TIMELINE-BENCH: EVALUATING AGENTS ON REALISTIC VIDEO-EDITING TASKS, FROM RAW FOOTAGE TO FINAL CUT

Anonymous authors
Paper under double-blind review

ABSTRACT

AI agents increasingly carry out long-horizon professional work, but their evaluations rarely require a finished creative deliverable. To this end, we introduce TIMELINE-BENCH, a benchmark of 56 real video-editing tasks, each asking an agent to turn raw production material into a finished video. Tasks range from selecting dialog takes and shaping interview footage into a story to cutting commercials from product shots, voiceovers and graphics. Every task provides a brief, source assets, a container and a set of tests. A task is resolved when the output passes every test. The tests check the delivery format, the content and the brief's explicit requirements, and include a quality test calibrated on 2,582 blind judgments by 43 video editors. We evaluate 16 agents that pair frontier models with coding-agent harnesses such as Codex, Claude Code and OpenCode. The best, GPT-6 Astra in Codex with curated editorial guidance, resolves only 15 of the 56 tasks (26.8%), and the average agent resolves 14.0%. Human editors prefer the reference edit in 83.5% of judgments. Most unresolved runs (562 of 771) fail only the quality test: agents perceive footage through stills and transcripts and check their renders for defects, not craft. We release the tasks, verifier and per-run results at https://timelinebench.vercel.app.

1 INTRODUCTION

Recent frontier models have substantially advanced the ability of AI agents to perform software engineering, scientific computing, and professional knowledge work (OpenAI, 2026a; Anthropic, 2026b). Models such as GPT-6 Astra and Claude Fable 5.1 combine stronger reasoning with improved computer use and sustained problem solving, enabling agents to carry out complex workflows across software applications. Systems such as Codex and Claude Code provide the tools to execute code, inspect intermediate results, and iteratively refine their work (OpenAI, 2025; Anthropic, 2026a). These capabilities also extend to creative applications: for example, OpenAI demonstrates Astra modeling a house in Blender and turning it into an interactive scene in Unreal Engine (OpenAI, 2026a). As agents take on a wider range of professional work, benchmarks must assess their ability to complete realistic workflows and produce useful deliverables. Evaluations such as GDPval, Agents' Last Exam, and AutomationBench reflect this growing emphasis (Patwardhan et al., 2026; Sun et al., 2026; Shepard & Salimans, 2026).

Video editing presents a demanding environment for such an evaluation. Used in filmmaking, advertising, education, and digital media, it requires interpreting a brief, understanding source footage, and coordinating picture, speech, music, and graphics over time. These decisions are interdependent: changing a shot can alter the meaning of the accompanying narration, while changing its duration can affect pacing and synchronization. Although broad professional benchmarks include media-related tasks (Patwardhan et al., 2026; Sun et al., 2026), their aggregate results provide limited insight into these editorial capabilities. A focused evaluation is therefore needed to assess whether agents can turn raw audiovisual material and a brief into a coherent finished video.

In this paper, we introduce TIMELINE-BENCH, a benchmark for evaluating agents on complete video-editing assignments, from raw production material to finished video. TIMELINE-BENCH assesses whether agents can carry out workflows encountered in filmmaking, documentary production,

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.

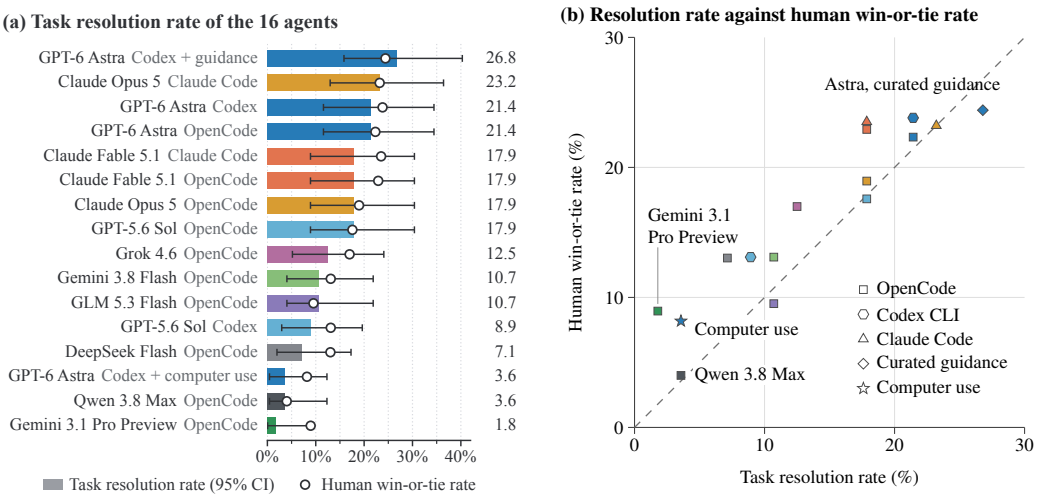

Figure 1: **Even the best agent resolves about a quarter of the tasks. (a)** Task resolution rate of the 16 agents on TIMELINE-BENCH, one run per task, with exact 95% confidence intervals; open circles give the human win-or-tie rate (Section 3.2). **(b)** The same two rates per agent, with quality margins held out by agent (Section 5). Colors follow (a), marker shapes give the harness or condition, and the dashed line marks equal rates.

advertising, and digital media, including constructing a narrative from interviews, assembling scenes from multiple takes, and producing promotional videos from mixed audiovisual assets. Each task in the benchmark is defined by a collection of source material, a project-specific brief, and a reproducible execution environment, with a reference video reserved for evaluation. Agents must interpret the brief, inspect the available assets, select and arrange footage, coordinate picture and sound, and render the final deliverable. We score each output with tests of delivery, content and brief compliance, including a quality test calibrated on blinded human preference, and report the human judgments separately. This protocol accommodates multiple valid editorial solutions while assessing both explicit requirements and viewing experience. Together, these assignments test audiovisual understanding, temporal reasoning, sustained tool use, and the interdependent creative decisions required to turn raw material into a coherent finished edit.

The remainder of this paper is structured as follows. We first describe TIMELINE-BENCH's construction and evaluation protocol. We then benchmark frontier LLMs and agents on 56 editing assignments. The best agent resolves 15 of the 56 tasks (Figure 1a). Finally, we analyze failure modes to inform future LLM and agent development, and compare harnesses, editorial guidance and computer use in Appendix I.

2 TIMELINE-BENCH

A TIMELINE-BENCH task consists of an edit brief, source assets, a Docker image, a set of tests, and a time limit (Figure 2). The brief describes the video the agent must produce, while the Docker image provides the tools and dependencies needed to complete the assignment. The tests assess whether the rendered video meets the technical specifications and satisfies the brief's content requirements. A task is successfully completed when all required tests pass and the benchmark performance is measured by the percentage of assigned tasks that are successfully completed. Evaluation focuses on the final video, allowing agents to choose their own workflow and produce different valid edits. Given the brief and source assets, an agent must inspect the material, select and arrange clips, coordinate the picture and sound, and export the requested video within the allotted time.



Figure 2: **An example TIMELINE-BENCH task, Two Player (EditStock).** Left: an excerpt of the brief the agent reads (`instruction.md`) and its production workspace. Middle: frames from the 185-second edit that GPT-6 Astra produced in OpenCode. Right: the tests of this run.

2.1 DATASET CONSTRUCTION

We construct TIMELINE-BENCH from 56 editing assignments drawn from three sources: 11 purchased project packages from EditStock[1], 15 publicly available editing projects from Cinestudy[2], and 30 newly commissioned projects from a professional video-editing agency. The agency supplied raw footage and finished reference edits. For Cinestudy, a professional editor selected reference edits from submissions linked publicly in the project-page comments, considering editing quality and availability. We prepare each assignment by organizing the available footage, audio, graphics, and supporting documents, and writing a brief that specifies the editorial objectives and delivery requirements.

2.2 VERIFICATION

We consider a task verified when its assignment is internally consistent, each of its tests has been validated against outputs with a known reference, and the agent has no access to the reference edit.

Assignment review. A task is well-specified if its brief describes an edit that the source assets can support and that the reference edit exemplifies. To confirm this, each brief was revised over several review rounds, after which a professional editor checked all 56 assignments for consistency between the brief, the source assets, and the reference edit (Appendix B).

Test validation. A test is valid if it accepts outputs that have the property it checks and rejects outputs that do not. Each task therefore ships a mechanical oracle, a deliberately low-craft render of the source assets that meets the delivery specification, on which the delivery tests (Section 3.1) are validated. The content and brief tests are validated on the reference edit, which passes all the content tests and all the brief tests. To confirm that these tests also reject defective outputs, we construct single-defect controls from real edits, for example, by muting the sound, blacking out the opening, reordering sections, or freezing the picture.

Reference isolation. An agent should not be able to pass a task by recovering its reference edit. Reference edits are stored separately from the task inputs and are never staged in the container. Agents may use the internet to consult tool documentation, but the briefs prohibit retrieving finished edits, and the traces of all agent runs contain no web fetch and no access to a reference edit (Appendix B).

[1] https://editstock.com
[2] https://cinestudy.org

2.3 COMPOSITION

TIMELINE-BENCH tasks vary widely. For example, **Klug Brand Story** asks for a sixty-second, interview-driven brand-story commercial from about four hours of dailies. **Beauty of Delhi** asks for a heritage travel commercial from 603 photographs and 14.9 seconds of camera footage, built largely from still-image sequences, photo holds and controlled crop movement. TIMELINE-BENCH comprises approximately 33.0 hours of primary source footage; source duration ranges from 2.3 to 240.4 minutes (median 12.4), and the 41 landscape and 15 portrait deliverables have duration windows from 25 to 315 seconds (per-collection statistics in Appendix A).

The assignments span narrative scenes, documentaries, interviews, action sequences, trailers, product advertisements, personal-branding videos, and lifestyle and travel films. A typical task requires seven of the 16 editing skills in Figure 3 (range 1–10), and 54 of the 56 tasks need at least one skill from each family. Following a script (53 tasks), music (46), dialog or voice-over editing (45) and titles, captions, or other on-screen text (45) are required almost everywhere, whereas take selection (30), sound effects (20), dual-system sync (7) and VFX compositing (7) appear in fewer tasks. The collections differ: every UGC task requires captions, color matching and reframing for portrait delivery, which almost no other task does, whereas Cinestudy and EditStock tasks hold most of the ambience, dual-system sync and VFX work.

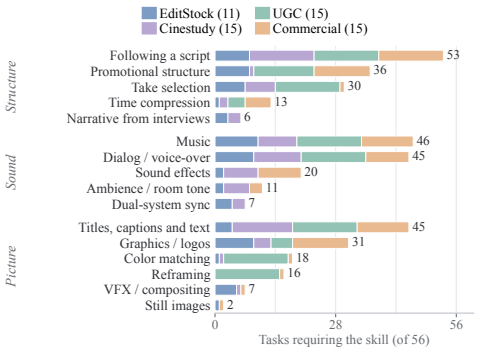

Figure 3: **Editing skills required across tasks.** Each bar counts the tasks whose brief, paperwork or supplied material requires the skill.

3 EVALUATION

Agent benchmarks often decide success with tests alone, because their tasks specify an acceptable end state (Jimenez et al., 2024; Merrill et al., 2026). In video editing, many different edits satisfy the same brief, so tests of its requirements are necessary but cannot establish quality. We therefore evaluate each output in three ways. First, tests check that the edit is a valid delivery and meets the brief's explicit requirements (Section 3.1). Second, a blind study of human preference has video editors compare each output with its task's reference edit (Section 3.2). Third, a quality test has multimodal LLM judges make the same comparison, calibrated on the human judgments (Section 3.3); an output that passes it and all other tests resolves its task.

3.1 TASK RESOLUTION RATE

An output *resolves* its task only if it passes every test. The tests inspect only the delivered video and are of four kinds. Four *delivery tests* per task check that the file exists and that its video stream, duration and audio stream match the brief. Six *content tests*, shared by all tasks, reject degenerate edits such as mostly silent, frozen or looped ones (Appendix C). *Brief tests*, one to six per task and 180 in total, check what the brief explicitly requires, such as using a supplied voice-over or keeping a scene order. Of these, 153 are programmatic: code matches the edit's frames and soundtrack against the task's source clips and recordings to find which ones it uses and where, reads its on-screen text with OCR (PP-OCR via RapidOCR; Du et al., 2020) and transcribes its speech (Whisper `small` via faster-whisper; Radford et al., 2023). Each such test then checks one of these facts, for example that a required line is spoken, that the supplied voice-over is heard or that clips appear in the required order. The other 27 brief tests concern visible content and are decided by a two-of-three majority of model judges (Gemini 3.8 Flash, GPT-6 Astra and Claude Opus 5.5; Google, 2026b; OpenAI, 2026a; Anthropic, 2026d; Appendix E). Finally, the *quality test* compares the edit with its task's reference edit (Section 3.3). Each agent runs once on each of the 56 tasks, and its *task resolution rate* is the share of tasks it resolves. A missing output resolves nothing.

3.2 Human Preference Study

In a blind pairwise study, 43 professional video editors compared each delivered output with its task's reference edit. They were paid hourly, independently of their answers, and report an average of at least two years of professional experience (Appendix F). The two edits appear as "Version A" and "Version B" in random order (Figure 7), next to a one-line statement of the project's audience and purpose. Human editors do not see the brief, the task title, the agent or other human editors' answers. They answer "Considering the complete viewing experience, which version would you choose for this project?" with *A preferred*, *B preferred* or *no meaningful preference*, or mark the pair *cannot assess* with a reason; we exclude those. Answers unlock only after both videos have played to the end at normal speed with sound, and the seek bar stays hidden until each video's first full viewing. Three distinct human editors judge each of the 863 pairs, giving 2,589 judgments, of which 2,582 are marked assessable. An agent's *win-or-tie rate*, as in GDPval (Patwardhan et al., 2026), is the share of judgments in which a human editor prefers its edit or has no meaningful preference, averaged within each task and then over tasks.

3.3 Quality Test

The quality test asks whether an edit is at least as good as its task's reference edit. Three multimodal LLM judges, Gemini 3.8 Flash, GPT-6 Astra and Claude Opus 5.5, score each edit separately, without seeing the other version. Like the human editors, they see the one-line statement of audience and purpose but not the brief. Additionally, they receive the study's written rubric of what makes an edit acceptable (Appendix D.1) and rate the edit from 1 to 10 overall and on story and assembly, pacing, picture, sound and graphics, which we combine into one score (Appendix D.2). Gemini 3.8 Flash watches the edit with sound. GPT-6 Astra and Claude Opus 5.5 accept only text and images, so they read contact sheets of one frame per second instead, together with four measurements computed from the file: cuts per minute from a shot-boundary detector (our re-implementation of PySceneDetect's adaptive detector; Castellano, 2026), the longest stretch of static picture, integrated loudness (EBU R128, measured with FFmpeg) and the share of silent runtime. Because judges use the 1-to-10 scale differently, we convert each judge's scores to z-scores, using that judge's mean and standard deviation over the 919 videos in the study (863 agent edits and 56 reference edits), and keep these constants fixed for new submissions. The *panel score* of an edit is the average of its three z-scores.

The test passes when the agent edit's panel score exceeds the reference edit's by at least the *tie margin* of the task's collection. A margin is fitted on a collection's study pairs: it is set so that the test's pass rate comes as close as possible to the human win-or-tie rate without exceeding it. Because the panel score averages z-scores, margins are measured in judge standard deviations; one judge standard deviation is 1.0 to 1.5 points on the 1-to-10 scale. We fit margins in two ways. For new submissions, which have no human judgments, the margins are fitted once on all 2,582 assessable judgments and then frozen: 0.30 (Cinestudy), 0.52 (Commercial), 0.21 (EditStock) and 1.54 (UGC) (Appendix D). For the results we report, fitting on an agent's own judgments would be circular, so each agent is scored with margins refitted per collection without its own judgments (leave-one-agent-out). These held-out margins stay within 0.09 of the frozen values. All margins are positive, so an agent edit must score higher than its reference edit to pass.

4 Experimental Setup

We evaluate 16 agents. An *agent* is a model running in a *harness*, the program that gives the model its tools and executes its commands. Each agent attempts each task once, giving 896 *runs*.

4.1 Agents

An agent's result depends on both its model and its harness, and model developers tune their own harnesses for their own models. We therefore run ten models in one open-source harness, OpenCode 1.18.31 (OpenCode, 2026): GPT-6 Astra (OpenAI, 2026a), GPT-5.6 Sol (OpenAI, 2026b), Claude Fable 5.1 (Anthropic, 2026b), Claude Opus 5 (Anthropic, 2026c), Gemini 3.1 Pro Preview (Google, 2026a), Gemini 3.8 Flash (Google, 2026b), DeepSeek Flash (DeepSeek, 2026), Grok 4.6 (xAI, 2026), GLM 5.3 Flash (Z.ai, 2026) and Qwen 3.8 Max (Qwen Team, 2026). Four of these models also run

Table 1: **Main results**, one run per task. *Tests*: runs, of 56, passing every test but the quality test. *Res.*: resolution rate. *Human*: human win-or-tie rate. With CIs in Table 8.

Model	Harness	Tests	Res. %	Human %
GPT-6 Astra	OpenCode	48	21.4	22.3
Claude Fable 5.1	OpenCode	50	17.9	22.9
Claude Opus 5	OpenCode	46	17.9	19.0
GPT-5.6 Sol	OpenCode	45	17.9	17.6
Grok 4.6	OpenCode	39	12.5	17.0
Gemini 3.8 Flash	OpenCode	42	10.7	13.1
GLM 5.3 Flash	OpenCode	41	10.7	9.5
DeepSeek Flash	OpenCode	43	7.1	13.0
Qwen 3.8 Max	OpenCode	35	3.6	4.0
Gemini 3.1 Pro Preview	OpenCode	19	1.8	9.0
Claude Opus 5	Claude Code	50	23.2	23.2
GPT-6 Astra	Codex CLI	48	21.4	23.8
Claude Fable 5.1	Claude Code	49	17.9	23.5
GPT-5.6 Sol	Codex CLI	45	8.9	13.1
GPT-6 Astra	Codex CLI, guidance	52	26.8	24.4
GPT-6 Astra	Codex CLI, comp. use	35	3.6	8.2
All agents		687	14.0	16.5

in their developers' own harnesses: GPT-5.6 Sol and GPT-6 Astra in Codex CLI 0.155.1 (OpenAI, 2025), and Claude Fable 5.1 and Claude Opus 5 in Claude Code 2.1.278 (Anthropic, 2026a). We use unmodified releases of all three harnesses, and every model runs at the highest reasoning effort its provider offers (Appendix H).

Curated guidance. One agent adds editorial guidance to GPT-6 Astra in Codex CLI. It receives 1,909 words of general editing advice: a skill file (`SKILL.md`) and six reference notes on reading the brief and footage, planning and building the edit, reviewing and delivering it, and using the installed video tools. The advice contains no task-specific answers. A one-line note in the agent's instruction file (`AGENTS.md`) says where to find it; everything else matches the unguided agent.

Computer use. Another agent runs GPT-6 Astra in Codex CLI's computer-use mode, in which it sees the screen and controls the mouse and keyboard but has no shell, so it cannot use command-line tools. It edits in the free edition of DaVinci Resolve (Blackmagic Design, 2026) on macOS. Its briefs are the same except for the tool instructions, which tell it to work in Resolve, and a Resolve project is already open with the task's media imported.

4.2 Environment and Budgets

The coding agents run in Linux containers with the tools an editor working in code needs: FFmpeg, Python and Node, the programmatic video frameworks Remotion and HyperFrames with a headless browser, Poppler for PDF paperwork, and a transcription command (AssemblyAI Universal-3.5 Pro; versions in Appendix H). Every run starts from a fresh harness state and the same input files, which are checked against their SHA-256 hashes before the run. Each run has a 300-minute limit and a whole machine to itself (32 vCPUs and 256 GB of memory), with no CPU or memory limits on the container. Agents may use the internet, but the briefs forbid retrieving finished edits, and we audit the traces for such retrievals (Section 2.2). Tasks are packaged in the Harbor format (Harbor Framework Team, 2026), which fixes each task's brief, inputs and delivery tests (Appendix H).

5 Results

Figure 1a shows the task resolution rate of every agent, and Table 1 gives the numbers (with confidence intervals in Table 8). GPT-6 Astra in Codex CLI with curated editorial guidance resolves the most tasks, 15 of 56 (26.8%), followed by Claude Opus 5 in Claude Code at 23.2% and by GPT-6 Astra in Codex CLI and in OpenCode at 21.4% each. With one run per task, each agent's rate has a 95% CI of about ± 11 points, so the agents near the top cannot be told apart. Overall, 125 of the 896 runs resolve their task (14.0%, 95% CI 11.7–16.4%), and Section 6 examines where the others fail. The median runtime per run ranges from 16 to 61 minutes and is unrelated to resolution (Spearman $\rho = 0.15$). Among the 12 agents with recorded costs, from \$0.13 to \$41.27 per run, more expensive agents tend to resolve more tasks ($\rho = 0.79$), but the best agent, GPT-6 Astra with curated guidance, costs an estimated \$9.76 per run (Codex CLI records only tokens; Appendix H.3).

Human preference. Human editors prefer the reference edit in 83.5% of the 2,582 assessable judgments, the agent edit in 11.1%, and have no meaningful preference in 5.5%. The win-or-tie rate

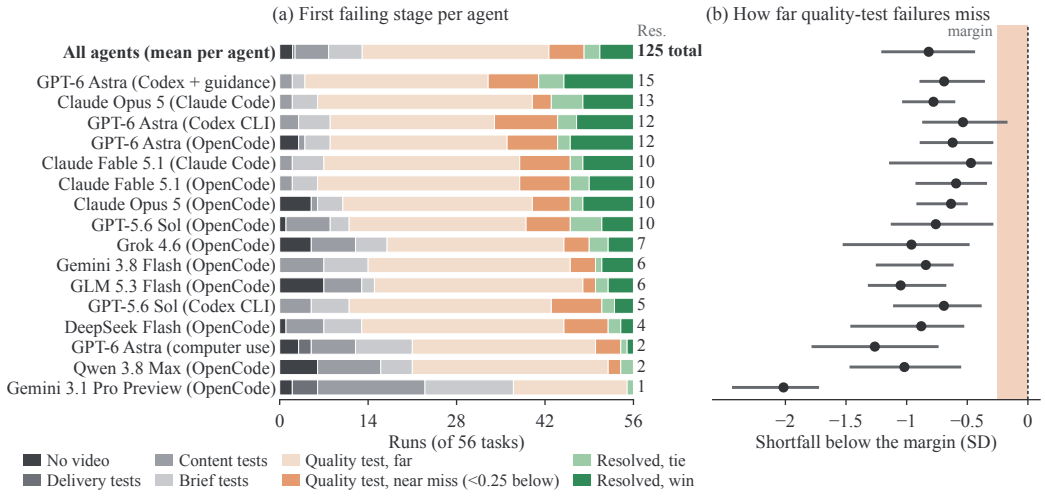

Figure 4: **Most unresolved runs pass every test and fail only the quality test.** (a) The 56 runs of each agent by the first test they fail (Section 3.1); quality-test failures are split into near misses (less than 0.25 panel standard deviations below the margin) and far misses, resolved runs into wins and ties. (b) How far each agent's quality-test failures fall below the margin, in judge standard deviations: median (dot) and interquartile range (bar); the shaded band marks near misses.

averages 16.5% over agents. It ranges from 24.4% for GPT-6 Astra with curated guidance, 8.3 points of which come from ties, to 4.0% for Qwen 3.8 Max.

Single judgments are noisy (Krippendorff's $\alpha = 0.10$), but per-agent rates are reliable (Spearman–Brown reliability 0.84 with three human editors per pair), robust to leaving out any human editor (Spearman $\rho \geq 0.96$) and unaffected by the order of the two versions (Appendices F and G.1). The resolution rate tracks the human win-or-tie rate across agents (Figure 1b). With margins refit without each agent's own judgments, the two rates differ by 3.0 points on average and by 7.2 at most (Gemini 3.1 Pro Preview), with Spearman $\rho = 0.93$ and Pearson $r = 0.93$ over the 16 agents.

6 Analysis

In this section, we ask where runs fail, why human editors prefer the reference edits and how agents work (full analysis in Appendix J, methods in Appendix K).

The quality test is where agents lose. Of the 771 unresolved runs, 562 (73%) pass the delivery, content and brief tests and fail only the quality test (Figure 4a); for 15 of the 16 agents this accounts for 61–90% of their losses, the exception being Gemini 3.1 Pro Preview, which loses 35 of its 55 unresolved runs to earlier tests. Most of these failures are narrow (Figure 4b): the median one falls 0.82 judge standard deviations short of the margin, about one point on the 1-to-10 scale, and 0.47 to 0.78 for the eight agents that resolve at least 10 tasks, against 2.02 for Gemini 3.1 Pro Preview. Because so many runs sit near the bar, the number resolved depends on where the margin is set, but the order of agents largely does not: lowering every margin by 0.1 standard deviations would resolve 27 more runs, and although agents a run or two apart can swap places, including the top two, the ranking barely moves (Spearman 0.96).

Human editors prefer the reference edit for craft that the agent edit lacks. We coded all 1,271 human editors' notes with Claude Opus 5.5 into 24 categories (second-coder $\kappa = 0.84$, author-audited; Appendix K.1). Of the 1,101 notes explaining a preference for the reference edit, 95% cite a strength of the reference edit but only 44% name any problem in the agent edit (Figure 5a). The strengths are finishing and editorial judgment, such as polish, shot selection, graphics, and transitions; only audio defects and repeated footage are main agent errors. Measurements agree: paired by task, agents change shots only 0.72 times as often as the reference edit and hold their longest static shot 1.5 times as long (Figure 5b), and human editors penalize the slower pace but not a faster one. Agents

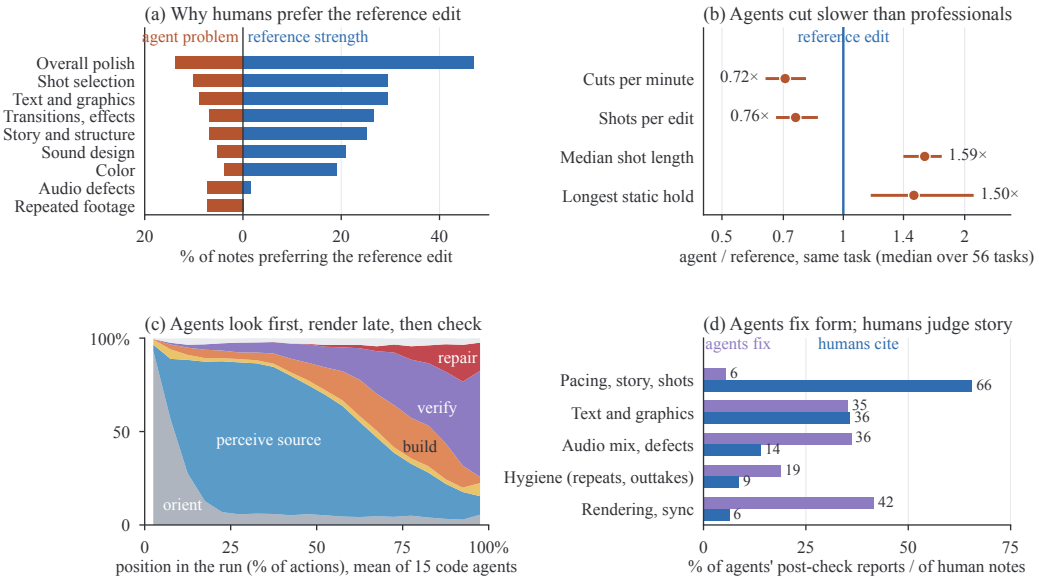

Figure 5: **Agents produce clean assemblies but not finished edits, and check for defects rather than quality.** (a) Share of the human notes preferring the reference edit that name a problem of the agent edit or a strength of the reference edit. (b) Agent edit relative to the reference edit of the same task (median over tasks of the mean ratio; 95% task-bootstrap interval). (c) Phases of the native actions of the 15 code agents over the course of a run. (d) What agents report fixing after checking their own render (343 reports) and what human editors cite in notes preferring the reference edit.

win on story, not on graphics ("[agent] gets into the heist story more directly"). Graphics decides 91% of tagged UGC judgments, story 84–86% in the narrative collections. Stronger agents fail more subtly. When humans reject an agent edit, their notes name a defect that a content test could catch, such as an audio glitch or repeated footage, for 10% of the six highest-rated agents but for 19% of the three weakest code agents (Gemini 3.1 Pro Preview, Qwen 3.8 Max and GLM 5.3 Flash). The computer-use agent is worst, at 28%: it leaves slates, crew talk or retakes in 18.5% of its rejected edits ("In [agent], you can hear the command 'Action!' That's not okay.").

Difficulty belongs to tasks. Agents tend to fail the same tasks: 35 tasks are resolved by no agent, whereas only about 5 would be if each agent's successes fell on random tasks. Tasks also differ more than agents do. If we model each human vote as depending on the agent's skill and the task's difficulty (a Rasch model), task difficulty varies 2.6 times as much as agent skill, and even the best agent is more likely to lose than to win or tie on 54 of the 56 tasks (Appendix J.3). The collection explains half of this: the human win-or-tie rate falls from 28.7% on Cinestudy to 6.8% on UGC. Within a collection, however, no task property or required editing skill predicts difficulty, and no agent is especially strong at particular skills.

Agents look first, render late and check for defects. Code agents spend 54% of their 111,636 native actions perceiving the source and 17% verifying their own render, but only 9% building; 98% of their perception precedes the first render, which comes 72–90% of the way through a run (Figure 5c). They see footage as stills (83% of native reads open frames or contact sheets) and hear it as transcripts and level measurements; GPT agents in Codex CLI even print audio excerpts as base64 text, in up to 55 of 56 runs. Checking is nearly universal (86–100% of runs for 14 of 15 code agents) and tracks resolution across agents ($\rho = 0.73$), but it targets form: only 5.5% of the problems agents report after a check concern pacing, story or shot choice, against 66% of human editors' notes (Figure 5d). Agents' final messages claim full success in 93% of runs, including 95.5% of edits that fail a test.

Table 2 summarizes that agents reliably do what a brief can state (delivering the specified video, meeting explicit requirements, keeping the timeline clean), but they cannot judge their own edits and remain weak at what a human editor adds: rhythm, shot selection and finishing.

Table 2: **Current agents as editors.** Evidence from Section 6 and Appendix J.

Ability	Evidence	Reading
Delivery and explicit requirements	857 of 863 videos meet the specification; 762 pass every brief test	Reliable
Sound	Required lines missed: 5.9% (transcripts); 41.5% for computer use	Via tools only
Timeline hygiene	Repeats, outtakes in 1–7% of top agents' losses; 18.5% for computer use	Mostly reliable
Checking one's own edit	Output checked in 86–100% of runs; 5.5% of fixes concern pacing, story, shots	Defects only
Judging one's own edit	93% of runs claim success, including 95.5% of edits that fail a test	Absent
Story assembly	Drives agent wins; narrative scenes are the best collection (28.7%)	Emerging
Rhythm and shot selection	0.72× the reference cut rate; shot selection cited in 29% of losses	Weak
Finishing	Graphics, transitions, sound, color: reference strengths in 19–47% of notes	Weak

7 Limitations

Each agent runs once per task, so our intervals omit run-to-run variance. The quality test was calibrated on the same human editors' votes that it is validated against, and it is valid per agent, not per edit (Appendix D). The reference edits are professional edits, not certified ground truth.

8 Related Work

Agent benchmarks. Outcome-graded agent benchmarks cover software engineering, terminal work and machine-learning research (Jimenez et al., 2024; Deng et al., 2026b; Merrill et al., 2026; Chan et al., 2025; Starace et al., 2025), and GDPval and Agents' Last Exam extend evaluation to professional deliverables judged by experts (Patwardhan et al., 2026; Sun et al., 2026); checklists for agent benchmarks call for frozen inputs and validated evaluators (Zhu et al., 2025; Kapoor et al., 2025). Timeline-Bench keeps outcome-graded tasks, but because an edit has no single correct answer, its quality test is calibrated on blind professional judgments.

Video editing and media agents. Video benchmarks measure components of editing, such as recognizing techniques or choosing cuts from one long video (Deng et al., 2026a; Ogata et al., 2026), or evaluate post-production operations and GUI trajectories in media software and analyze their technical failures (Cao et al., 2026; Hu et al., 2026; Heo et al., 2026; Ai et al., 2026); dedicated systems build editing agents (Sandoval-Castañeda et al., 2025; Zhou et al., 2026). Video-understanding benchmarks ask multiple-choice questions about long or audio-visual video (Fu et al., 2025; Wu et al., 2024; Hong et al., 2026), whereas the agents we evaluate perceive footage through still frames and transcripts. Timeline-Bench evaluates complete assignments from raw material to a delivered edit, compares coding and GUI agents, and characterizes failures of craft from editors' notes.

Judging subjective outputs. Human preference ranks systems (Chiang et al., 2024; Patwardhan et al., 2026), and model judges approximate it (Zheng et al., 2023) but are biased by position (Wang et al., 2024; Shi et al., 2025), also for video (Ogata et al., 2026). We therefore score each edit with a cross-laboratory panel calibrated on human judgments and validated per agent (Appendix N).

9 Conclusion

We introduce Timeline-Bench, a benchmark of 56 complete video-editing assignments in which agents turn raw production material and a brief into finished edits. Across 16 agents, the strongest resolves 15 tasks, while the benchmark's quality test closely tracks human preference ($\rho = 0.93$), revealing that the main gap is craft rather than compliance: 562 of 771 unresolved runs pass every other requirement but fail quality. Agents cut at just 0.72× the reference pace, rely heavily on still frames and transcripts, inspect renders for technical defects rather than pacing or story, and claim success in 93% of runs. The results suggest that creative benchmarks need human-calibrated quality evaluation, while capable agents need native audio-visual perception, finishing tools, and the ability to judge creative quality, not merely correctness. We release the tasks, verifier, and per-run results to enable rigorous measurement of progress.

REPRODUCIBILITY STATEMENT

Sections 2–4 define the tasks, tests, quality test, human study and agents. Appendix H gives the exact model identifiers, harness versions, reasoning settings and launch commands, Appendix B the task-verification procedure, Appendix G the statistical methods, and Appendix M the release. The anonymous site (https://timelinebench.vercel.app) and its anonymized code mirror (https://timelinebench.vercel.app/code) provide the 56 Harbor tasks, whose briefs, tests and oracle solutions are byte-identical to those the coding agents were evaluated against; the verifier with its frozen constants and the reference edits' panel scores, so that new edits can be scored without the reference edits; every per-run outcome; and a script that recomputes the automatic-evaluation results and the count-based human-study statistics without model calls, network access, media or Docker. Statistics that need per-judgment editor records, such as the two-way editor intervals, cannot be recomputed from the release, because those records are withheld to protect participants. Judges called again may answer slightly differently, since provider serving is not guaranteed to be identical over time. Media access is gated (Appendix M) and is not needed to review this paper.

ETHICS STATEMENT

The human study asked 43 freelance video editors for blind preference judgments between two complete edits. Human editors were recruited individually, and their pay did not depend on their answers. Before the first comparison, each human editor read an on-screen description of the task and of the interaction data recorded (playback controls, viewing coverage, answer changes, clicks, tab visibility and time spent; no camera, microphone, screen or keystrokes), and gave consent through a required acknowledgment. We did not seek institutional ethics review. The study collected expert preference judgments about non-sensitive video material, and apart from the contact details used to administer the study and the interaction data listed above, the only personal information collected was self-reported editing experience. We release only aggregate statistics and per-run vote counts; no editor identities, per-person telemetry or per-judgment records are published, and human editors' free-text notes appear only as short anonymous excerpts in this paper.

The EditStock project packages were purchased under license, the commissioned productions were made for the benchmark with rights to the material, and the Cinestudy projects are publicly available editing exercises. None of the source footage, reference edits or agent outputs is redistributed publicly. Identifiable people in the commissioned material are covered by releases, and the access terms forbid identifying, profiling or imitating the people shown (Appendix M). Our results describe the sampled agents and tasks, not the abilities or employability of video editors.

AI USE STATEMENT

In this work, AI models are part of the method, and we used generative AI tools for implementation, analysis and writing. Within the method, the 16 evaluated agents are AI systems; a panel of three AI judges scores edits for the quality test and decides the judge-based brief checks (Section 3.3, Appendix E); a speech-recognition model transcribes edits for code-based brief checks; the brief-test rubrics were drafted by one AI agent, adversarially reviewed by a second and verified by the authors, and are further validated by the oracle rule, single-defect controls and cross-laboratory judge agreement; two model annotators (Claude Opus 5.5 and Claude Sonnet 5) and a third adjudicating model pass tagged the editing skills of each task, and the authors cross-checked the tags (Appendix A.1); Claude Opus 5.5 coded the human editors' notes, with Claude Sonnet 5 as second coder, and labeled the agents' final messages, and the authors checked these codes and labels (Appendix K); and GPT-6 Astra labeled shell-command failures, and the authors checked those labels (Appendix L). We also used generative AI tools to write and test code for the analyses and figures, to search and synthesize the literature, and to draft and edit this manuscript. Formulating mathematical claims and writing proofs are not applicable to this work. We have reviewed all AI-assisted work: every citation was checked against its arXiv, publisher or vendor record, two independent recomputations reproduce every deterministic number in the result files with no discrepancies, and the authors read and revised all AI-assisted text, code and analyses. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Jiaxin Ai, Yukang Feng, Fanrui Zhang, Jianwen Sun, Zizhen Li, Chuanhao Li, Yifan Chang, Wenxiao Wu, Ruoxi Wang, Mingliang Zhai, and Kaipeng Zhang. ProSoftArena: Benchmarking hierarchical capabilities of multi-modal agents in professional software environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 34586–34595, June 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Ai_ProSoftArena_Benchmarking_Hierarchical_Capabilities_of_Multi-modal_Agents_in_Professional_Software_CVPR_2026_paper.html.

Anthropic. How Claude Code works, 2026a. URL https://code.claude.com/docs/en/how-claude-code-works. Documentation, accessed September 22, 2026.

Anthropic. Introducing Claude Fable 5.1 and Claude Mythos 5.1, sep 2026b. URL https://www.anthropic.com/claude-fable-and-mythos-5-1. Anthropic model announcement, accessed September 25, 2026.

Anthropic. Introducing Claude Opus 5, July 2026c. URL https://www.anthropic.com/news/claude-opus-5. Published July 24, 2026.

Anthropic. Introducing Claude Opus 5.5, September 2026d. URL https://www.anthropic.com/claude-opus-5-5. Published September 22, 2026.

Blackmagic Design. DaVinci Resolve 21, 2026. URL https://www.blackmagicdesign.com/products/davinciresolve. Non-linear editing, color, effects and audio post-production software. Accessed September 25, 2026.

William Brown. Some experimental results in the correlation of mental abilities. *British Journal of Psychology*, 3(3):296–322, 1910. doi: 10.1111/j.2044-8295.1910.tb00207.x.

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. URL https://arxiv.org/abs/2605.27705.

Brandon Castellano. PySceneDetect: Video cut detection and analysis tool. https://github.com/Breakthrough/PySceneDetect, 2026. Version 0.7.1.

Jun Shern Chan, Neil Chowdhury, Oliver Jaffe, James Aung, Dane Sherburn, Evan Mays, Giulio Starace, Kevin Liu, Leon Maksin, Tejal Patwardhan, Aleksander Madry, and Lilian Weng. MLE-bench: Evaluating machine learning agents on machine learning engineering. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/7e3767db483c942b883eb4f8cfb74e31-Abstract-Conference.html.

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Banghua Zhu, Hao Zhang, Michael Jordan, Joseph E. Gonzalez, and Ion Stoica. Chatbot Arena: An open platform for evaluating LLMs by human preference. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 8359–8388. PMLR, 2024. URL https://proceedings.mlr.press/v235/chiang24b.html.

C. J. Clopper and E. S. Pearson. The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika*, 26(4):404–413, 1934. doi: 10.1093/biomet/26.4.404.

DeepSeek. DeepSeek-V4.1-Flash: Smarter, faster, more efficient, September 2026. URL https://api-docs.deepseek.com/news/news260910. API news; the `deepseek-flash` alias is served by DeepSeek-V4.1-Flash. Accessed September 25, 2026.

Andong Deng, Dawei Du, Zhenfang Chen, Wen Zhong, Fan Chen, Guang Chen, Chia-Wen Kuo, Longyin Wen, Chen Chen, and Sijie Zhu. VEBench: Benchmarking large multimodal models for real-world video editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Findings*, pp. 2187–2196, June 2026a. URL https://openaccess.thecvf.com/content/CVPR2026F/html/Deng_VEBench_Benchmarking_Large_Multimodal_Models_for_Real-world_Video_Editing_CVPRF_2026_paper.html.

Xiang Deng, Jeff Da, Edwin Pan, Yannis Yiming He, Charles Ide, Kanak Garg, Niklas Lauffer, Andrew Park, Chetan Rane, Karmini Sampath, Maya Krishnan, Srivatsa Kundurthy, Sean Hendryx, Zifan Wang, Chen Bo Calvin Zhang, Noah Jacobson, Bing Liu, and Brad Kenstler. SWE-Bench Pro: Can AI agents solve long-horizon software engineering tasks? In *International Conference on Machine Learning*, 2026b. URL https://icml.cc/virtual/2026/poster/61047.

Yuning Du, Chenxia Li, Ruoyu Guo, Xiaoting Yin, Weiwei Liu, Jun Zhou, Yifan Bai, Zilin Yu, Yehua Yang, Qingqing Dang, and Haoshuang Wang. PP-OCR: A practical ultra lightweight OCR system. *arXiv preprint arXiv:2009.09941*, 2020.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, Peixian Chen, Yanwei Li, Shaohui Lin, Sirui Zhao, Ke Li, Tong Xu, Xiawu Zheng, Enhong Chen, Caifeng Shan, Ran He, and Xing Sun. Video-MME: The first-ever comprehensive evaluation benchmark of multi-modal LLMs in video analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24108–24118, June 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Fu_Video-MME_The_First-Ever_Comprehensive_Evaluation_Benchmark_of_Multi-modal_LLMs_in_CVPR_2025_paper.html.

Google. Gemini 3.1 Pro: A smarter model for your most complex tasks, February 2026a. URL https://blog.google/innovation-and-ai/models-and-research/gemini-models/gemini-3-1-pro/. Published February 19, 2026.

Google. Introducing Gemini 3.8 Flash and 3.8 Flash Cyber, September 2026b. URL https://blog.google/innovation-and-ai/models-and-research/gemini-models/3-8-flash-and-3-8-flash-cyber/. Published September 2, 2026.

Harbor Framework Team. Harbor: A framework for evaluating and optimizing agents and models in container environments. Software, concept DOI 10.5281/zenodo.20953922, 2026. URL https://github.com/harbor-framework/harbor.

Andrew F. Hayes and Klaus Krippendorff. Answering the call for a standard reliability measure for coding data. *Communication Methods and Measures*, 1(1):77–89, 2007. doi: 10.1080/19312450709336664.

Chiyeong Heo, Jaechang Kim, Junhyuk Kwon, Hoyoung Kim, Dongmin Park, Jonghyun Lee, and Jungseul Ok. MMTB: Evaluating terminal agents on multimedia-file tasks. arXiv preprint arXiv:2605.10966, 2026. URL https://arxiv.org/abs/2605.10966.

Jack Hong, Shilin Yan, Jiayin Cai, Xiaolong Jiang, Yao Hu, and Weidi Xie. WorldSense: Evaluating real-world omnimodal understanding for multimodal LLMs. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2502.04326. arXiv:2502.04326; ICLR 2026 acceptance per the arXiv record.

Haobo Hu, Xiangwu Guo, Zhiheng Chen, Difei Gao, Haotian Liu, Libiao Jin, and Qi Mao. CutVerse: A compositional GUI agents benchmark for media post-production editing. arXiv preprint arXiv:2605.19484, 2026. URL https://arxiv.org/abs/2605.19484.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. SWE-bench: Can language models resolve real-world GitHub issues? In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=VTF8yNQM66.

Sayash Kapoor, Benedikt Stroebl, Zachary S. Siegel, Nitya Nadgir, and Arvind Narayanan. AI agents that matter. *Transactions on Machine Learning Research*, 2025. URL https://openreview.net/forum?id=Zy4uFzMviZ.

Klaus Krippendorff. *Content Analysis: An Introduction to Its Methodology*. SAGE Publications, 4th edition, 2018. doi: 10.4135/9781071878781.

Mike Merrill, Alexander Shaw, Nicholas Carlini, Boxuan Li, Harsh Raj, Ivan Bercovich, Lin Shi, Jeong Shin, Thomas Walshe, E. Kelly Buchanan, Junhong Shen, Guanghao Ye, Haowei Lin, Jason Poulos, Maoyu Wang, Marianna Nezhurina, Di Lu, Orfeas Menis Mastromichalakis, Zhiwei Xu, Zizhao Chen, Yue Liu, Robert Zhang, Leon Liangyu Chen, Anurag Kashyap, Jan-Lucas Uslu, Jeffrey Li, Jianbo Wu, Minghao Yan, Song Bian, Vedang Sharma, Ke Sun, Steven Dillmann, Akshay Anand, Andrew Lanpouthakoun, Bardia Koopah, Changran Hu, Etash Guha, Gabriel Dreiman, Jiacheng Zhu, Karl Krauth, Li Zhong, Niklas Muennighoff, Robert Amanfu, Shangyin Tan, Shreyas Pimpalgaonkar, Tushar Aggarwal, Xiangning Lin, Xin Lan, Xuandong Zhao, Yiqing Liang, Yuanli Wang, Zilong (Ryan) Wang, Changzhi Zhou, David Heineman, Hange Liu, Harsh Trivedi, John Yang, Junhong Lin, Manish Shetty, Michael Yang, Nabil Omi, Negin Raoof, Shanda Li, Terry Yue Zhuo, Wuwei Lin, Yiwei Dai, Yuxin Wang, Wenhao Chai, Shang Zhou, Dariush Wahdany, Ziyu She, Jiaming Hu, Zhikang Dong, Yuxuan Zhu, Sasha Cui, Ahson Saiyed, Arinbjörn Kolbeinsson, Christopher Rytting, Ryan Marten, Yixin Wang, Jenia Jitsev, Alex Dimakis, Andy Konwinski, and Ludwig Schmidt. Terminal-Bench: Benchmarking agents on hard, realistic tasks in command line interfaces. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=a7Qa4CcHak. ICLR 2026 poster; arXiv:2601.11868.

Katsuya Ogata, Zongshang Pang, Mayu Otani, and Yuta Nakashima. MEDit-Bench: A dataset for evaluating message-driven narrative video editing. arXiv preprint arXiv:2607.25300, 2026. URL https://arxiv.org/abs/2607.25300.

OpenAI. Introducing Codex, May 2025. URL https://openai.com/index/introducing-codex/. Accessed September 22, 2026.

OpenAI. GPT-6 Astra: A new generation of intelligence, sep 2026a. URL https://openai.com/index/gpt-6-astra/. OpenAI product announcement, accessed September 25, 2026.

OpenAI. GPT-5.6: Frontier intelligence that scales with your ambition, 2026b. URL https://openai.com/index/gpt-5-6/. Launch post for GPT-5.6 Sol, Terra and Luna; accessed September 25, 2026.

OpenCode. OpenCode: The open source AI coding agent, 2026. URL https://opencode.ai/. Accessed September 25, 2026.

Art B. Owen. The pigeonhole bootstrap. *The Annals of Applied Statistics*, 1(2):386–411, 2007. doi: 10.1214/07-AOAS122.

Tejal Patwardhan, Rachel Dias, Elizabeth Proehl, Grace Kim, Michele Wang, Olivia Watkins, Simon Fishman, Marwan Aljubeh, Phoebe Thacker, Laurance Fauconnet, Natalie Kim, Samuel Miserendino, Gildas Chabot, David Li, Patrick Chao, Michael Sharman, Alexandra Barr, Amelia Glaese, and Jerry Tworek. GDPval: Evaluating AI model performance on real-world economically valuable tasks. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/290c2430f91912204f30bbcc990fff1d-Abstract-Conference.html.

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

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine Mcleavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 28492–28518. PMLR, 2023. URL https://proceedings.mlr.press/v202/radford23a.html.

Marcelo Sandoval-Castañeda, Bryan Russell, Josef Sivic, Gregory Shakhnarovich, and Fabian Caba Heilbron. EditDuet: A multi-agent system for video non-linear editing. In *SIGGRAPH Conference Papers '25*, pp. 2:1–2:11. ACM, 2025. doi: 10.1145/3721238.3730761.

Daniel Shepard and Robin Salimans. AutomationBench. arXiv preprint arXiv:2604.18934, 2026. URL https://arxiv.org/abs/2604.18934.

Lin Shi, Chiyu Ma, Wenhua Liang, Xingjian Diao, Weicheng Ma, and Soroush Vosoughi. Judging the judges: A systematic study of position bias in LLM-as-a-judge. In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pp. 292–314, 2025. URL https://aclanthology.org/2025.ijcnlp-long.18/.

Charles Spearman. Correlation calculated from faulty data. *British Journal of Psychology*, 3(3): 271–295, 1910. doi: 10.1111/j.2044-8295.1910.tb00206.x.

Giulio Starace, Oliver Jaffe, Dane Sherburn, James Aung, Jun Shern Chan, Leon Maksin, Rachel Dias, Evan Mays, Benjamin Kinsella, Wyatt Thompson, Johannes Heidecke, Amelia Glaese, and Tejal Patwardhan. PaperBench: Evaluating AI's ability to replicate AI research. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 56843–56873. PMLR, 2025. URL https://proceedings.mlr.press/v267/starace25a.html.

Yiyou Sun, Xinyang Han, Weichen Zhang, et al. Agents' last exam. arXiv preprint arXiv:2606.05405 (v2), 2026. URL https://arxiv.org/abs/2606.05405v2.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Lingpeng Kong, Qi Liu, Tianyu Liu, and Zhifang Sui. Large language models are not fair evaluators. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9440–9450. Association for Computational Linguistics, 2024. URL https://aclanthology.org/2024.acl-long.511/.

Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. LongVideoBench: A benchmark for long-context interleaved video-language understanding. In *Advances in Neural Information Processing Systems (Datasets and Benchmarks Track)*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/329ad516cf7a6ac306f29882e9c77558-Abstract-Datasets_and_Benchmarks_Track.html.

xAI. Introducing Grok 4.6, August 2026. URL https://x.ai/news/grok-4-6. Published August 12, 2026.

Z.ai. GLM-5.3-Flash: Frontier intelligence, flash cost, August 2026. URL https://z.ai/blog/glm-5.3-flash. Published August 26, 2026.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-bench and Chatbot Arena. In *Advances in Neural Information Processing Systems (Datasets and Benchmarks Track)*, volume 36, pp. 46595–46623, 2023. URL https://papers.nips.cc/paper_files/paper/2023/hash/91f18a1287b398d378ef22505bf41832-Abstract-Datasets_and_Benchmarks.html.

Hengji Zhou, Lingxuan Huang, Jian Wang, Bing Zhou, Si Wu, Lianghao Xia, and Chao Huang. VideoAgent: All-in-one framework for video understanding and editing. *arXiv preprint arXiv:2606.23327*, 2026. URL https://arxiv.org/abs/2606.23327.

Yuxuan Zhu, Tengjun Jin, Yada Pruksachatkun, Andy Zhang, Shu Liu, Sasha Cui, Sayash Kapoor, Shayne Longpre, Kevin Meng, Rebecca Weiss, Fazl Barez, Rahul Gupta, Jwala Dhamala, Jacob Merizian, Mario Giulianelli, Harry Coppock, Cozmin Ududec, Antony Kellermann, Jasjeet Sekhon, Jacob Steinhardt, Sarah Schwettmann, Arvind Narayanan, Matei A. Zaharia, Ion Stoica, Percy Liang, and Daniel Kang. Establishing best practices in building rigorous agentic benchmarks. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/f316275b44ee2de533102913828a8107-Abstract-Datasets_and_Benchmarks_Track.html.

APPENDIX CONTENTS

A Task Composition **16**
A.1 Editing Skills 16

B Task Verification and Quality Control **16**

C Content Tests **17**

D Quality Test **17**
D.1 Rubric 17
D.2 Scoring 18
D.3 Margin Calibration 18
D.4 Individual Edits 18

E Brief Compliance **18**

F Human Study **18**

G Statistical Methods **18**
G.1 Reliability of Per-Agent Rates 20

H Agent Configurations and Runtime **20**
H.1 Launch Commands 20
H.2 Execution Environment 21
H.3 Cost and Runtime 21

I Detailed Results **22**
I.1 Matched Contrasts 22

J Extended Analysis **23**
J.1 Where Runs Fail 23
J.2 Why Human Editors Prefer the Reference Edits 23
J.3 Task Difficulty 23
J.4 Trajectory-Level Analysis 24

K Analysis Methods **24**
K.1 Coding of Human Editors' Notes 24
K.2 Task Difficulty and Skill Profiles 24
K.3 Trajectory Phases and Process Features 25

L Command-Failure Analysis **26**

M Release and Licensing **26**

N Related Evaluation Settings **26**

A TASK COMPOSITION

Table 3 gives per-collection statistics of TIMELINE-BENCH.

Table 3: **Composition of TIMELINE-BENCH by collection.** Source hours sum each task's primary moving-image footage. Medians are per task; source minutes and the source/reference ratio exclude **Beauty of Delhi**, which is built from 603 stills. Video files count every video container in a task's inputs. L/P: landscape/portrait; HD: 1920×1080 or 1080×1920; 4K: 3840×2160 or 4096×2160.

		Total	Median per task				Delivery		
Collection	Tasks	source hours	source minutes	reference seconds	source/ reference	video files	format	fps	window (s)
EditStock	11	18.8	69.5	67.1	57.3	96	L; HD, 4K	23.976, 24	29–305
Cinestudy	15	10.3	27.0	110.9	14.6	2	L; HD	23.976, 24, 25	35–315
UGC	15	1.4	5.7	39.1	8.4	50	P; HD	25	25–55
Commercial	15	2.4	9.6	63.1	8.9	76	L; HD	24	44–125
All	56	33.0	12.4	60.3	12.0	55.5	41 L, 15 P	23.976, 24, 25	25–315

A.1 EDITING SKILLS

Two model annotators, Claude Opus 5.5 and Claude Sonnet 5, independently tagged each of the 56 tasks with the 16 editing skills of Figure 3. For each task they read the brief, the requirements of its brief-test rubric and an inventory of the supplied material, and every tag had to cite a short verbatim quote from this evidence. A skill counts as required only if the brief or its paperwork asks for it, or if the supplied material and the brief make it unavoidable; a skill that would merely be good practice is not tagged. The annotators agree on 831 of the 896 task–skill pairs (92.7%; Cohen's $\kappa = 0.85$). A third, adjudicating model pass resolved the 65 disagreements against the quoted evidence, and the authors cross-checked every tag against its quoted evidence.

B TASK VERIFICATION AND QUALITY CONTROL

Figure 6 summarizes the verification procedure of Section 2.2.

Retrieval rule. Every brief contains two paragraphs common to all tasks, which point the agent to the tool documentation and state: "Do not search for, retrieve, watch, or copy finished edits or reference videos for this project, including public examples. Web search and fetching are allowed for tool documentation and general editing techniques."

Oracles. Each task's oracle builds a mechanical assembly: a few supplied shots, normalized and concatenated or looped to the middle of the duration window, with a supplied audio stem underneath.

Reference isolation. The MD5 hashes of all 56 original reference files match none of the 5,003 task input files, so no reference edit is staged in any workspace. We scanned the complete traces of all 896 runs, including tool calls and their outputs. No run made a web-fetch call, downloaded a reference edit or accessed the reference store; the only web activity is 29 web searches in 18 computer-use runs, and no reference identifier appears in their traces.

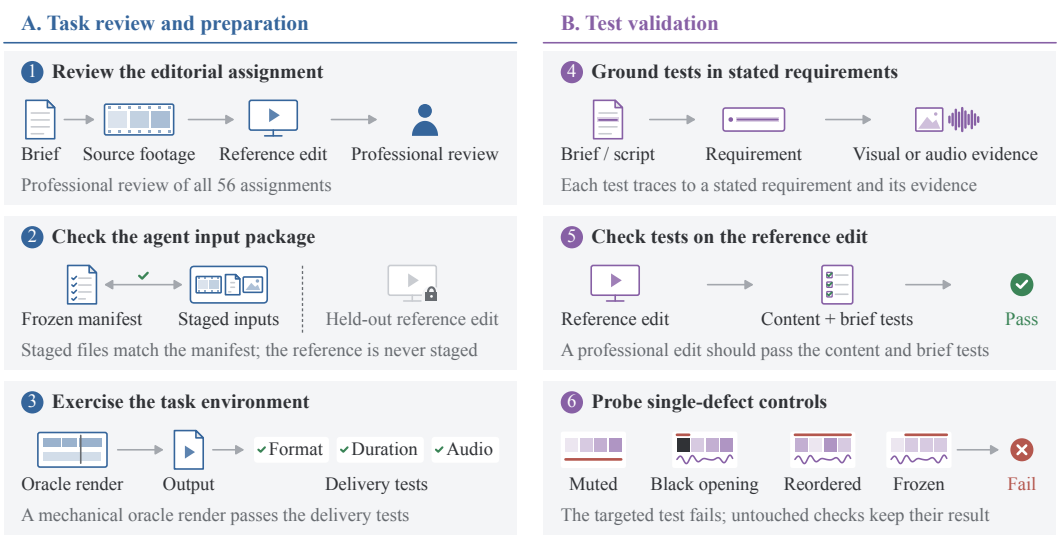

Figure 6: **Our task verification process.** (A) Each assignment is reviewed for consistency between brief, source assets and reference edit; its staged inputs are matched to a frozen manifest with the reference edit held out; and a mechanical oracle exercises the delivery tests. (B) Each content and brief test is grounded in the brief and in visual or audio evidence, checked on the reference edit, and probed with single-defect controls that should fail the targeted test.

C Content Tests

Table 4 lists the six content tests and their limits.

Table 4: **Content tests.** Each content test fails a video whose measurement exceeds the limit.

Content test	Measurement	Limit
Mostly silent	share of the runtime that is silent (no audio track also fails)	0.50
Mostly frozen	share of the runtime with a frozen picture	0.60
Repeated footage	duration of footage shown more than once	1 s
Dead air	longest internal silence	5 s
Channel imbalance	level difference between the left and right channels	6 dB
Faces cut by the frame edge	share of frames with a face in which a face is cut by the edge	0.30

D Quality Test

D.1 Rubric

The judges receive the study rubric verbatim.

Study rubric

Purpose: "We are comparing complete video edits for their stated audience and purpose. There is no expected winner." *Scope:* "Assess editorial craft. Mandatory brand, asset, and technical delivery requirements are evaluated separately. Use the same stated audience and purpose for both versions." *Acceptable:* "The edit communicates its message clearly, with coherent assembly, suitable pacing, and effective picture and sound. Different creative approaches can all be acceptable." *Needs revision:* "An editorial weakness meaningfully harms the viewing experience: for example, confusing structure, distracting repetition, unsuitable pacing, or disruptive picture or sound. Judge viewer impact, not the time needed to fix it." *Minor issues:* "Optional stylistic changes and minor imperfections alone do not require rejection. 'Both acceptable' does not mean equal quality. Either or both versions may need revision." *Acceptability question:* "Does this version meet a professional standard of editorial quality for the stated audience and purpose?"

D.2 Scoring

Each pass is a single call in which the judge returns integer scores from 1 to 10 for five dimensions (story and assembly, pacing, picture, sound and graphics), up to five timestamped defects, a yes-or-no answer to the acceptability question and an integer overall score from 1 to 10, where 10 is the best professional edit to expect for the project, 6 is acceptable and 3 or lower needs substantial revision. A judge's score of an edit is $\overline{o} + 0.2\,\overline{d}$, where $\overline{o}$ is the overall score and $\overline{d}$ the mean of the five dimension scores, both averaged over the judge's passes (Gemini 3.8 Flash makes two); the dimension term mainly breaks ties between equal overall scores, and the defects and the acceptability answer are not used.

D.3 Margin Calibration

The tie margin t_C of collection C is fitted on the collection's study pairs, each a delivered agent edit with its task's reference edit, and on the human editors' assessable votes on those pairs. Let q_C be the share of these votes that prefer the agent edit or report no meaningful preference as per human judgement, and let the gap g of a pair be the panel score of the agent edit minus that of the reference edit as per the judge model. Then t_C is the smallest observed gap x at which the share of the collection's pairs with $g \geq x$ does not exceed q_C. The frozen margins are fitted in this way on all 2,582 assessable votes on the 863 pairs.

D.4 Individual Edits

The quality test is valid per agent, not per edit: on single edits the panel agrees with the human editors' majority at $\kappa = 0.15$, about as well as one human editor agrees with the majority of the others ($\kappa = 0.14$), and the per-task pass rate correlates with the human per-task rate at only $\rho = 0.32$ across the 56 tasks.

E Brief Compliance

Each judge-based brief test asks a local, binary question about visible content, never quality, that can be answered from the picture alone and is phrased so that yes means the requirement is met. Gemini 3.8 Flash watches the edit with sound, GPT-6 Astra reads contact sheets at 1 fps, and Claude Opus 5.5 reads denser sheets of up to 3 fps over the span the question covers. Each judge answers yes, no or cannot tell, and a two-of-three majority of yes-or-no answers decides the test.

F Human Study

Experience. Human editors reported their professional editing experience in four bands: less than one year (4), one to three years (27), four to seven years (8) and eight years or more (4).

Order of the versions. Human editors preferred the agent edit in 10.5% of judgments when it was Version A and in 11.7% when it was Version B, a difference of -1.2 points (95% CI -3.7 to 1.1, bootstrap over tasks).

G Statistical Methods

Resolution rates. An agent's resolution rate is a binomial proportion over the 56 tasks, and the overall rate one over the 896 runs; both are reported with exact Clopper–Pearson 95% intervals (Clopper & Pearson, 1934).

Two-way intervals. Human win-or-tie rates depend on two crossed samples, the tasks and the human editors. We therefore draw three families of 2,000 bootstrap replicates that resample tasks, human editors or individual judgments with replacement, and recompute every rate in each replicate. Let V_{task}, V_{editor} and V_{vote} be the replicate variances of a rate under the three schemes. Each one-factor

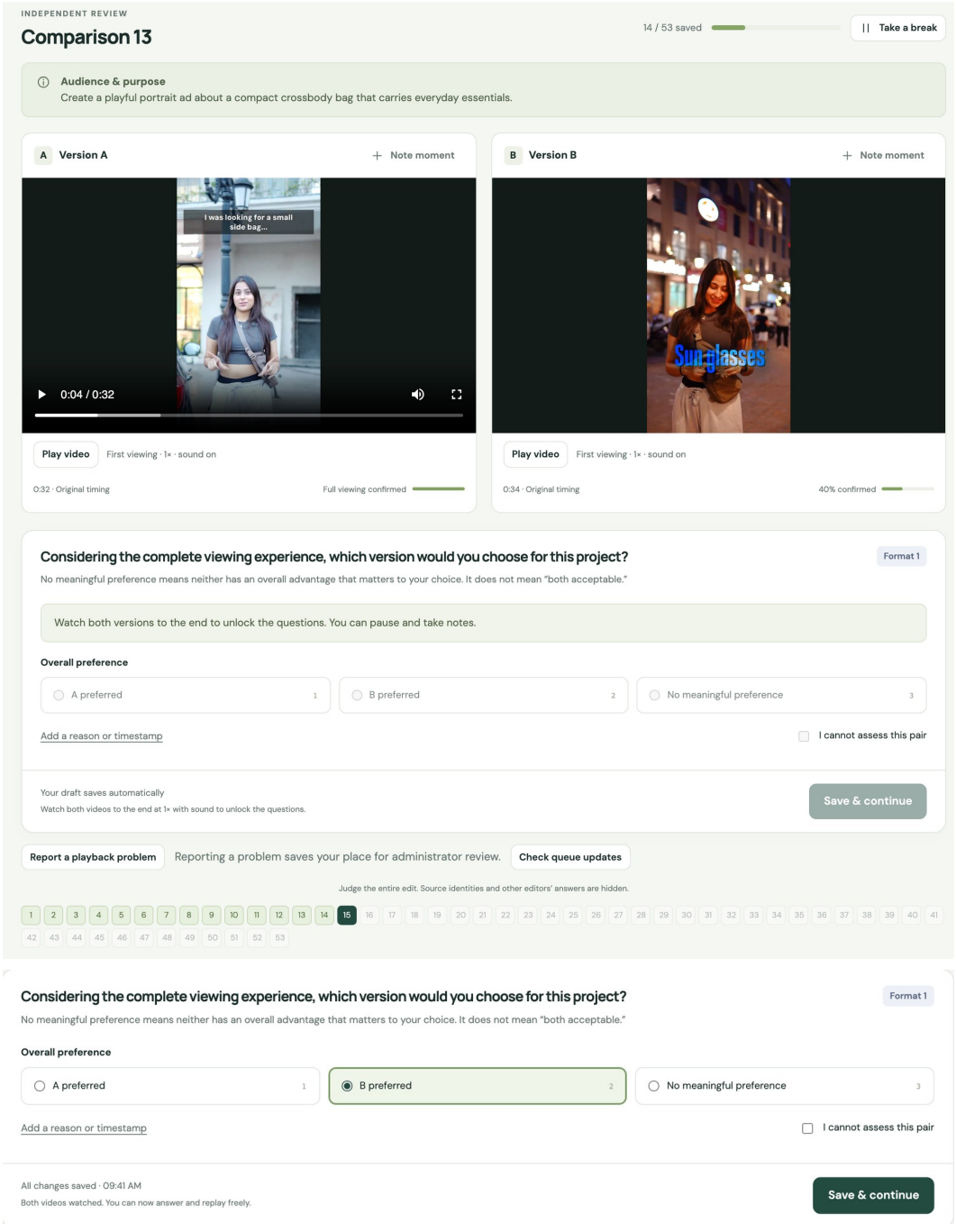

Figure 7: **The comparison page of the study interface.** Top: before both videos have been watched to the end, the answer controls are disabled; Version A has been viewed in full, so its seek bar is shown, while Version B is still in its first viewing. Bottom: the answer panel after both viewings.

bootstrap also carries the judgment-level noise, so the sum of the first two counts that noise twice (Owen, 2007), and the moment-corrected estimator subtracts one copy,

$$\widehat{\mathrm{SE}}^2 = \max\big(V_{\text{task}} + V_{\text{editor}} - V_{\text{vote}},\ V_{\text{task}}\big), \tag{1}$$

where the floor keeps the interval at least as wide as the task bootstrap's. The 95% interval is the estimate $\pm 1.96\,\widehat{\mathrm{SE}}$, truncated to $[0, 100]$.

G.1 Reliability of Per-Agent Rates

Single judgments. Over the 2,582 assessable judgments, Krippendorff's α (Krippendorff, 2018; Hayes & Krippendorff, 2007) with the ordinal metric is 0.103.

Per-agent rates. For every pair with three assessable judgments, we form one list of 16 win-or-tie rates from one judgment and a second list from the other two. Averaged over 3,000 random choices of the held-out judgment, the Pearson correlation of the two lists is $r = 0.6997$. Writing a rate computed from one judgment per pair as a true rate plus independent noise, $X = T + E$, with reliability $\rho_1 = \mathrm{Var}(T)/\mathrm{Var}(X)$, the two lists share only T, and averaging two judgments halves the noise variance, so

$$r = \rho_1 \sqrt{\frac{2}{1+\rho_1}}, \qquad \rho_1 = \frac{r^2 + \sqrt{r^4 + 8r^2}}{4} = 0.632. \tag{2}$$

The reported rates average three judgments per pair, so by the Spearman–Brown formula (Spearman, 1910; Brown, 1910) their reliability, the correlation predicted between two lists of win-or-tie rates each computed from its own three human editors, is

$$\rho_3 = \frac{3\rho_1}{1+2\rho_1} = 0.838. \tag{3}$$

Leaving out human editors. When each of the 43 human editors is left out in turn, the ranking of agents by the share of judgments preferring the agent edit keeps a Spearman correlation of $\rho \geq 0.964$ with the full ranking.

H Agent Configurations and Runtime

Table 5 gives each agent's model identifier, provider route and reasoning setting.

H.1 Launch Commands

Each coding run stages the task's inputs, writes the brief to `/workspace/brief.md` byte for byte as the task's `instruction.md`, and launches the stock CLI without interaction, with one of these commands for OpenCode, Codex CLI and Claude Code; *model* and *prompt* stand for the identifier in Table 5 and the prompt below:

```
opencode run --model model --format json prompt
codex exec --model model --json --skip-git-repo-check
--dangerously-bypass-approvals-and-sandbox prompt
claude --print --model model --effort max --output-format stream-json
--verbose --dangerously-skip-permissions --settings settings prompt
```

OpenCode reads its configuration from the `OPENCODE_CONFIG_CONTENT` environment variable. Codex CLI is configured with `model_reasoning_effort` and `plan_mode_reasoning_effort` set to `max`, and the default subagent model and effort set to the selected model and `max`. Claude Code runs with `CLAUDE_CODE_EFFORT_LEVEL=max`, with the subagent model and every model alias pinned to the selected model, and with *settings* that allow only the selected model.

The prompt is a single message, identical for every coding agent:

Table 5: **Agent configurations.** Model identifiers are the exact strings sent to the provider. OpenRouter routes are restricted to the named provider with fallbacks disabled. Every reasoning setting is the provider's maximum and also applies to helper and subagent roles. DeepSeek Flash is the `deepseek-flash` alias, served by DeepSeek-V4.1-Flash.

Model	Model identifier	Provider route	Reasoning setting
OpenCode 1.18.31			
Gemini 3.1 Pro Preview	`google/gemini-3.1-pro-preview`	Google API	`thinkingLevel: high`
Gemini 3.8 Flash	`google/gemini-3.8-flash`	Google API	`thinkingLevel: high`
Claude Fable 5.1	`anthropic/claude-fable-5-1`	Anthropic API	adaptive thinking, `effort: max`
Claude Opus 5	`anthropic/claude-opus-5`	Anthropic API	adaptive thinking, `effort: max`
GPT-5.6 Sol	`openai/gpt-5.6-sol`	OpenAI API	`reasoningEffort: max`
GPT-6 Astra	`openai/gpt-6-astra`	OpenAI API	`reasoningEffort: max`
DeepSeek Flash	`deepseek/deepseek-flash`	DeepSeek API	`reasoning_effort: max`
Grok 4.6	`openrouter/x-ai/grok-4.6`	OpenRouter, xAI	`reasoning.effort: xhigh`
GLM 5.3 Flash	`openrouter/z-ai/glm-5.3-flash`	OpenRouter, Z.ai (FP8)	`reasoning.effort: max`
Qwen 3.8 Max	`openrouter/qwen/qwen3.8-max-0902`	OpenRouter, Alibaba	`xhigh`
Codex CLI 0.155.1			
GPT-5.6 Sol	`gpt-5.6-sol`	OpenAI Responses	`reasoning.effort: max`
GPT-6 Astra	`gpt-6-astra`	OpenAI Responses	`reasoning.effort: max`
GPT-6 Astra, curated guidance	`gpt-6-astra`	OpenAI Responses	`reasoning.effort: max`
Claude Code 2.1.278			
Claude Fable 5.1	`claude-fable-5-1`	Anthropic Messages	adaptive thinking, `output_config.effort: max`
Claude Opus 5	`claude-opus-5`	Anthropic Messages	adaptive thinking, `output_config.effort: max`
Codex 0.155.1 with Computer Use, DaVinci Resolve on macOS			
GPT-6 Astra	`gpt-6-astra`	OpenAI API	`model_reasoning_effort: max`, plan mode `max`

```
Read the brief at /workspace/brief.md and complete it.  Available
tools and environment are documented at /opt/benchmark-tools/README.md.
Follow the brief's deliverable requirements and output paths exactly.
```

(A resource summary measured when the container starts, with advice to fit process counts, render concurrency and batch sizes to the available CPU, memory and disk.)

The computer-use agent is launched with:

```
codex exec --json --ephemeral --skip-git-repo-check --enable
computer_use -m gpt-6-astra -C workspace -c features.shell_tool=false
-c features.code_mode=false -c sandbox_mode="read-only" -c
model_provider="openai" -c openai_base_url=OpenAI API endpoint -c
forced_login_method="api" -c model_reasoning_effort="max" -c
plan_mode_reasoning_effort="max" -c check_for_update_on_startup=false -c
approval_policy="never" -c service_tier="default" -c
agents.default_subagent_model="gpt-6-astra" -c
agents.default_subagent_reasoning_effort="max" -
```

Here *workspace* is the task's Mac workspace, and the final dash makes Codex read its prompt from standard input.

H.2 Execution Environment

Table 6 gives the coding agents' Linux environment. The delivery tests run afterwards on the output files in the task's own image (Debian 13.7, ffprobe 7.1.5, pytest 8.4.2, Python 3.12.14), without network access and with 4 CPUs and 8 GB of memory.

H.3 Cost and Runtime

Runtime is the CLI's wall-clock time per run. OpenCode and Claude Code record each run's model cost. Codex CLI records token counts but no dollar cost, so we estimate its cost from the tokens at

Table 6: **Linux execution environment of the 15 coding agents.**

Component	Version or content
Base image	`node:22.22.2-bookworm-slim` (Debian 12), linux/amd64
Node.js	22.22.2
Python	3.11 virtual environment with Pillow 11.3.0, NumPy 2.2.6, SciPy 1.15.3, soundfile 0.13.1, OpenCV (headless) 4.12.0.88 and pypdf 6.0.0
FFmpeg and ffprobe	5.1.9 (Debian 12 package)
Remotion	4.0.524 with its CLI and media packages; React 19.3.0
HyperFrames	0.8.40 with GSAP 3.15.0
Browser	Chrome Headless Shell 149.0.7790.0
Documents	Poppler utilities (Debian 12 package)
Transcription	`transcribe` command calling AssemblyAI Universal-3.5 Pro

base-rate prices in dollars per million tokens (uncached input, cache read, cache write, output): 10, 1, 12.5 and 50 for GPT-6 Astra, and 4, 0.4, 5 and 20 for GPT-5.6 Sol. Output tokens include reasoning tokens.

Table 7: **Runtime and model cost per agent,** over all 56 runs of each agent. An asterisk marks token-based cost estimates for Codex CLI.

Model	Harness / guidance	Median runtime per run (min)	Mean cost per run ($)
GPT-6 Astra	OpenCode	26.2	9.67
Claude Fable 5.1	OpenCode	60.6	16.96
Claude Opus 5	OpenCode	51.4	13.83
GPT-5.6 Sol	OpenCode	27.1	5.96
Grok 4.6	OpenCode	25.9	3.46
Gemini 3.8 Flash	OpenCode	25.1	6.96
GLM 5.3 Flash	OpenCode	49.4	0.23
DeepSeek Flash	OpenCode	33.5	0.13
Qwen 3.8 Max	OpenCode	48.1	2.39
Gemini 3.1 Pro Preview	OpenCode	15.9	3.45
Claude Opus 5	Claude Code	58.5	20.90
GPT-6 Astra	Codex CLI	19.6	8.71*
Claude Fable 5.1	Claude Code	57.0	41.27
GPT-5.6 Sol	Codex CLI	17.9	5.00*
GPT-6 Astra	Codex CLI, curated guidance	21.4	9.76*
GPT-6 Astra	Codex CLI, computer use	52.7	41.61*

I Detailed Results

I.1 Matched Contrasts

Table 9 compares pairs of agents that differ in one component, matching their edits task by task. No harness or guidance contrast is significant, while computer use significantly lowers both the win-or-tie rate and the number of resolved tasks.

Table 8: **Main results with 95% confidence intervals**, one run per task (Table 1).

Model	Harness / guidance	Resolution rate % [95% CI]	Human win-or-tie % [95% CI]
GPT-6 Astra	OpenCode	21.4 [11.6, 34.4]	22.3 [14.8, 29.8]
Claude Fable 5.1	OpenCode	17.9 [8.9, 30.4]	22.9 [15.0, 30.9]
Claude Opus 5	OpenCode	17.9 [8.9, 30.4]	19.0 [11.6, 26.4]
GPT-5.6 Sol	OpenCode	17.9 [8.9, 30.4]	17.6 [9.4, 25.8]
Grok 4.6	OpenCode	12.5 [5.2, 24.1]	17.0 [10.1, 23.9]
Gemini 3.8 Flash	OpenCode	10.7 [4.0, 21.9]	13.1 [7.3, 18.9]
GLM 5.3 Flash	OpenCode	10.7 [4.0, 21.9]	9.5 [4.9, 14.2]
DeepSeek Flash	OpenCode	7.1 [2.0, 17.3]	13.0 [6.9, 19.2]
Qwen 3.8 Max	OpenCode	3.6 [0.4, 12.3]	4.0 [0.2, 7.8]
Gemini 3.1 Pro Preview	OpenCode	1.8 [0.0, 9.6]	9.0 [3.9, 14.0]
Claude Opus 5	Claude Code	23.2 [13.0, 36.4]	23.2 [15.3, 31.1]
GPT-6 Astra	Codex CLI	21.4 [11.6, 34.4]	23.8 [16.2, 31.4]
Claude Fable 5.1	Claude Code	17.9 [8.9, 30.4]	23.5 [15.8, 31.2]
GPT-5.6 Sol	Codex CLI	8.9 [3.0, 19.6]	13.1 [7.6, 18.6]
GPT-6 Astra	Codex CLI, curated guidance	26.8 [15.8, 40.3]	24.4 [17.1, 31.7]
GPT-6 Astra	Codex CLI, computer use	3.6 [0.4, 12.3]	8.2 [3.0, 13.3]
All agents		14.0 [11.7, 16.4]	16.5

Table 9: **Matched contrasts**, first minus second. n: tasks with both edits judged. ΔW/T: difference in human win-or-tie rate over these tasks, in points, with 95% CI and Holm-adjusted paired-permutation p. *Resolved*: tasks resolved by each agent (Holm-adjusted exact McNemar p).

Contrast	n	ΔW/T [95% CI]	p	Resolved (p)
GPT-5.6 Sol: Codex CLI − OpenCode	55	−4.2 [−13.2, +4.7]	1.00	5 vs 10 (0.63)
GPT-6 Astra: Codex CLI − OpenCode	53	+0.9 [−7.5, +9.4]	1.00	12 vs 12 (1.00)
Claude Fable 5.1: Claude Code − OpenCode	56	+0.6 [−9.6, +10.8]	1.00	10 vs 10 (1.00)
Claude Opus 5: Claude Code − OpenCode	51	+5.9 [−2.9, +14.7]	1.00	13 vs 10 (1.00)
GPT-6 Astra: curated guidance − none	56	+0.6 [−8.4, +9.6]	1.00	15 vs 12 (1.00)
GPT-6 Astra: computer use − code	53	−16.4 [−24.0, −8.7]	<0.001	2 vs 12 (0.038)

J Extended Analysis

J.1 Where Runs Fail

Only 6 of the 863 delivered videos fail a delivery test, and 762 pass every brief test. The most common brief-test failure is a required spoken line that is not heard: code agents, which hear through transcription, miss 5.9% of required lines, and the computer-use agent, which cannot hear, misses 41.5%.

J.2 Why Human Editors Prefer the Reference Edits

Per-category shares of the notes are in Table 10. When human editors prefer the agent edit, they tick story and assembly more often than when they reject it (84% against 72% of tagged judgments) and graphics much less often (35% against 55%). Human editors report slates, crew talk or retakes left in 18.5% of the computer-use agent's losses against 1.4% for the six agents with the highest human win-or-tie rate, and repeated footage in 18.5% against 3.6%. An agent edit loses 6.6 points of win-or-tie per halving of its cut rate below the reference edit's (95% CI 3.2–10.0), whereas cutting faster than the reference edit is not penalized.

J.3 Task Difficulty

A Rasch model of the human votes, with an ability per agent and a difficulty per task (Appendix K.2), places agents and tasks on one scale (Figure 8). Tasks vary 2.6 times as much as agents, and on 54 of the 56 tasks even the best agent has less than even odds of a win or tie. Collection explains half of the variation in task difficulty ($\eta^2 = 0.51$). Within collections, no task descriptor or required skill predicts difficulty after Holm correction (all $|r| \leq 0.24$), and once each task's difficulty is removed, the agent-by-skill grid of win-or-tie rates is at chance level (permutation $p = 0.55$).

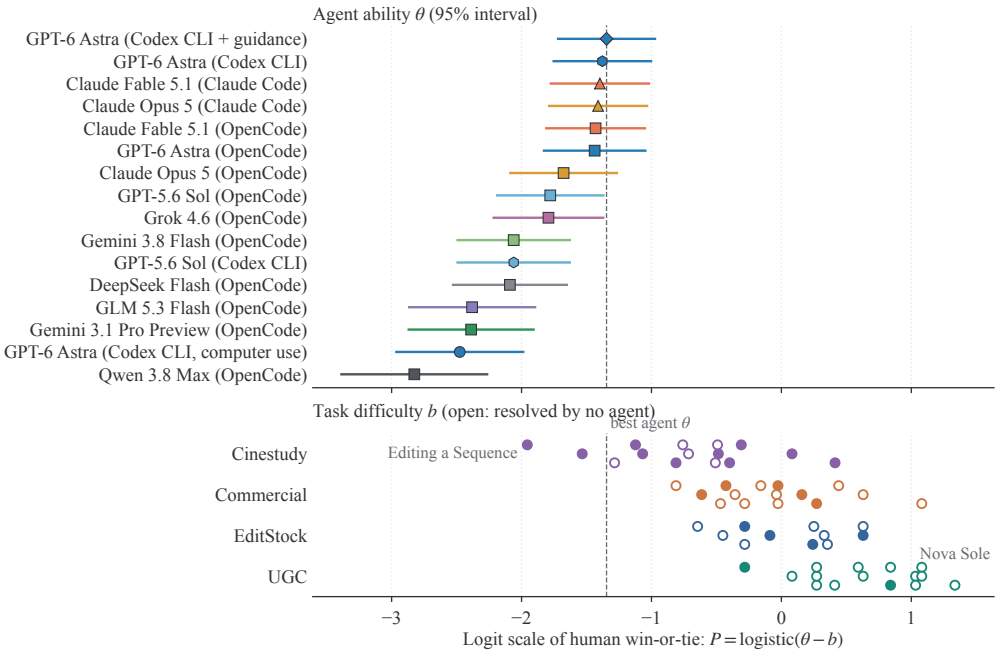

Figure 8: **Tasks vary more than agents.** Agent ability (top) and task difficulty (bottom) on one logit scale, from a Rasch model of the human win-or-tie votes; an agent has even odds of a win or tie on a task whose difficulty equals its ability. Open circles: tasks resolved by no agent. The dashed line marks the best agent.

J.4 Trajectory-Level Analysis

Within an agent, no process feature is associated with the human win-or-tie rate or with resolution after Holm correction (Figure 9). A check-then-re-render loop goes with a higher human win-or-tie rate when agents are compared on the same task ($+2.1$ points per standard deviation, Holm $p = 0.03$), but within an agent the association shrinks to $+1.5$ points and does not survive correction.

K Analysis Methods

K.1 Coding of Human Editors' Notes

Material. Human editors answered the optional question "What most influenced your judgment?" by ticking dimensions (story and assembly, pacing, picture, sound, graphics, other), by writing a note, or both. Of the 2,582 assessable judgments, 931 have at least one ticked dimension (tagged judgments), and 1,271 have a note.

Coding. Claude Opus 5.5 coded each note into claims, each with one of 24 categories (Table 10), a target (agent edit, reference edit or both) and a polarity (problem or strength). Comparative praise of the preferred edit is coded as a strength of that edit, not as a problem of the other.

Validation. Claude Sonnet 5 re-coded 150 random notes with the same prompt. On the presence of each category in a note, the two coders agree at a mean Cohen's κ of 0.84 over the 19 categories with at least ten positives.

K.2 Task Difficulty and Skill Profiles

Rasch model. We fitted a crossed random-effects logistic model, $\text{logit}\, p = \mu + \theta_{\text{agent}} - b_{\text{task}}$, to the human win-or-tie votes (863 delivered runs, three votes each) by Laplace-approximate marginal likelihood. The task standard deviation is 0.83 against 0.51 for agents, a variance ratio of 2.6.

Table 10: **Why human editors prefer the reference edit**, by category: share of the 1,101 notes on judgments preferring the reference edit that name a problem of the agent edit (AP), a strength of the reference edit (RS), or either (95% interval from resampling human editors).

Category	AP (%)	RS (%)	Either (%)
Overall polish	13.8	46.9	52.8 [42.9, 63.0]
Shot selection	10.0	29.3	34.2 [24.7, 42.9]
Text and graphics	8.9	29.4	31.8 [24.8, 38.9]
Transitions and effects	6.7	26.6	29.2 [18.4, 40.3]
Story and structure	6.9	25.1	27.6 [19.8, 36.0]
Sound design	5.1	20.9	23.1 [15.7, 30.6]
Color	3.7	19.0	20.4 [11.7, 31.2]
Pacing	2.1	17.2	18.4 [12.8, 24.4]
Music choice	3.5	15.7	17.3 [11.8, 22.8]
Cut quality	5.5	11.0	15.1 [10.7, 19.4]
Hook and opening	2.6	9.7	10.9
Framing	2.7	7.7	9.4
Branding and product	1.6	8.2	8.6
Audio defects	7.2	1.6	8.1 [4.6, 11.9]
Repeated footage	7.3	0.2	7.3 [3.0, 12.2]
Ending	2.6	4.1	5.9
Dialog and voice-over	2.8	3.7	5.9
Sync	3.3	1.9	4.5
Mix and levels	1.4	2.9	4.2
Text placement	1.5	1.7	2.8
Picture defects	1.5	0.5	2.0
Silence	0.7	0.1	0.8
Length	0.1	0.2	0.3

Task features. We correlated the Rasch task difficulty with 20 task descriptors (collection, portrait delivery, requested duration, number and duration of source files, presence of a script, and the 16 editing skills), after removing collection means, with collection-stratified permutation tests and Holm correction.

Skill profiles. For each run we subtracted the mean outcome of the other 15 agents on the same task, which removes task difficulty, and compared these residuals between tasks that do and do not require each skill, within collections, testing all agent–skill cells jointly for heterogeneity by permutation.

K.3 Trajectory Phases and Process Features

Phase assignment. We assigned every native action of the 896 traces to one phase by rules over the tool and its arguments. Reading the brief, documentation or file listings is orienting; frame extraction, contact sheets, image reads of source frames, `ffprobe` of inputs, transcription and level measurement of source files are perceiving the source; writing an edit list, plan or to-do list is planning; any command that writes the output file, or an intermediate render, is building; any command that reads, measures, transcribes or extracts frames from the agent's own render is verifying; and a build that follows a verification of the same output is repairing. Commands that match several phases take the latest phase in this order.

Process and outcome. Figure 9 shows eleven of the 17 process features; Holm correction is over all 34 tests (17 features, two outcomes).

Self-reports. Claude Opus 5.5 labeled the final agent message of each of the 896 runs as claiming full success, partial success or failure, or as a progress note without a claim, and the authors checked the labels.

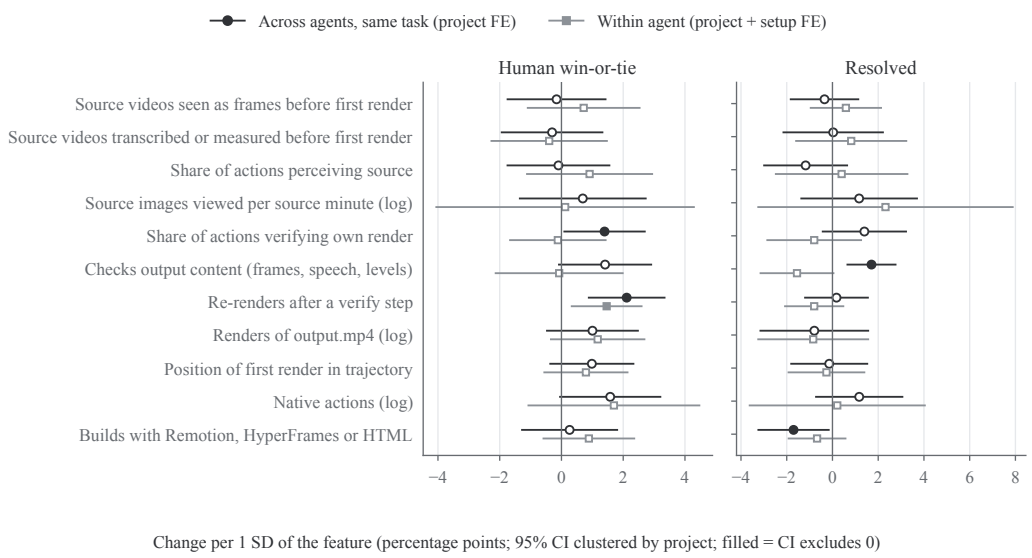

Figure 9: **Process features and outcomes.** Change in the human win-or-tie rate (left) and in resolution (right) per standard deviation of each process feature, in linear probability models with task fixed effects (comparing agents on the same task; circles) and with task and agent fixed effects (comparing runs of the same agent; squares). Filled markers: the 95% interval, clustered by task, excludes zero (before correction for multiple tests). 810 delivered code-agent runs.

L Command-Failure Analysis

We labeled all 55,628 recorded shell commands of the 15 code agents with GPT-6 Astra as a judge. Of the 52,737 commands with a decided label, 6.0% fail.

M Release and Licensing

Code, task definitions, rubrics, frozen constants and results of Timeline-Bench 1.0 are released under Apache-2.0. Each Harbor task holds its configuration, the brief, an environment definition, an input-staging script, an oracle solution and the four delivery tests; the verifier package adds the content tests, the brief tests with their 56 rubrics, and the quality test. The results include every per-run outcome with sanitized judge outputs, and the human study is released as aggregate statistics and per-run vote counts.

Source media are available only to approved research teams, for research and evaluation. The UGC and Commercial productions were commissioned for the benchmark, with rights to the material and releases for the people, locations and brands shown. The access terms forbid redistributing inputs, outputs, stills, frames or clips; training generative models on the inputs or anything derived from them; training or tuning on the benchmark; and identifying, profiling or imitating the people shown.

N Related Evaluation Settings

Table 11 compares Timeline-Bench with the closest evaluation settings by what each receives, what it scores and how it scores it.

Table 11: **Related evaluation settings.**

Benchmark	Input	Output scored	Evaluation	Human judgment of outputs
VEBench (Deng et al., 2026a)	Edited videos and a question	Answer, clip choice or time span	Accuracy; temporal overlap	None
MEDit-Bench (Ogata et al., 2026)	One long video and an editing message	Cut list	Temporal overlap with professional edits	User study on a subset (1,620 evaluations)
AgenticVBench (Cao et al., 2026)	Source videos and a brief or storyboard	Video with a manifest or report	Programmatic verifiers; 1,069 binary rubric items for repurposing	Expert rubric grading; three-editor human baseline on a subset
CutVerse (Hu et al., 2026)	GUI application state and an objective	GUI trajectory	Milestone checks	Not reported
ProSoftArena (Ai et al., 2026)	Real desktop and a task	Final state or artifact	Execution scripts; subjective comparison with human work on creative tasks	No rating reported
MultiMedia-TerminalBench (Heo et al., 2026)	Terminal workspace with media files	File artifact	Task verifiers with binary and partial success	None
GDPval (Patwardhan et al., 2026)	Request and reference files	Work product	Blinded expert pairwise comparison with expert deliverables	Occupation experts
TIMELINE-BENCH (ours)	Raw production material and brief	Rendered video	Delivery tests, content tests, brief tests and a quality test calibrated on human editors, combined into a resolution rate	43 video editors; 2,589 blind judgments against the reference edit