the primary deliverable. An empty unresolved-segment list reports that all segments were resolved.

C.4.3 END-TO-END PRODUCTION

Both End-to-End Production settings require a complete recap, an editing plan, and the execution records that connect selected source intervals and rendered segments to the assembled output. In Narration-Conditioned Production, the plan must preserve the supplied narration while specifying its visual and audio realization. In From-Scratch Production, the system additionally authors the narration from the source film. The final video covers the complete assigned production. Both settings require delegation to planning and execution agents. The recorded trajectory retains planner and executor invocation markers to document this handoff.

D EVALUATION DETAILS

This appendix specifies the evidence and scoring procedures underlying Completeness (C), Verification (V), and Quality (Q). The video settings comprise Plan-Conditioned Execution, Narration-Conditioned Production, and From-Scratch Production; both Planning settings submit plans without rendered videos. Their inputs and required artifacts follow Table 1 and Appendix C.4. Unless otherwise stated, scores below refer to one trial and lie in $[0, 1]$.

D.1 EVALUATION INPUTS AND REFERENCE EVIDENCE

Submitted evidence. Each trial is evaluated from its submitted artifacts. Planning provides narration and an editing plan; video tasks additionally provide the rendered recap and production records, including source intervals, output timelines, and subtitle and voiceover metadata. Delivery and structural checks inspect the package; media checks and audiovisual judgments inspect the rendered output.

Reference evidence. Human-revised film-understanding annotations supply two references prepared before evaluation: factual and relational criteria for assessing the submitted narration and plan, and key events with source-film intervals for matching and scoring selected footage.

Judging procedure. Completeness and verification use deterministic checks of the submitted artifacts. `gemini-3.8-flash` judges content, footage–narration consistency, production quality, and pairwise preferences. It also semantically matches selected segments to reference events before deterministic temporal scoring. The judge receives the relevant submission and reference criteria or event descriptions; the complete annotation record and model identity are withheld.

D.2 COMPLETENESS

The evaluator inspects the output package for the setting-specific artifacts listed in Appendix C.4. The primary deliverable is a parseable plan for planning tasks and a playable rendered video for video tasks. Missing primary output, or a video that cannot be decoded, sets $h = 0$ in Equation 3, giving the trial zero composite credit. Other artifact requirements contribute through the weighted checklist in that equation. This is the sole explicit hard gate; neither verification nor quality introduces an additional threshold that overrides the composite score. Their numerical scores may nevertheless equal zero.

D.3 VERIFICATION

Structural compliance. The inputs are the submitted production records, including segment lists, narration records, and timelines. Deterministic checks assess required fields, internal consistency, valid ordering and timing, and obligations such as preserving supplied narration. For applicable structural checks $\mathcal{A}_{\text{str}}$, with scores u_k and weights $w_k > 0$, the structural score is

$$v_{\text{str}} = \frac{\sum_{k \in \mathcal{A}_{\text{str}}} w_k u_k}{\sum_{k \in \mathcal{A}_{\text{str}}} w_k}. \tag{8}$$

Checks that cannot be evaluated are marked unavailable and omitted from this denominator. In Flexible-Timing Planning, timestamp-dependent checks are omitted when optional timestamps are absent.

Media compliance. The evaluator analyzes the rendered video together with its subtitle and voiceover metadata. Five operators inspect stream validity, subtitle statistics, integrated loudness, cut density, and speech rate (Table 9). Stream validity is binary. The remaining operators assign full credit within a target range and decrease linearly toward zero outside it. Unavailable operators, including subtitle checks without subtitle metadata, are excluded from the mean.

For a two-sided full-credit interval $[a, b]$ with bandwidth $B = b - a > 0$, the score for measurement x is

$$H(x; a, b) = \max\left\{0,\ 1 - \frac{2\max\{a - x,\ 0,\ x - b\}}{b - a}\right\}. \tag{9}$$

The score reaches zero half a bandwidth beyond either boundary, at $a - B/2$ and $b + B/2$. Thus, loudness receives zero at or below -29 LUFS or at or above -9 LUFS, and speech rate receives zero at or below 225 or at or above 525 characters/min.

Table 9: Media-compliance inputs and full-credit conditions. Continuous operators use linear decay outside the target range; stream validity uses a binary decision.

Operator	Evidence	Full-credit condition
Stream validity	Decoding and stream metadata	Decodable video, required audio/video tracks, and positive duration
Subtitle constraints	Subtitle text and timing	5–10 characters/s and at most 20 characters per line
Integrated loudness	Rendered audio track	-24 to -14 LUFS
Cut density	Detected scene changes	8–40 cuts/min
Speech rate	Narration character count and voiceover duration	300–450 characters/min

Let $\mathcal{M}$ contain the available media operators. Their scores m_ℓ are averaged equally, and verification combines structural and media compliance as

$$m = \frac{1}{|\mathcal{M}|} \sum_{\ell \in \mathcal{M}} m_\ell, \qquad \mathsf{V} = \begin{cases} 0.8\, v_{\text{str}} + 0.2\, m, & \text{video settings,} \\ v_{\text{str}}, & \text{planning settings.} \end{cases} \tag{10}$$

Planning excludes all operators requiring rendered media. Technical measurements are assigned to verification so that model judgments can focus on meaning and editorial effect without re-scoring these thresholds.

D.4 Quality

D.4.1 Content

Evidence and decisions. The judge receives the submitted narration, editing plan, and binary criteria derived from the reference annotations. Each criterion tests a verifiable factual or relational claim. Using these materials and its knowledge of the film, the judge returns pass, fail, or unverifiable. Unverifiable decisions are excluded from the scored items. The eight categories and their fixed importance weights are listed in Table 10.

Table 10: Content categories and fixed weights reflecting their importance to accurate film understanding. Each category contains film-specific binary criteria. Percentages are shown to two decimal places; the aggregation normalizes their sum.

ID	Category	Assessment target	Weight
C9	Information attribution	Distinguishing objective facts from beliefs, claims, and inferences, preserving attribution and certainty.	15.90%
C1	Key factual details	Correct ages, locations, quantities, objects, states, and actions.	14.87%
C7	Plot discovery and causal links	Preserving how plot information is discovered through clues, revelations, and the events that establish them.	14.23%
C3	Character motivation and intent	Correct goals, intentions, and motives underlying characters' decisions and actions.	13.61%
C8	Causal explanation	Correct explanations of why events occur and how causes or conditions lead to their outcomes.	13.08%
C4	Action attribution	Assigning each action to the character who performs it.	12.81%
C2	Core-plot fidelity	Avoiding contradictions of central characters, events, and outcomes, distinguishing contradiction from omission.	8.15%
C10	Character identity distinction	Keeping characters and references to their identities distinct without confusing individuals.	7.34%

Category scores and fixed weights. Let n_k^+ and n_k^- be the numbers of pass and fail decisions in category k. Its score is

$$c_k = \frac{n_k^+}{n_k^+ + n_k^-}. \tag{11}$$

Let ω_k be the fixed importance weight of category k in Table 10, shared by the systems being compared. Over categories with available scores $\mathcal{K}_{\text{content}}$, content quality is

$$q_{\text{content}} = \frac{\sum_{k \in \mathcal{K}_{\text{content}}} \omega_k c_k}{\sum_{k \in \mathcal{K}_{\text{content}}} \omega_k}. \tag{12}$$

All verifiable items within a category contribute equally. Category weights reflect the importance of each aspect of film understanding and remain fixed across submissions.

D.4.2 Sourcing

Temporal localization. The inputs are reference key events with source-film intervals and the recorded source intervals of footage selected for each narration segment. The judge first associates

selected segments with reference events semantically. Subsequent calculations operate on the source-film timeline, rather than the output timeline.

For a matched selected interval τ_i^V and reference interval τ_{ref}^V, define temporal intersection-over-union and reference coverage as

$$\text{IoU}(\tau_i^V, \tau_{\text{ref}}^V) = \frac{|\tau_i^V \cap \tau_{\text{ref}}^V|}{|\tau_i^V \cup \tau_{\text{ref}}^V|}, \qquad \text{IoR}(\tau_i^V, \tau_{\text{ref}}^V) = \frac{|\tau_i^V \cap \tau_{\text{ref}}^V|}{|\tau_{\text{ref}}^V|}. \tag{13}$$

Here $|\cdot|$ is temporal duration. Let $\mathcal{E}$ be the reference event set and $\mathcal{E}_{\text{match}}$ the matched events. Coverage breadth is $b = |\mathcal{E}_{\text{match}}|/|\mathcal{E}|$. Within the matched events, let $\overline{\text{IoR}}$ and $\overline{\text{IoU}}$ be the mean overlap measures, and h_5 the proportion of matched events whose temporal centers differ by at most 5 seconds. For any nonempty interval collection $\mathcal{T}$, its center is the midpoint of its temporal envelope:

$$\text{ctr}(\mathcal{T}) = \frac{\min_{[x,y)\in\mathcal{T}} x + \max_{[x,y)\in\mathcal{T}} y}{2}. \tag{14}$$

Let $\mathcal{T}_e$ and $\mathcal{R}_e$ contain the selected and reference intervals associated with matched event e. The hit rate is

$$h_5 = \frac{1}{|\mathcal{E}_{\text{match}}|} \sum_{e\in\mathcal{E}_{\text{match}}} \mathbf{1}[|\text{ctr}(\mathcal{T}_e) - \text{ctr}(\mathcal{R}_e)| \le 5\,\text{s}]\,. \tag{15}$$

For a single interval this is the midpoint difference. For disjoint intervals, the envelope extends from the earliest start to the latest end; internal gaps do not affect its center. This envelope convention defines the hit distance. Localization combines breadth and quality as

$$q_{\text{loc}} = b\left(0.5\,\overline{\text{IoR}} + 0.3\,\overline{\text{IoU}} + 0.2\,h_5\right). \tag{16}$$

Unmatched events reduce breadth and are excluded from matched-event quality averages. This avoids penalizing an unmatched event again in the quality term. With a nonempty reference set and no matched events, the localization score is zero.

Footage–narration consistency. The output is divided at narration-segment boundaries. For each segment, the judge receives its narration text and rendered clip, including the narration audio. It checks three aspects: coverage of the people, objects, actions, and events mentioned; absence of irrelevant footage or padding; and temporal agreement between visual evidence and narration, including premature revelations or delayed visual support.

Let $g_i \in \{0, 1\}$ be the segment decision. The segment passes only if all three aspects pass. Any failure, or a critical defect such as footage from the wrong film, a black screen, or non-source-film material, gives $g_i = 0$. With output duration $d_i^{\text{out}} > 0$ and evaluated segment set $\mathcal{I}$, consistency is the duration-weighted pass rate

$$q_{\text{con}} = \frac{\sum_{i\in\mathcal{I}} d_i^{\text{out}} g_i}{\sum_{i\in\mathcal{I}} d_i^{\text{out}}}. \tag{17}$$

These durations refer to the rendered output, not selected source spans or target lengths in the plan.

Sourcing aggregation. The sourcing score is the equal-weight mean of the two assessments:

$$q_{\text{sourcing}} = \frac{q_{\text{loc}} + q_{\text{con}}}{2}. \tag{18}$$

D.4.3 PRODUCTION

The judge views the complete rendered video and assigns one of four ordered levels to each of six aspects: interpretive insight, narrative engagement, organization and allocation of detail, writing quality, audiovisual presentation, and voiceover expressiveness. Audiovisual presentation concerns production polish and the emotional fit between sound and image; measurable media constraints remain in verification.

For level $L_k \in \{1, 2, 3, 4\}$ and applicable aspects $\mathcal{K}_{\text{prod}}$, the trial-level score is

$$q_{\text{production}} = \frac{1}{|\mathcal{K}_{\text{prod}}|} \sum_{k\in\mathcal{K}_{\text{prod}}} \frac{L_k - 1}{3}. \tag{19}$$

The levels map to 0, $1/3$, $2/3$, and 1, with equal weight across applicable aspects. Audiovisual presentation is marked inapplicable when audio is unavailable.

D.5 Missing Scores and Trial Aggregation

Both planning settings use $\mathsf{Q} = q_{\text{content}}$. For video settings, the base weights of content, sourcing, and production are 0.50, 0.30, and 0.20. If a dimension has not been evaluated, available dimensions are renormalized using these base weights. Writing $\mathcal{D}_Q$ for available dimensions and α_d for their base weights gives

$$\mathsf{Q} = \frac{\sum_{d \in \mathcal{D}_Q} \alpha_d q_d}{\sum_{d \in \mathcal{D}_Q} \alpha_d}. \tag{20}$$

With all three dimensions available, this reduces to Equation 5. A missing assessment carries no separate quality penalty under this rule; a missing primary deliverable still receives zero through completeness.

The composite is computed within each trial as $\mathsf{S} = \mathsf{CVQ}$. For each model, setting, and instance, up to three independent trials are included. Their composites are averaged within the instance, followed by an equal-weight mean across instances, as in Equation 7. Trials that fail to deliver remain in the denominator with zero scores; they are not removed as missing assessments. Component means, including production ratings, are descriptive summaries and are not multiplied after averaging to obtain the composite. The cross-setting mean in Table 2 assigns equal weight to the five setting scores.

D.6 Pairwise Editing-Quality Assessment

Pairwise evaluation complements absolute production ratings in From-Scratch Production. The protocol compares all $\binom{8}{2} = 28$ model pairs within each source-film instance. For a model pair, all available trials on one side are compared with all available trials on the other side. Three available videos per model therefore give nine comparisons per model pair and instance. Missing videos are skipped in this procedure.

Each comparison presents only videos A and B, without model identities or separate narration text. Their order is randomized, and outputs are mapped back to model identities after judging. Visible A/B labels identify the two videos, and the prompt includes an instruction against position bias. The same judge selects a winner on ten dimensions and on an overall judgment, with no tie option.

For models a and b, let W_{ab} be the number of valid overall judgments favoring a, and N_{ab} the number of valid comparisons. The pairwise win rate is

$$p_{ab} = \frac{W_{ab}}{N_{ab}}, \qquad N_{ab} > 0. \tag{21}$$

Dimension-specific win rates are computed analogously. The overall ranking uses the unweighted mean against the seven other models:

$$r_a = \frac{1}{7} \sum_{b \neq a} p_{ab}. \tag{22}$$

Self-comparisons are excluded, and each opponent receives equal weight regardless of its number of valid comparisons. Because missing videos are skipped, these rates describe preferences among available outputs; delivery reliability remains represented in the composite score.

D.7 Human Ranking Assessment

Human ranking assessment evaluates the eight models in From-Scratch Production across ten instances and three rounds, giving $G = 30$ instance–round groups. For each instance i and model m, a random permutation π_{im} of its three trial slots is drawn. Round $r \in \{1, 2, 3\}$ selects slot $\pi_{im}(r)$, so each slot is used exactly once across the three rounds of an instance. Sampling includes slots without a completed video to retain failed or absent trials in the assessment.

Within each group, the eight models are randomly assigned anonymous labels A–H. The label-to-model correspondence is stored separately and withheld from assessors, who rank the available videos by overall preference. Selected slots without a completed video are treated as missing outputs and

placed below all available videos. If k outputs are missing in a group, they share the average of the final k ranks:

$$r_{\text{missing}} = \frac{(9-k)+8}{2} = \frac{17-k}{2}. \tag{23}$$

With six available videos and two missing outputs, each missing output receives rank 7.5.

Let r_{gm} denote the rank of model m in group g. Each group receives equal weight when computing the mean rank and normalized score:

$$\bar{r}_m = \frac{1}{G}\sum_{g=1}^{G} r_{gm}, \qquad s_m = \frac{8-\bar{r}_m}{7}, \qquad G = 30. \tag{24}$$

Models are ordered by ascending mean rank; lower mean ranks and higher normalized scores indicate better performance. Agreement with the LLM evaluation is assessed using Spearman and Kendall rank correlations between the final model orderings. These statistics describe agreement at the aggregate model level under different missing-output rules: human assessment assigns bottom ranks, whereas LLM pairwise assessment skips missing videos. They do not measure agreement on individual videos or judgments.