VibePrevisBench: Evaluating Agentic Coding for 4D Scene Generation in Blender

Yuzhe Chen*§ Muxi Diao* Lele Yang Wenbin Xu Xuyang Liu Qile Su
Changyi Liu† Jiankang Chen Kaiyu Jiang Haoyu Zhang Hongyi Fu
Jianxiong Wang Tianke Zhang Haonan Fan Yingxin Li Bingyi Li Xu Wang
Xuanyu Zheng† Bin Wen‡ Tingting Gao Wentao Zhang Jiale Cao‡

Tianjin University Kling AI Beijing University of Posts and Telecommunications
Zhongguancun Academy The Hong Kong Polytechnic University

Website: https://vibeprevisbench.github.io

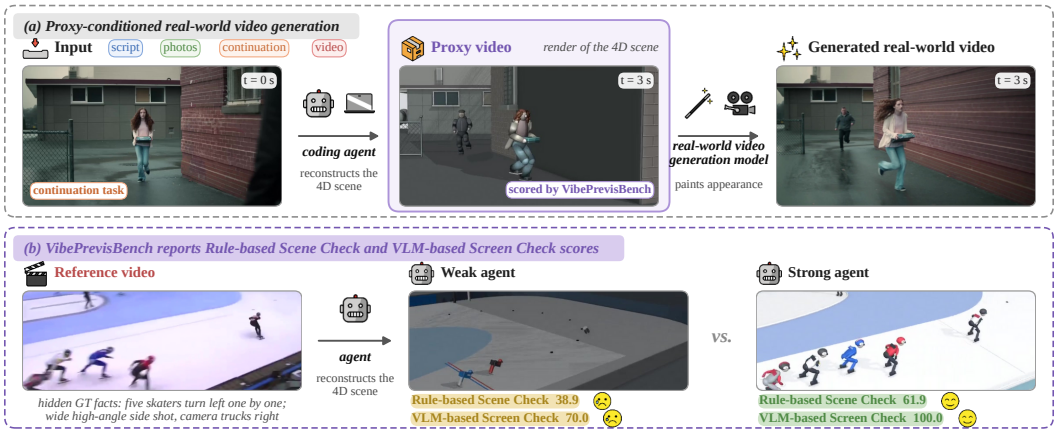


Figure 1. **Proxy-conditioned real-world video generation and what VibePrevisBench evaluates.** (a) From a continuation brief (first frame), a coding agent reconstructs a 4D scene and renders its graybox proxy video on which a controllable real-world video generation model paints appearance [3, 31]. (b) VibePrevisBench scores Rule-based Scene Check of 4D scene and VLM-based Screen Check of its proxy video against hidden ground truth facts.

Abstract

Controllable real-world video generation models can condition on a rendered graybox proxy video, which fixes layout, motion and camera. Coding agents can build such videos by reconstructing a 4D scene in Blender, yet no benchmark checks both these videos and the long, multi-shot scenes behind them. We introduce **VibePrevisBench**, a benchmark for 4D scene reconstruction: an agent turns a brief into an executable 4D scene and renders its proxy video. Its 242 tasks span four input kinds, including scripts, photos, continuations, and reference videos. Each delivery is scored along two evaluation dimensions. **Rule-based Scene Check** re-runs the code and compares the exported 4D scene with hidden ground truth facts in nine dimensions. **VLM-based Screen Check** asks a VLM judge whether the facts that pass can be recognized in the proxy video. We analyze the evaluation results and trajectories of 11 frontier coding agents and identify three key observations. First, multimodal understanding separates the agents, especially in understanding space and camera motion. Second, all agents seem to already master basic camera work and staging, and fail at challenging requests. Third, agents inspect their own renders but fail to correct their errors. Agents thus need better visual understanding of space and camera, explicit checks for challenging requests, and item-by-item verification of renders against the brief.

*Equal contribution. †Project lead. ‡Corresponding author.
§This work was conducted during the author's internship at Kling AI.

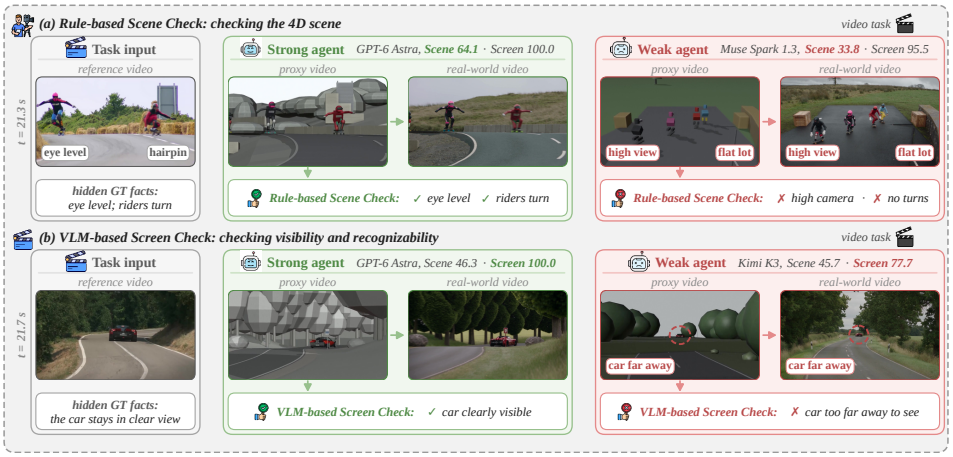


Figure 2. **Two complementary evaluation dimensions.** Each panel compares a reference video with strong and weak agents' proxy videos and the corresponding real-world videos generated by Seedance 2.5. (a) Rule-based Scene Check evaluates the 4D scene, and detects the weak agent's high camera and missed turns. (b) VLM-based Screen Check assesses visibility and recognizability: the weak agent's car passes the in-frame and turning checks but is too distant to recognize clearly. The real-world videos inherit their proxy videos' errors. In addition, we note that the generated real-world videos keep the errors of their proxy videos, demonstrating the importance of proxy video generation quality.

1 Introduction

Controllable real-world video generation models now take previsualization (previs), a rough 3D mock-up of a shot, as input and paint the real-world video on top [3, 31, 42, 60]. This input is a graybox proxy video, the untextured render of a 4D scene whose layout, people, actions, cameras and cuts unfold over time. Conditioning on proxy video makes all of these more controllable [4, 20, 30, 36, 44], but building it by hand takes skilled artists and time [23, 67]. Coding agents can build it instead: from a script, photos, a continuation or real footage, they write Blender code, render previews, inspect their frames and revise [7, 8, 10, 32, 34, 36]. We call this task 4D scene generation: the agent writes the executable 4D scene that a brief describes or shows, and renders its proxy video. Figure 1(a) shows this pipeline. *The agent becomes the director; the real-world video generation model, the renderer.*

The proxy video shapes the final real-world video: the real-world video generation model keeps the camera and layout of proxy video, so an agent's errors carry into the resulted real-world video, as Fig. 2 shows. Therefore, it is important to generate the high-quality proxy video. To evaluate the quality of proxy video, we need to check two things. First, is the 4D scene right? In Fig. 2(a), the task asks for riders turning at eye level, but the weak agent films them from high above a flat lot. The proxy video looks fine, and the VLM judge even gives it 95.5. Thus, it is necessary to find this error by rule-based scene check in 4D scene. Second, can a viewer see it is right? In Fig. 2(b), the car is in frame and turning, so the rules pass it. However, the car is a tiny dot on screen, which can be found by the VLM-based screen check. This happens often: 31% of annotated actions pass the rules but are hard or impossible for a viewer to see in the proxy video, as Appendix C.3 reports.

Existing benchmarks support such an evaluation only in part. Table 1 compares them along four dimensions and shows three gaps. **(i) Narrow inputs.** Benchmarks of dynamic scenes mostly start from text prompts [56, 70] or videos to reconstruct [47, 52, 54]. A director, in contrast, may start from a script, photos of a real location or footage to restage, where each source tests a different skill. **(ii) Static content.** Scene-building benchmarks mostly target single objects, static scenes or short clips without people [13, 18, 19, 47, 65]. None tests a long, multi-shot scene in which several people's routes, event timing, camera moves and cuts must hold together. **(iii) Screen-only evaluation.** Dynamic-scene benchmarks score

Table 1. **Comparison with related benchmarks.** *Staging*: the scene has people, interaction between people or with objects, and actions at given times. *Filming*: camera placement or motion is specified, and several shots are joined by cuts. *Evaluation*: the model builds a *4D scene* that changes over time; it is checked item by item *vs. GT*; the frames are scored *on screen*; *checks/task* counts scored items per task. *Duration*: target length in seconds, the median for ours. ✓ yes, ○ partial, ✗ no, – not reported. BVB reuses the videos and questions of VSI-Bench. Only VibePrevisBench is ✓ in every column.

	Input				Staging				Filming		Evaluation			
Benchmark	Text	Image	Video	#Tasks	Duration (s)	People	Interaction	Timed actions	Camera	Multi-shot	4D scene	vs. GT	On screen	Checks/task
Real-world video generation														
VBench-2.0 [70]	✓	✗	✗	–	–	✓	○	✗	○	○	✗	✗	✓	–
WorldScore [12]	✓	✓	✗	3k	–	✗	✗	✗	✓	✗	✗	○	✓	–
FilmBench [56]	✓	✓	✗	1.2k	–	✓	○	✗	○	✓	✗	✗	✓	–
PROWBench [28]	✓	○	✓	170	5	✓	✓	✓	✓	✗	✗	○	✓	–
CodeVideoBench [36]	✓	✓	✓	22	5	○	○	✗	✓	✗	✗	✗	✓	–
Video understanding														
VSI-Bench [62]	✗	✗	✓	288	∼100	✗	✗	○	✗	✗	✗	✗	✗	18
CameraBench [37]	✗	✗	✓	3k	5.7	○	✗	✗	✓	✗	✗	✗	✗	–
VDC [5]	✗	✗	✓	1k	28	✓	○	✗	○	✗	✗	✗	✗	–
OmniCompass [39]	✗	○	✓	820	354	✓	○	✓	✗	✗	✗	✗	✗	3
Video reconstruction in code														
4DCodeBench [47]	✗	✗	✓	200	∼7	✗	✗	○	✗	✗	✓	○	✓	6
BVB [52]	✗	✗	✓	288	∼100	✗	✗	✗	○	✗	○	✗	✓	18
CineRecon [54]	✗	✗	✓	50	15	✓	✓	✓	✓	○	✓	✗	✓	193–400
Scene building in code														
3DCodeBench [18]	✓	✓	✗	212	–	✗	✗	✗	✗	✗	○	✓	✓	–
BlenderWorld [13]	✓	✓	✓	223	–	○	○	✗	○	✗	○	○	✓	12
Code4Scene [65]	✓	✓	✗	190	–	✗	✗	✗	✗	✗	○	✓	✓	–
CinemaBench [68]	✓	✗	✓	810	∼10	○	✗	✗	✓	✗	✗	✗	✓	–
VibePrevisBench (ours)	✓	✓	✓	242	40	✓	✓	✓	✓	✓	✓	✓	✓	**408**

only proxy or real-world video frames [54, 56], while checks of scene state against ground truth cover only static scenes or short clips [18, 47, 65].

To address these gaps, we introduce **VibePrevisBench**, a benchmark for 4D scene reconstruction: an agent turns a brief into an executable 4D scene and renders its proxy video. Figure 1b shows what it scores: the 4D scene and its proxy video, both checked against hidden GT facts. VibePrevisBench has three key features that match the three gaps mentioned above. **(i) Diverse inputs.** Briefs come in four kinds, scripts, photos, continuations and reference videos, across seven scene domains, as Fig. 3 shows. **(ii) Staged and filmed scenes.** Most tasks stage several people in multi-shot scenes of about 40 s, so routes, event timing, camera moves and cuts are tested together. **(iii) Two evaluation dimensions. Rule-based Scene Check** re-runs the delivered code, exports the 4D scene frame by frame and checks it against hidden GT facts in nine dimensions. **VLM-based Screen Check** asks a VLM judge whether a viewer can recognize each entity, read each action and follow the story in the proxy video.

We evaluate 11 frontier coding agents, and we make three observations. First, multimodal understanding separates the agents, especially in understanding space and camera motion. The gap between the best and the worst agent grows from 12.0 Rule-based Scene Check points on scripts to 19.9 on photos, while weak agents require more rounds facing failed API calls. Second, all agents master only basic camera work and staging, and fail at challenging requests. When the reference films from a low angle, 66% of shots come out at eye level, whereas eye-level shots go wrong only 9% of the time. Third, agents inspect their own renders but fail to correct their errors. Even when agents say their renders match the reference, camera-angle errors are just as frequent as in runs without such claims. These observations call for better visual understanding of space and camera, explicit checks for challenging requests, and item-by-item verification of renders against the brief.

In summary, our contributions are:

- **A benchmark for 4D scene reconstruction.** VibePrevisBench has 242 tasks over four input kinds: scripts, photos, continuations and reference videos, in seven scene domains. Most tasks stage several people in long, multi-shot scenes, where we can test the moving trajectory of people, the event happening time, the camera moves and cuts together.
- **Two evaluation dimensions.** Rule-based Scene Check tests whether the 4D scene is right

by checking it against hidden GT facts, and VLM-based Screen Check tests whether what is right can be seen in the proxy video.

- **A diagnostic study of 11 coding agents.** From their scores, trajectories and ablations, three observations show what separates the agents, where they all fail and why they do not correct themselves.

2 Related Work

Proxy-conditioned real-world video generation. Real-world video generation models follow ever richer conditions, from camera paths [1, 21, 58] and object trajectories [17, 55, 72] to a proxy video [4, 20, 30, 44, 46, 63, 71]. A proxy video is the render of a coarse 4D scene, which previs tools and agents build [11, 23, 32, 43, 67]. It fixes layout, motion and camera at once, so the real-world video inherits the scene's mistakes. Real-world video generation benchmarks evaluate the real-world video: most score it with learned or VLM judges, including benchmarks that test real-world video generation models as world models [2, 12, 27, 33, 40, 50, 59, 70], and some check it against ground truth for short, single-shot physics or motion [6, 14, 29]. Multi-shot benchmarks rely on judges or detected cuts [48, 56, 57], and others check whether a real-world video generation model renders what a program specifies [28, 45]. All of them evaluate the real-world video, not the 4D scene behind it or the coding agent that builds this scene.

Understanding a video by rebuilding it. Video understanding benchmarks test perception through answers, groundings or captions [5, 16, 39, 41, 53, 62, 64], and camera benchmarks show that VLMs still misread camera motion [9, 37]. An answer can be right even when the model's picture of the scene is not, and writing the scene out explicitly exposes such gaps [25, 38]. Rebuilding a video in code takes this furthest, since every object, motion and camera must be written out; existing benchmarks do so for physics clips, indoor scans, embodied demonstrations and short cinematic clips [47, 51, 52, 54]. None of them both rebuilds multi-shot videos of several people and checks the rebuilt scene against ground truth, as the last columns of Tab. 1 show.

Building scenes with code. LLM agents write Blender code for objects, scenes, animation and films [22, 24, 26, 49, 61, 66], and recent code world models already write the 4D scene behind a proxy video as code and leave rendering to a real-world video generation model [7, 8, 15, 34, 36]. Yet benchmarks for such agents mostly target edits, objects or static scenes [13, 18, 19, 65, 69], and camera benchmarks only add cameras to a finished scene [35, 68]. None tests multi-shot scenes in which several people act and the camera moves and cuts.

As Tab. 1 shows, VibePrevisBench fills these gaps: an agent reads a script, photos, a continuation or footage and writes the 4D scene behind a proxy video, which Rule-based Scene Check and VLM-based Screen Check verify against hidden GT facts and on screen. VibePrevisBench thus evaluates the agent that builds the scene, not only its rendered video.

3 VibePrevisBench

VibePrevisBench evaluates 4D scene reconstruction along the dimensions in Tab. 1. Each of its 242 tasks pairs a brief with hidden GT facts that can be checked one by one. Figure 3 shows the four input kinds and seven scene domains they span.

3.1 Task formulation

Briefs and deliveries. A task $\tau = (b, m, E, o)$ gives an English brief b, optional media m, a list E of places and subjects with ids, and output requirements o: the frame count T, frame rate and resolution of the proxy video. The input kinds differ only in what specifies the scene. A script describes it in prose alone, without distances or angles. Photo and reference

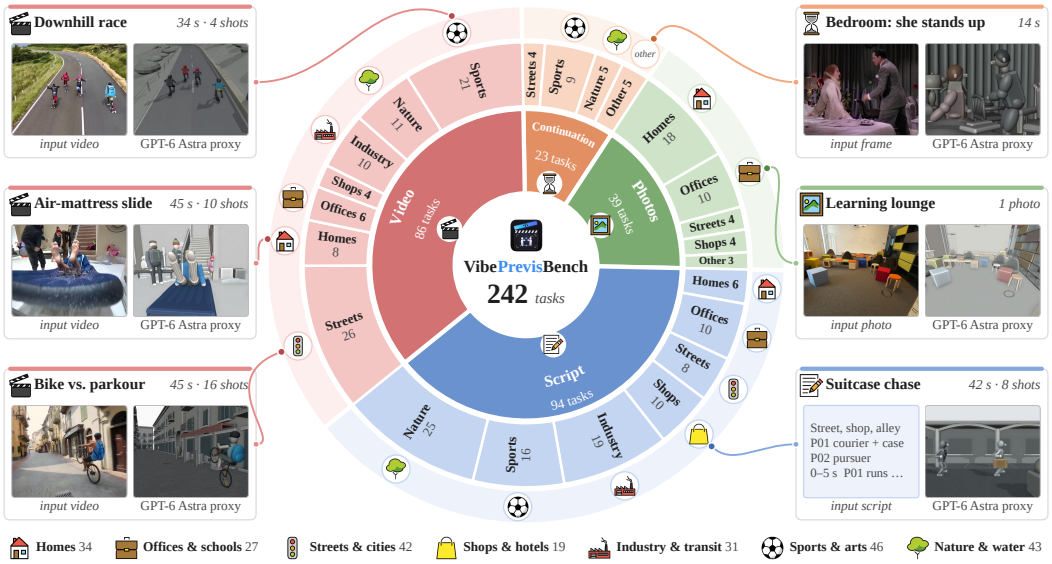


Figure 3. **Overview of VibePrevisBench.** The 242 tasks span four input kinds (inner ring) and seven scene domains (outer ring), with spans proportional to task counts. Each card is a real task with its input and a frame of GPT-6 Astra's proxy video.

video tasks show it in m, as one to five photos or a clip, while b only names the entries of E. A continuation shows the opening frame of a clip and describes the next seconds. The agent answers with a Blender program p. To score it, we run p and read out the 4D scene $S = (\{x_e(t)\}_{e \in E}, c(t), \mathcal{K})$: for every place and subject listed in E, its oriented box $x_e(t)$ at each frame $t = 1, \ldots, T$, together with the camera $c(t)$ and the cut times $\mathcal{K}$. Rendering S gives the proxy video V, with the same T frames.

Rubric. The hidden GT facts form a rubric $F = \{f_j\}$, where each f_j is one checkable statement about the scene. The facts state which subjects exist, where they stand and face, how the places connect, who moves where and when, and how each camera frames, moves and cuts. For example, a fact may require that P01 stops between 3.9 and 4.9 s, within a 1.5 s tolerance. Its score $s_j(S) \in [0, 1]$ measures how well the agent's scene S meets f_j. A fact met within its tolerance scores 1, and a metric fact loses credit linearly once its error exceeds the tolerance. All tolerances are fixed before any agent is run. Each fact belongs to one of nine dimensions: entities, layout, spaces, picture structure, routes, events, camera, cuts and physical validity. Picture structure compares the depth, normals and segmentation of each shot with the reference frames, so only photo, continuation and video tasks have it; Appendix Tab. 3 defines all nine. Appearance is never checked, since the real-world video generation model supplies it. Whatever the input kind, the agent delivers the same kind of program p, which is scored through the same S and V against facts of the same form. Scores thus compare directly across input kinds.

3.2 Rule-based Scene Check and VLM-based Screen Check

Rule-based Scene Check asks whether the scene is right, and VLM-based Screen Check whether what is right can be seen. Figure 4 follows one task through both, and Appendix B.4 validates them with injected defects, re-scoring, three VLM judges and a human study.

Delivery and replay. Each delivery must pass three gates: its program runs on its own, renders the full video and reopens in Blender. A delivery that fails a gate scores 0 on both checks. Such failures stay in the average, so an agent gains nothing by failing hard tasks. Every other program is re-run from scratch in a fresh sandbox, which exports S frame by frame and renders V. Scores come only from this replay, never from videos the agent rendered in its own sandbox, so a video cannot show a scene the program does not build.

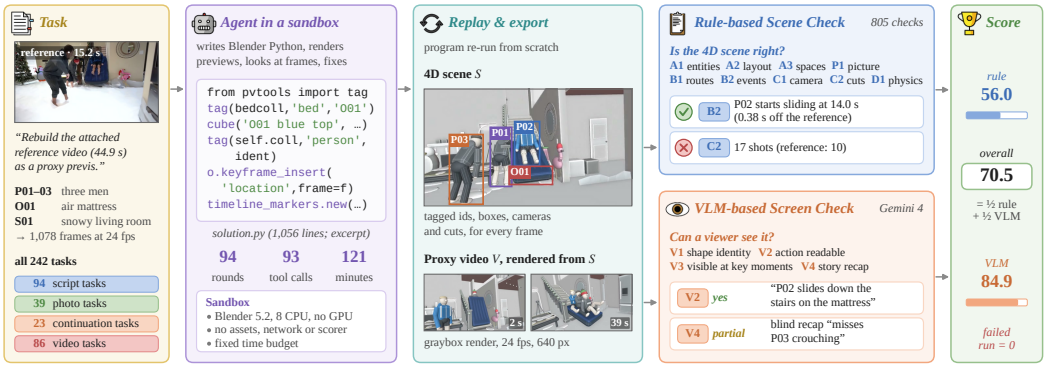


Figure 4. **One task through the evaluation pipeline** (video task M50_049, solved by GPT-6 Astra). The agent writes Blender Python in a sandbox and inspects its own renders. We re-run the delivered program from scratch. The replay exports the 4D scene S, the tagged boxes, cameras and cuts of every frame, and renders the proxy video V from it. Rule-based Scene Check compares S with the hidden GT facts, two of 805 checks shown; VLM-based Screen Check asks the VLM judge about V, only for items the rules passed, two verdicts shown.

Rule-based Scene Check. It binds subjects and places by their tagged ids and matches untagged objects by class. Script facts are relations, since scripts give no coordinates, whereas media facts are metric and are compared after each place is aligned to the ground truth by a similarity transform. The score is then built in three steps. First, the facts of one kind form a block b, whose raw score P_b is the mean of s_j over its facts. Second, each block is corrected for chance and for missing entities: $Q_b = \mathrm{clip}\big((P_b - R_b)/(1 - R_b)\big)\,\kappa_b$, where clip bounds the value to $[0, 1]$. The chance rate R_b is the P_b that the agent's own objects earn when shuffled at random within each place, and κ_b is the share of the block's facts whose entities exist. A random layout thus scores 0, extra clutter raises R_b along with P_b, and every missing entity lowers κ_b. Third, blocks combine into dimension scores Q_d with weights fixed in advance, and Scene $= 100 \cdot \frac{1}{|D_\tau|} \sum_{d \in D_\tau} Q_d$ averages them over the dimensions D_τ that the task exercises. The scorer is deterministic, and Appendix B.2 details the scoring.

VLM-based Screen Check. A VLM judge, Gemini 4, watches the proxy video at 2 fps, plus an orbit render of the delivered scene on photo tasks, blind to the agent's name and the object names. Without the brief, it first retells the clip. It then receives the brief, the reference video on video tasks, and checklists from the hidden GT facts, limited to the items the rules passed, and rates each item as no, partial or yes, $v_{k,i} \in \{0, \frac{1}{2}, 1\}$. The items fall into four dimensions: recognition, action, visibility and story. Recognition counts as yes only if the blind retelling already named the subject or place, and story asks whether the retelling recounts the story beats. With $\bar{v}_k$ the mean verdict of dimension k, Screen $= 100 \cdot \frac{1}{|K_\tau|} \sum_{k \in K_\tau} \bar{v}_k$, where photo tasks use recognition only, and the overall score is (Scene + Screen)/2. Appendix B.3 gives the judge's rubric.

3.3 Construction

VibePrevisBench is built in the three stages of Fig. 5. Appendix A.1 details the taxonomies, the screening rules and the pruning.

Stage 1: Collection and curation. Two frozen taxonomies, of what a scene holds and of how a task is asked, set quotas that new tasks fill. Clips for video and continuation tasks come from over 27k source videos. Keeping a few clips per source with several moving people, camera motion or cuts leaves 1.8k clips, of which 388 pass VLM checks for live action, visible events and distinct subjects. Human reviewers choose among these, and Appendix A.1 describes the steps from these clips to the final video and continuation tasks. Of the 39 photo tasks, 38 draw on scan datasets with measured 3D boxes and camera poses.

Stage 2: Task construction. Every brief is derived from a fixed scene. For a script, a model

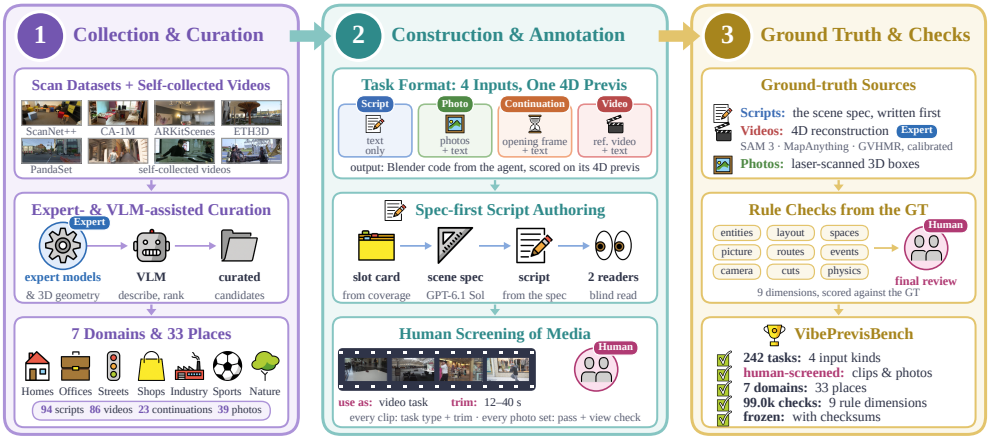


Figure 5. **How VibePrevisBench is built.** (1) Scan datasets and self-collected videos are curated with expert models and a VLM into seven scene domains and 33 places. (2) Scripts are written spec-first and must survive two blind readers; humans select and trim clips and screen photo sets. (3) Hidden GT facts come from the scene spec, 4D reconstruction or scanned 3D boxes, and humans review the resulting checks in nine dimensions.

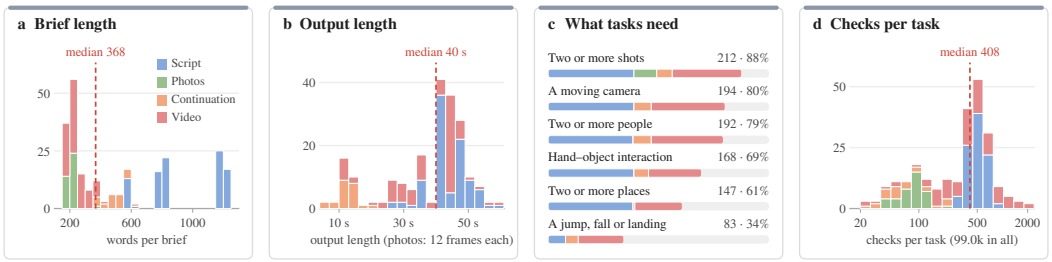


Figure 6. **Task statistics**, stacked by input kind; dashed lines mark medians. (**a**) Brief length in words; script lengths cluster near 600, 800 and 1,200 words. (**b**) Output length; photo tasks show each photo for 12 frames, then sweep the place for 8 s. (**c**) Number and share of tasks with each staging demand, split by input kind. (**d**) Rule checks per task, log scale; one fact can expand into several checks, such as pairwise event orders.

builds the scene from a card and then writes the script, each scored claim citing its sentence. Two blind readers from the Claude and DeepSeek families see only the script and write down where everything is and when it happens; a claim that neither recovers is reworded or dropped. A media brief gives each place and subject an id and a description. Human reviewers check and correct every brief and its hidden GT facts.

Stage 3: Ground truth and checks. Scripts take their hidden GT facts from the built scene, in which every claim is checked to hold, and photo tasks from their scans. For videos, most continuations and one photo task, a pipeline estimates it: it recovers shots, subjects, cameras, depth and bodies and lifts them to 3D, as Appendix Fig. 10 shows. Facts the pipeline cannot verify are marked unscorable and never count. All 11 agents then ran the candidates, and tasks that did not separate them were dropped, judged by item-rest correlation and spread rather than by any single agent's score. Tasks, media, hidden GT facts and scorer are frozen with checksums, so every score can be reproduced from the release.

3.4 Statistics

Coverage and scale. VibePrevisBench contains 242 tasks: 94 scripts, 86 reference videos, 39 photo tasks and 23 continuations. As Fig. 3 shows, each of the seven domains holds 19 to 46 tasks and each of the 33 places 3 to 13. In Fig. 6a, briefs run 153 to 1,200 words with a median of 368: scripts spell out the scene in about 800 words, while media tasks mainly list

Table 2. **Main results on 242 tasks.** Rule-based Scene Check and VLM-based Screen Check per input kind and over all tasks (*All*); VLM-based Screen Check uses the Gemini 4 judge, which rates only recognition on photos; *Overall* is their mean. Failed runs score 0. Per column, the best score is in **bold** and the second best is underlined. The reasoning effort of each agent is given in gray.

Agent	Rule-based Scene Check					VLM-based Screen Check					Overall
	Script	Photo	Cont.	Video	All	Script	Photo	Cont.	Video	All	
GPT-6 Astra max	**85.6**	55.8	**79.8**	**56.8**	**70.0**	**76.5**	93.6	**91.1**	**89.8**	**85.4**	**77.7**
GPT-6.1 Sol max	84.4	55.3	78.1	56.0	69.0	73.3	**93.8**	89.5	87.3	83.1	76.1
Gemini 4 Argon high	82.1	57.1	78.5	55.8	68.4	72.7	90.3	88.8	85.6	81.7	75.0
Opus 5.5 high	83.0	**59.8**	78.2	53.9	68.4	73.0	90.4	82.5	81.6	79.8	74.1
GPT-5.6 Sol max	80.8	44.2	73.0	49.0	62.9	69.4	92.5	81.1	78.7	77.5	70.2
DeepSeek V4.1 Flash max	80.1	43.2	73.0	45.7	61.3	70.8	90.2	82.4	78.8	77.9	69.6
Kimi K3 xhigh	81.3	42.4	73.9	45.2	61.5	69.3	81.2	80.9	72.5	73.5	67.5
Gemini 3.8 Flash high	78.6	43.1	71.4	46.2	60.7	59.1	87.7	77.6	68.3	68.7	64.7
GLM 5.3 Flash high	76.0	39.9	71.0	42.6	57.8	61.0	84.8	76.3	70.2	69.5	63.7
Doubao Seed 2.1 Pro max	73.6	40.4	65.7	42.2	56.3	56.1	86.8	69.1	67.2	66.2	61.3
Muse Spark 1.3 xhigh	77.4	40.5	67.2	40.1	57.2	58.3	84.7	72.5	61.3	65.0	61.1

ids. Outputs run 6 to 60 s with a median of 40 s, as Fig. 6b shows.

Staging demands and checks. Figure 6c shows that 88% of tasks have two or more shots, 80% a moving camera, 79% two or more people, 69% hand–object interaction and 61% two or more places, and 71% combine several shots, a moving camera and two or more people. A task carries 20 to 2,175 checks, 408 at the median and 99.0k in all, as Fig. 6d shows, and exercises a median of eight of the nine dimensions. Events hold 57% of the checks, since event order is checked pair by pair, but Rule-based Scene Check averages dimensions rather than checks, so they do not dominate the score. Appendix A.3 splits them by input kind.

4 Experiments

4.1 Setup

Evaluated agents. We evaluate 11 frontier models as coding agents, each at the reasoning effort given in gray in Tab. 2. The GPT models run in the Codex CLI and all other models in Claude Code. An agent writes and runs Blender code, renders previews, inspects its own frames and revises. A small toolkit tags entities and places with the task's ids, so facts bind to objects without visual matching.

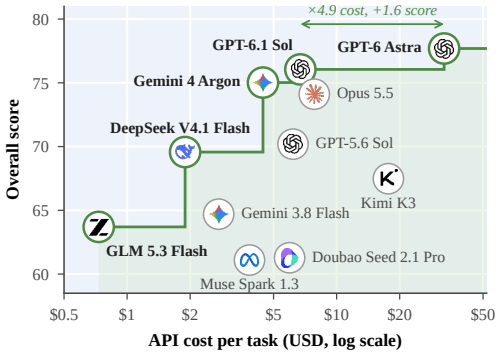


Figure 7. **Cost vs. score.** Mean API cost per task at list prices (log scale) against the overall score; the staircase is the cost–score frontier.

Protocol. Each run takes place in a sandbox with Blender 5.2, eight CPU cores and no GPU. The agent has a budget of 4 hours and no external assets, network or access to the scorer, so it can rely only on what the brief gives it. Every task is run once, and scores are averaged over tasks with equal weight. Input kinds exercise different dimensions and obtain their hidden GT facts differently, so agents are compared within a kind.

4.2 Main results

Table 2 reports both checks per input kind; Appendix C.1 adds per-dimension profiles. GPT-6 Astra leads with an overall score of 77.7, ahead of GPT-6.1 Sol at 76.1, Gemini 4 Argon at 75.0 and Opus 5.5 at 74.1. Five of the ten adjacent pairs cannot be separated by

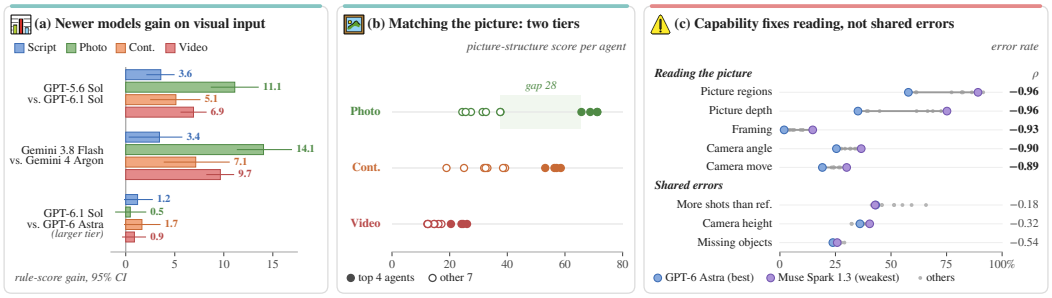


Figure 8. **Where agents improve, and where they do not.** (a) Rule-based Scene Check gain of the second model over the first in each pair, per input kind, paired over tasks; whiskers show 95% bootstrap intervals. The top two pairs are successive generations of one family, while the bottom pair is a step up in tier within one generation at about five times the cost. (b) Picture-structure score per agent and input kind, which matches each shot's depth, normals and segmentation to the reference frames; filled dots mark the four agents with the highest overall score. (c) Error rates of the best and the weakest agent, the other nine in gray; ρ is the Spearman correlation with the overall score, and Appendix C.2 defines each rate.

a paired bootstrap over tasks, while the overall ranking holds across 18 scoring variants in Appendix B.4, at Kendall $\tau \geq 0.85$ with GPT-6 Astra always first. The top four lead the rest by at least 3.9 overall points. GPT-6 Astra leads Rule-based Scene Check on every input kind except photos, where Opus 5.5 leads, as Appendix C.2 details. No agent comes close to solving the benchmark. On only 35 of the 242 tasks does any agent reach a Rule-based Scene Check score of 90. The remaining gap is shared, not divided: the best of all 11 agents on every task raises the overall score only to 80.6, and the top four alone already reach 80.5.

Cost and score come apart. Per-task cost varies 45-fold, from $0.73 for GLM 5.3 Flash to $32.59 for GPT-6 Astra. Five agents sit on the cost–score frontier in Fig. 7. GPT-6.1 Sol reaches 98% of the leader's score at 21% of the cost, and DeepSeek V4.1 Flash 90% at 6%. Paying more does not buy rank: Kimi K3 is second in cost but seventh in score.

4.3 Observations and implications

Three observations show what separates the agents, where they all fail and why they do not correct themselves, and we analyze each and give its implications.

Observation 1: Multimodal understanding separates the agents, especially in understanding space and camera motion. The more an input relies on vision, the wider the gap between agents. In Tab. 2, the gap between the best and worst agent grows from 12.0 Rule-based Scene Check points on text-only scripts to 19.9 on photos. This gap comes mainly from errors in space and camera. On photos, picture structure and camera account for 63% of the gap between the top four and the other seven, entities and layout for only 9%. The top four score 66–71 on picture structure, the other seven 24–38, as Fig. 8b shows. The bottleneck is reading images and videos, not building the scene. Given a text description of the hidden GT facts on five video tasks, DeepSeek V4.1 Flash gains 14.0 points, as Appendix Tab. 5 reports, nearly the full 16.7-point gap between the best and the worst agent on videos. Model updates gain here too: in Fig. 8a, a newer model of the same family gains 11–14 points on photos but only 3–4 on scripts. A larger tier of the same generation, GPT-6 Astra over GPT-6.1 Sol, gains at most 1.7 points on any input kind. Writing code is not the bottleneck. The three weakest agents often call an API removed in Blender 5, as Fig. 9c shows, yet later rounds repair these calls. Within each agent, runs with a failed API call score within 0.1 points of runs without one. *Progress on this step thus depends on visual understanding of space and camera, whereas more knowledge of the Blender API saves rounds, not points.*

Observation 2: All agents master only basic camera work and staging, and fail at challenging requests. Appendix Fig. 14a shows that the errors of all 11 agents concentrate on

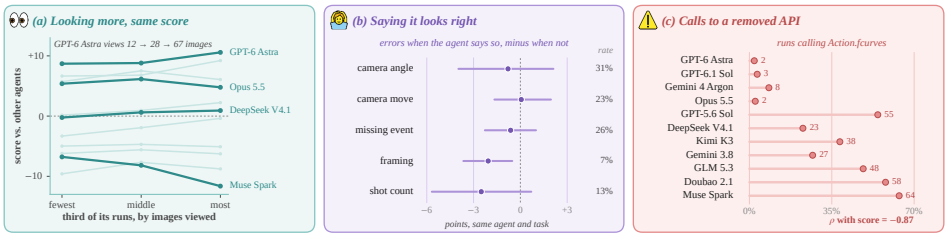


Figure 9. **How agents look, judge and code.** (**a**) Score relative to the other agents' task mean, by third of an agent's runs ranked by images viewed. (**b**) Error rate in runs where the agent says its render matches or looks right, minus runs where it does not, with agent and task held fixed; whiskers show 95% bootstrap intervals and *rate* is the mean error rate. (**c**) Share of runs calling `Action.fcurves`, removed in Blender 5. Rows sorted by overall score.

challenging requests, which they turn into the common choice. When the reference films from a low angle, 66% of shots come out at eye level rather than high. Only 4% of orbits around a subject stay orbits, and 47% become sideways moves. Half of the video and continuation runs have more shots than the reference, while only 4% have fewer. Objects mentioned only in prose go missing 35% of the time. The common version of each request, by contrast, mostly succeeds: eye-level shots go wrong 9% of the time, follow and push-in moves 5–8%, and listed objects go missing 11% of the time. Challenging requests are likely scarce in the data agents learn from, so agents fall back on what they know. Two observations fit this reading. The errors lean one way, toward the common choice, whereas a misreading would err both ways. They also barely vary across agents: in Fig. 8c, errors in reading the picture fall with the overall score, whereas every agent films 32–41% of shots at the wrong camera height. *Agents therefore need explicit checks for challenging requests, such as camera height and move, shot count and objects named only in prose.*

Observation 3: Agents inspect their own renders but fail to correct their errors. Every agent inspects its renders, viewing 13–40 images per run after its first Blender call, yet more looking does not help. In Fig. 9a, the third of runs that looks most views 3.3 times as many images as the third that looks least, yet scores only 0.8 points higher at the median once task difficulty is removed. Removing every image from DeepSeek V4.1 Flash's context on eight script tasks changes its score by only -0.8 points, as Appendix Tab. 5 reports. The reason is that agents cannot tell a wrong render from a right one. In 1,377 of 2,582 runs with comments after a render, the agent states that its render matches the reference or looks right. With agent and task held fixed, these runs get camera angles, camera moves and events wrong as often as the other runs, as Fig. 9b and Appendix C.2 show. Only framing and shot count, which show at a glance, are 2–3 points lower. *More renders and looking thus do not correct errors; agents need item-by-item verification of renders against the brief.*

5 Conclusion

We introduced VibePrevisBench to evaluate coding agents that build executable 4D scenes and render proxy videos from four kinds of input. Rule-based Scene Check evaluates the scene against hidden GT facts, while VLM-based Screen Check evaluates how clearly those facts read on screen. Analysis of the evaluation results and trajectories of 11 agents yields three observations. Visual understanding of space and camera separates agents. Challenging staging and filming requests remain difficult across capability levels. Agents inspect their renders, but their judgments do not reliably correspond to fewer errors. These observations point to better visual understanding, explicit checks for challenging requests, and verification of renders against the brief. VibePrevisBench makes progress on these capabilities measurable.

References

[1] Jianhong Bai, Menghan Xia, Xiao Fu, Xintao Wang, Lianrui Mu, Jinwen Cao, Zuozhu Liu, Haoji Hu, Xiang Bai, Pengfei Wan, and Di Zhang. Recammaster: Camera-controlled generative rendering from a single video. In *ICCV*, 2025.

[2] Hritik Bansal, Clark Peng, Yonatan Bitton, Roman Goldenberg, Aditya Grover, and Kai-Wei Chang. Videophy-2: A challenging action-centric physical commonsense evaluation in video generation. *arXiv preprint arXiv:2503.06800*, 2025.

[3] ByteDance Seed. Seedance 2.5: clay-render (greybox) video reference control. https://seed.bytedance.com/en/seedance2_5, 2026.

[4] Ziqi Cai, Taoyu Yang, Zheng Chang, Si Li, Han Jiang, Shuchen Weng, and Boxin Shi. Lighting-grounded video generation with renderer-based agent reasoning. *arXiv preprint arXiv:2604.07966*, 2026.

[5] Wenhao Chai, Enxin Song, Yilun Du, Chenlin Meng, Vashisht Madhavan, Omer Bar-Tal, Jenq-Neng Hwang, Saining Xie, and Christopher D. Manning. AuroraCap: Efficient, performant video detailed captioning and a new benchmark. In *ICLR*, 2025.

[6] Kaiwei Chen, Moran Li, Zhenye Gan, Yabiao Wang, Chengjie Wang, Kexin Zhang, Yong Liu, and Jiangning Zhang. PVBench: Quantifying physical plausibility in video generation with Object-Level physical measurements. *arXiv preprint arXiv:2610.26711*, 2026.

[7] Ye Chen, Xuanhong Chen, Yupeng Zhu, Liming Tan, Zhewen Wan, Yuxuan Xiong, Tielong Wang, Jinfan Liu, et al. World narrative model for highly controllable video generation: A paradigm shift from pixel sampling to physical world orchestration. *arXiv preprint arXiv:2606.31946*, 2026.

[8] Yiwen Chen, Guosheng Lin, and Chi Zhang. Code world model: Coding agent as world brain. *arXiv preprint arXiv:2608.25927*, 2026.

[9] Chong Cheng, Ran Zhang, Xinyuan Li, Alan Zhao, Hao Wang, and Xi Wang. When left is right? Camera-Motion understanding beyond screen. *arXiv preprint arXiv:2610.17242*, 2026.

[10] Muxi Diao, Wenbin Xu, Lele Yang, Xuyang Ma, Yuzhe Chen, Jiankang Chen, Kaiyu Jiang, Changyi Liu, Haoyu Zhang, Hongyi Fu, Jianxiong Wang, Tianke Zhang, Haonan Fan, Yingxin Li, Bingyi Li, Xu Wang, Xuanyu Zheng, Bin Wen, Tingting Gao, Han Li, Wentao Zhang, Weiran Xu, Kongming Liang, and Zhanyu Ma. VibeVideoBench: Evaluating multimodal agents on Code-Driven video generation. *arXiv preprint arXiv:2610.35864*, 2026.

[11] Bing Duan, Qiang Guo, Linpu Li, Zhijian Mao, Min Zhu, Zhirui Ren, Yiwei Yan, Xi Chu, et al. PACE: Precise AI cinematic expression. *arXiv preprint arXiv:2609.19853*, 2026.

[12] Haoyi Duan, Hong-Xing Yu, Sirui Chen, Li Fei-Fei, and Jiajun Wu. Worldscore: A unified evaluation benchmark for world generation. In *ICCV*, 2025.

[13] Xinyu Fang, Yuchi Wang, Weihao Yu, Rongyao Fang, Zhaohai Li, Zhixun Li, Tiancheng Gu, Xilin Wei, Yuhang Zang, Hongsheng Li, Dahua Lin, Guofeng Zhang, Shuai Bai, Zhibo Yang, and Dayiheng Liu. Blenderworld: Benchmarking visual agents for 3d creation from real-world references. *arXiv preprint arXiv:2610.23986*, 2026.

[14] Yusu Fang, Tiange Xiang, Tian Tan, Narayan Schuetz, Scott Delp, Li Fei-Fei, and Ehsan Adeli. HumanScore: Benchmarking human motions in generated videos. *arXiv preprint arXiv:2604.20157*, 2026.

[15] Zixun Fang, Yawen Shao, Kai Zhu, Jie Xiao, Shihan Chen, Yu Liu, Xueyang Fu, Yang Cao, Wei Zhai, and Zheng-Jun Zha. Code plans, diffusion renders: Open-Ended generative world modeling. *arXiv preprint arXiv:2609.26458*, 2026.

[16] 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 *CVPR*, 2025.

[17] Xiao Fu, Xian Liu, Xintao Wang, Sida Peng, Menghan Xia, Xiaoyu Shi, Ziyang Yuan, Pengfei Wan, Di Zhang, and Dahua Lin. 3dtrajmaster: Mastering 3d trajectory for multi-entity motion in video generation. In *ICLR*, 2025.

[18] Yipeng Gao, Lei Shu, Genzhi Ye, Xi Xiong, Ameesh Makadia, Meiqi Guo, Laurent Itti, and Jindong Chen. 3dcodebench: Benchmarking agentic procedural 3d modeling via code. *arXiv preprint arXiv:2606.01057*, 2026.

[19] Yunqi Gu, Ian Huang, Jihyeon Je, Guandao Yang, and Leonidas Guibas. Blendergym: Benchmarking foundational model systems for graphics editing. In *CVPR*, 2025.

[20] Zekai Gu, Rui Yan, Jiahao Lu, Peng Li, Zhiyang Dou, Chenyang Si, Zhen Dong, Qifeng Liu, Cheng Lin, Ziwei Liu, Wenping Wang, and Yuan Liu. Diffusion as shader: 3d-aware video diffusion for versatile video generation control. In *SIGGRAPH*, 2025.

[21] Hao He, Yinghao Xu, Yuwei Guo, Gordon Wetzstein, Bo Dai, Hongsheng Li, and Ceyuan Yang. Cameractrl: Enabling camera control for text-to-video generation. In *ICLR*, 2025.

[22] Liu He, Yizhi Song, Hejun Huang, Pinxin Liu, Yunlong Tang, Daniel Aliaga, and Xin Zhou. Kubrick: Multimodal agent collaborations for synthetic video generation. In *CVPR Workshops*, 2025.

[23] Erzhen Hu, Frederik Brudy, David Ledo, George Fitzmaurice, and Fraser Anderson. Previzwhiz: Combining rough 3d scenes and 2d video to guide generative video previsualization. In *CHI*, 2026.

[24] Ziniu Hu, Ahmet Iscen, Aashi Jain, Thomas Kipf, Yisong Yue, David A. Ross, Cordelia Schmid, and Alireza Fathi. Scenecraft: An llm agent for synthesizing 3d scene as blender code. In *ICML*, 2024.

[25] Hsiang-Wei Huang, Junbin Lu, Kuang-Ming Chen, Jianxu Shangguan, Cheng-Yen Yang, and Jenq-Neng Hwang. CaMo: Camera motion grounded evaluation and training for vision-language models. *arXiv preprint arXiv:2605.20165*, 2026.

[26] Ian Huang, Guandao Yang, and Leonidas Guibas. Blenderalchemy: Editing 3d graphics with vision-language models. In *ECCV*, 2024.

[27] Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. Vbench: Comprehensive benchmark suite for video generative models. In *CVPR*, 2024.

[28] Zheng-Hui Huang, Guixu Lin, Yu-Ju Tsai, Jian-Kai Zhu, Fengbo Lan, Yu-Lun Liu, Yung-Yu Chuang, Kaipeng Zhang, and Zhixiang Wang. PROWBench: Do video models render what the program specifies? *arXiv preprint arXiv:2610.02205*, 2026.

[29] Swarnim Jain and Shangzhe Wu. RigidBench: Evaluating Rigid-Body physics in video generation models. *arXiv preprint arXiv:2608.15555*, 2026.

[30] Zeyinzi Jiang, Zhen Han, Chaojie Mao, Jingfeng Zhang, Yulin Pan, and Yu Liu. Vace: All-in-one video creation and editing. In *ICCV*, 2025.

[31] Kuaishou Kling Team. Kling 4.0: depth-map and greybox video references. https://kling.ai/blog/kling-4-vs-3, 2026.

[32] Bingliang Li, Zhenhong Sun, Jiaming Bian, Yuehao Wu, Yifu Wang, Hongdong Li, Yatao Bian, Huadong Mo, and Daoyi Dong. Storyblender: Inter-shot consistent and editable 3d storyboard with spatial-temporal dynamics. *arXiv preprint arXiv:2604.03315*, 2026.

[33] Dacheng Li, Yunhao Fang, Yukang Chen, Shuo Yang, Shiyi Cao, Justin Wong, Michael Luo, Xiaolong Wang, Hongxu Yin, Joseph E. Gonzalez, Ion Stoica, Song Han, and Yao Lu. Worldmodelbench: Judging video generation models as world models. *arXiv preprint arXiv:2502.20694*, 2025. arXiv preprint (venue ?).

[34] Haodong Li, Tianfei Ren, Xiaoxiao Ma, Chunmei Qing, Zhen Fang, Sipeng He, Ziyu Guo, Haoyu Wu, Juanxi Tian, Yihang Zou, Ruichuan An, Dongzhi Jiang, Boxue Yang, Ji Xie, Xu Huang, Wenhao Yan, Jialv Zou, Zhengrong Yue, Yaxin Luo, Xiaotong Li, Yuzhu Wang, Junyan Ye, Jinjing Zhao, Zehui Chen, Lin Chen, Renye Yan, Feng Zhao, and Pheng-Ann Heng. VideoCoCo: Code-as-CoT for physically-consistent video generation via an agentic dual-engine system. *arXiv preprint arXiv:2607.27380*, 2026.

[35] Qianru Li, Xuyang Chen, Erkin Türköz, Lu Liu, Xuqin Wang, Liqiu Meng, Tao Wu, and Yanfeng Zhang. CinemaTraj: Composing atomic camera trajectories for 3d scenes with LLM agents. *arXiv preprint arXiv:2607.26910*, 2026.

[36] Zhiyang Liang, Jiawei Zhang, Guocheng Qian, Dongdong Chen, and Ziyu Wan. Code video model, 2026. Technical report.

[37] Zhiqiu Lin, Siyuan Cen, Daniel Jiang, Jay Karhade, Hewei Wang, Chancharik Mitra, Tiffany Ling, Yuhan Huang, Sifan Liu, Mingyu Chen, Rushikesh Zawar, Xue Bai, Yilun Du, Chuang Gan, and Deva Ramanan. Towards understanding camera motions in any video. In *NeurIPS*, 2025.

[38] Junze Liu, Kun Qian, Florian Dubost, Kai Zhong, Arvind Srinivasan, Nan Chen, Anping Wang, Sam Zhang, Alejandro Mottini, Qingjun Cui, and Tian Wang. 3d primitives are a spatial language for VLMs. *arXiv preprint arXiv:2605.12586*, 2026.

[39] Xuyang Liu, Shihao Wang, Xinyu Wei, Minghan Li, Zeqing Wang, Zuhao Yang, Zichen Wen, and Lei Zhang. Omnicompass: A holistic benchmark for omni-modal video grounding and understanding. *arXiv preprint arXiv:2610.18829*, 2026.

[40] Fanqing Meng, Jiaqi Liao, Xinyu Tan, Wenqi Shao, Quanfeng Lu, Kaipeng Zhang, Yu Cheng, Dianqi Li, Yu Qiao, and Ping Luo. Towards world simulator: Crafting physical commonsense-based benchmark for video generation. In *ICML*, 2025.

[41] Jiahao Meng, Yue Tan, Qi Xu, Haochen Wang, Zhongwei Ren, Weisong Liu, Yuhao Wang, Renrui Zhang, Xiangtai Li, Haodong Duan, Yunhai Tong, and Ming-Hsuan Yang. VideoZeroBench: Probing the limits of video MLLMs with Spatio-Temporal evidence verification. *arXiv preprint arXiv:2604.01569*, 2026.

[42] Thomas Muender, Thomas Fröhlich, and Rainer Malaka. Empowering creative people: Virtual reality for previsualization. In *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–6. ACM, 2018.

[43] Shufeng Nan, Mengtian Li, Sixiao Zheng, Yuwei Lu, Han Zhang, and Yanwei Fu. Mind-of-Director: Multi-modal Agent-Driven film previsualization via collaborative Decision-Making. *arXiv preprint arXiv:2603.14790*, 2026.

[44] NVIDIA, Hassan Abu Alhaija, Jose Alvarez, Maciej Bala, Tiffany Cai, Tianshi Cao, Liz Cha, Joshua Chen, Mike Chen, Francesco Ferroni, Sanja Fidler, Dieter Fox, Yunhao Ge, Jinwei Gu, Ali Hassani, Michael Isaev, Pooya Jannaty, Shiyi Lan, Tobias Lasser, Huan Ling, Ming-Yu Liu, Xian Liu, Yifan Lu, Alice Luo, Qianli Ma, Hanzi Mao, Fabio Ramos, Xuanchi Ren, Tianchang Shen, Xinglong Sun, Shitao Tang, Ting-Chun Wang, Jay Wu, Jiashu Xu, Stella Xu, Kevin Xie, Yuchong Ye, Xiaodong Yang, Xiaohui Zeng, and Yu Zeng. Cosmos-transfer1: Conditional world generation with adaptive multimodal control. *arXiv preprint arXiv:2503.14492*, 2025.

[45] Dicong Qiu, Zhiyuan Xu, Yaosheng Liu, Feng Han, and Bo Ye. RenderBench: Benchmarking Render-to-Real video transfer with reconstructed digital twins. *arXiv preprint arXiv:2610.08684*, 2026.

[46] Xuanchi Ren, Tianchang Shen, Jiahui Huang, Huan Ling, Yifan Lu, Merlin Nimier-David, Thomas Müller, Alexander Keller, Sanja Fidler, and Jun Gao. Gen3c: 3d-informed world-consistent video generation with precise camera control. In *CVPR*, 2025.

[47] Ruihong Shen, Žiga Kovačič, Peter Kulits, Xingrui Wang, Zizhang Li, Joshua B. Tenenbaum, Alan Yuille, Jieneng Chen, and Jiajun Wu. 4dcodebench: Benchmarking agents on inverse graphics of dynamic scenes. *arXiv preprint arXiv:2610.03715*, 2026.

[48] Haoyuan Shi, Yunxin Li, Nanhao Deng, Zhenran Xu, Xinyu Chen, Longyue Wang, Baotian Hu, and Min Zhang. Msvbench: Towards human-level evaluation of multi-shot video generation. *arXiv preprint arXiv:2602.23969*, 2026.

[49] Chunyi Sun, Junlin Han, Weijian Deng, Xinlong Wang, Zishan Qin, and Stephen Gould. 3d-GPT: Procedural 3d modeling with large language models. *arXiv preprint arXiv:2310.12945*, 2023.

[50] Kaiyue Sun, Kaiyi Huang, Xian Liu, Yue Wu, Zihan Xu, Zhenguo Li, and Xihui Liu. T2v-compbench: A comprehensive benchmark for compositional text-to-video generation. In *CVPR*, 2025.

[51] Jinzhou Tang, Zijun Zhang, Jing Yang, Yuchen Yan, Kun Zhou, Lingjun Mao, Ruobing Han, Jinglin Cao, Wenpeng Xu, Lukun He, Minghao Fu, Fan Feng, and Biwei Huang. Video2World: Benchmarking coding agents for interactive world modeling from embodied videos. *arXiv preprint arXiv:2610.04432*, 2026.

[52] Yolo Y. Tang, Daiki Shimada, Jiayue Meng, Jing Bi, Pinxin Liu, Yicheng Wang, Yunzhong Xiao, Zhangyun Tan, Zeliang Zhang, Chao Huang, Susan Liang, Qianxiang Shen, Luchuan Song, Ali Vosoughi, Mingqian Feng, Melika Filvantorkaman, and Chenliang Xu. Bvb: Benchmarking agentic video understanding via programmatic reconstruction in blender. *arXiv preprint arXiv:2609.15478*, 2026.

[53] Jiawei Wang, Liping Yuan, Yuchen Zhang, and Haomiao Sun. Tarsier: Recipes for training and evaluating large video description models. *arXiv preprint arXiv:2407.00634*, 2024.

[54] Jialei Wang, Zehan Wang, XiaoYang Wu, Ziang Zhang, Siyu Chen, Jiaer Huang, Yujia Zhang, Tianyu Pang, and Hengshuang Zhao. CineRecon: Benchmarking frontier models for agentic cinematic reconstruction. *arXiv preprint arXiv:2610.16577*, 2026.

[55] Qinghe Wang, Yawen Luo, Xiaoyu Shi, Xu Jia, Huchuan Lu, Tianfan Xue, Xintao Wang, Pengfei Wan, Di Zhang, and Kun Gai. Cinemaster: A 3d-aware and controllable framework for cinematic text-to-video generation. In *SIGGRAPH*, 2025.

[56] Shengyi Wang, Niantong Li, Guangzheng Hu, Hong Qi, Fei Ding, Weixu Qiao, Jinlin Wang, Xiaotong Lv, et al. FilmBench: A film-grade benchmark for cinematic video generation. *arXiv preprint arXiv:2607.24241*, 2026.

[57] Yuji Wang, Yuheng Chen, Teng Hu, Ran Yi, Yijia Hong, Han Feng, Weijian Cao, Chengjie Wang, Lizhuang Ma, and Jiangning Zhang. PersonaShot: Benchmarking Person-Centric narrative continuity in Multi-Shot video generation. *arXiv preprint arXiv:2608.16717*, 2026.

[58] Zhouxia Wang, Ziyang Yuan, Xintao Wang, Tianshui Chen, Menghan Xia, Ping Luo, and Ying Shan. Motionctrl: A unified and flexible motion controller for video generation. In *SIGGRAPH*, 2024.

[59] Zeqing Wang, Xinyu Wei, Bairui Li, Zhen Guo, Jinrui Zhang, Hongyang Wei, Keze Wang, and Lei Zhang. Videoverse: Does your t2v generator have world model capability to synthesize videos? *arXiv preprint arXiv:2510.08398*, 2025. arXiv preprint (venue ?).

[60] Hock Hian Wong. Previsualization: Assisting filmmakers in realizing their vision. In *SIGGRAPH Asia 2012 Courses*, pages 1–20. ACM, 2012.

[61] Zhenran Xu, Longyue Wang, Jifang Wang, Zhouyi Li, Senbao Shi, Xue Yang, Yiyu Wang, Baotian Hu, Jun Yu, and Min Zhang. Filmagent: A multi-agent framework for end-to-end film automation in virtual 3d spaces. *arXiv preprint arXiv:2501.12909*, 2025.

[62] Jihan Yang, Shusheng Yang, Anjali W. Gupta, Rilyn Han, Li Fei-Fei, and Saining Xie. Thinking in space: How multimodal large language models see, remember, and recall spaces. In *CVPR*, 2025.

[63] Zhangsihao Yang and Mengyi Shan. CoaG: Cylinders on a grid for coarse 3D layout control in video generation. *arXiv preprint arXiv:2609.24208*, 2026.

[64] Linli Yao, Yuancheng Wei, Yaojie Zhang, Lei Li, Xinlong Chen, Feifan Song, Ziyue Wang, Kun Ouyang, Yuanxin Liu, Lingpeng Kong, Qi Liu, Pengfei Wan, Kun Gai, Yuanxing Zhang, and Xu Sun. TimeChat-Captioner: Scripting Multi-Scene videos with Time-Aware and structural Audio-Visual captions. *arXiv preprint arXiv:2602.08711*, 2026.

[65] Xiaokang Ye, Siddhant Hitesh Mantri, Zimeng Chen, Edward Zhang, Zhaoxu Zheng, Yuanheng Li, Yizhao Chen, Tianyang Huang, and Lianhui Qin. Code4Scene: Benchmarking coding agents for constructing and editing 3D scenes. *arXiv preprint arXiv:2609.36777*, 2026.

[66] Shaofeng Yin, Jiaxin Ge, Zora Zhiruo Wang, Chenyang Wang, Xiuyu Li, Michael J. Black, Trevor Darrell, Angjoo Kanazawa, and Haiwen Feng. Vision-as-inverse-graphics agent via interleaved multimodal reasoning. *arXiv preprint arXiv:2601.11109*, 2026.

[67] Yuyang Yin, Zixiang Li, Longxuan Deng, Hongkai Li, Shifang Zhao, Junnan Liu, Weirong Huang, Mengyu Wang, Tianxiao Fu, Yikai Wang, Peng-Shuai Wang, Xiaojie Jin, Yao Zhao, and Yunchao Wei. Stateflow: Building, evolving, and accessing 3d world states for previsualization. *arXiv preprint arXiv:2608.12314*, 2026.

[68] Yiming Zhang, Jiarui Guo, Haojia Wei, Yifei Liu, Yifan Yang, Hongjie Zhang, Xue Yang, and Zhihang Zhong. FilmFlow: Agentic system for 3D virtual film cinematography. *arXiv preprint arXiv:2610.30753*, 2026.

[69] Yifei Zhao, Xiangxin Zhou, Wenhao Yang, Jiaqi Tang, Pu Jian, Huanjin Yao, Jiarui Yao, Haowei Lin, Chunchao Guo, Zhuo Chen, Wenkai Lyu, Jianzhu Ma, Xueqian Wang, and Wenxi Zhu. SceneActBench: Can agents act on the 3D scenes they see? *arXiv preprint arXiv:2607.22393*, 2026.

[70] Dian Zheng, Ziqi Huang, Hongbo Liu, Kai Zou, Yinan He, Fan Zhang, Lulu Gu, Yuanhan Zhang, Jingwen He, Wei-Shi Zheng, Yu Qiao, and Ziwei Liu. Vbench-2.0: Advancing video generation benchmark suite for intrinsic faithfulness. *arXiv preprint arXiv:2503.21755*, 2025.

[71] Zanwei Zhou, Jiazhong Cen, Jiemin Fang, Yumeng He, Chen Yang, Sikuang Li, Fanpeng Meng, Zhikuan Bao, Wei Shen, and Qi Tian. Proxyup: Training-free proxy-conditioned video generation for controllable dynamics. *arXiv preprint arXiv:2607.03732*, 2026.

[72] Yingshuang Zou, Xiaoyu Shi, Yawen Luo, Ziang Li, Qinghe Wang, Mengfei Li, Chengfeng Zhao, Xintao Wang, Pengfei Wan, Kun Gai, and Yuan Liu. Mise-en-Video: Fine-Grained 3D control for Multi-Shot video generation. *arXiv preprint arXiv:2610.26801*, 2026.

Appendix

A Benchmark details

A.1 Task construction

Coverage first. Two frozen taxonomies tag every task. One covers what a scene holds: places, spaces, layout relations, subjects, actions, camera and editing, and genre, in 67 categories. The other covers how a task is asked: input kind, writing style, information level, organization and length. Each tag and each kind of place has a minimum number of tasks, and the gaps became slots for new tasks, routed to three lanes.

Scripts. Each slot becomes a card fixing genre, writing style, length, places, people, actions and a story seed. Scripts come in seven writing styles, from screenplays and shot lists to ad briefs and novel-like prose, at about 600 to 1,200 words. The hidden GT facts are written before the script: a model (GPT-6.1 Sol) first codes the scene, then writes the script, then the claims, each citing the sentence it comes from and checked to hold in the coded scene. Automatic gates check every task's scoring and output specification. Two blind readers from the Claude and DeepSeek families, who see only the script, try to recover every claim; a claim both misread is reworded and re-read up to twice, then dropped if still misread.

Videos and continuations. We collected candidate clips ourselves. The screening keeps clips of at least 5 s and 2 fps, with a short edge of at least 480 pixels for local clips and 720 for web videos, and drops clips whose adjacent frames are more than 85% identical. Local clips are cut into windows of 25 to 50 s, a seeded eighth also into 15 s windows, and web videos into 25–45 s video windows and 12 s continuation windows; each source keeps at most three or four windows that overlap by at most 60%. Windows are ranked by subject presence, the number of moving people or vehicles and its changes, motion beyond the camera's own, camera motion, cuts, person size and duration, and flagged for letterboxing, burned-in text, repeated frames and darkness. We then pre-screened the windows by model ratings of contact sheets (Gemini 4, Claude Sonnet 5.5), and tagged the survivors by place, genre, people, shots and moving vehicles. Human reviewers decided whether a clip becomes a video task, a continuation or neither. Quotas then capped over-represented content such as vehicles, and borders and logos were cropped. A video brief gives an id and a description for each place and subject, and the output must start where the clip starts and cut where it cuts. A continuation brief gives the clip's opening frame and an account of the next 6 to 20 s, mostly following the clip. Figure 10 shows how one clip's hidden GT facts are built.

Photos. We drew on scan datasets with measured 3D boxes and camera poses, indoors and in street scenes, and used each scene once. Boxes from the hidden GT facts were drawn back onto every photo and checked first by a model and then by a human; candidates whose boxes did not match the photo were rejected. The output must show each photo's view for 12 frames and then sweep the place for 8 s. One photo task instead uses five keyframes of a real clip, and its hidden GT facts are estimated as for a video.

Pruning. All 11 agents ran the pool, and we dropped tasks that did not discriminate between agents: low item-rest correlation or little spread across agents, never a single agent's score. Finally we removed tasks with source or licensing problems or an unscorable run.

Model assistance and fairness. Several of the evaluated models help build tasks. GPT-6.1 Sol writes the script specs and scripts; the blind readers come from other families, Claude and DeepSeek; Claude Sonnet 5.5, of the same family as Opus 5.5, pre-screens clips; and Gemini 3.8 and Gemini 4 register and observe the subjects behind the hidden GT facts of videos and continuations, while Gemini 4 also rates web clips. Three safeguards limit their influence. Agents are scored by rule checks on facts, never by similarity to a model's own output; the hidden GT facts are never edited to make a reader pass; and every brief

and its facts are reviewed and corrected by hand. Ranked by Rule-based Scene Check, the agents show no strong self-preference. GPT-6.1 Sol is second on scripts and over all tasks; DeepSeek V4.1 Flash is seventh on scripts and over all tasks; and Opus 5.5 is third over all tasks and fourth on videos. The Gemini models rank somewhat higher on the media tasks whose hidden GT facts they helped estimate: Gemini 4 Argon is fourth over all tasks but third on videos and second on continuations, and Gemini 3.8 Flash is eighth over all tasks but sixth on videos. A small preference cannot be ruled out; the judge's own-family bias is measured separately in Appendix B.4.

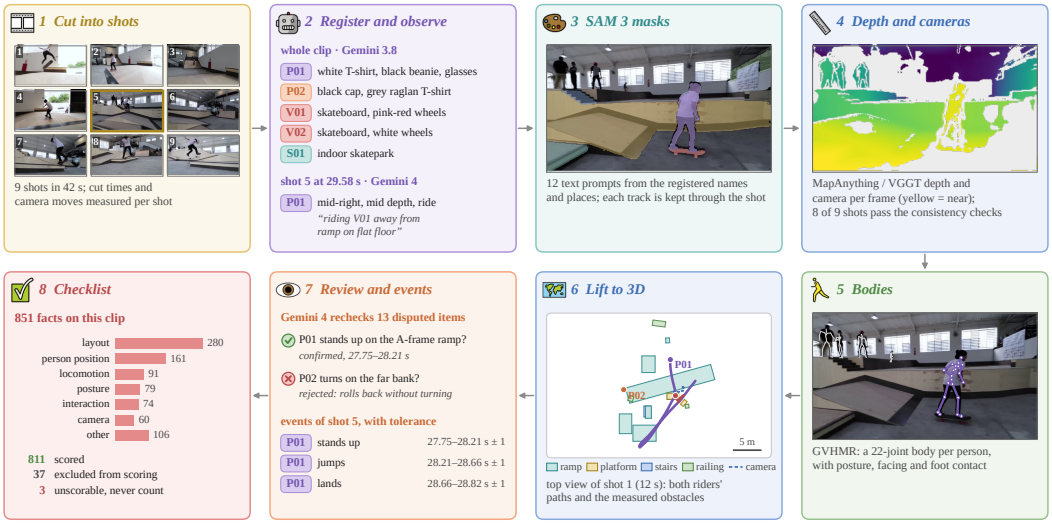


Figure 10. **Hidden ground truth facts for a video task.** The eight pipeline steps on one clip: shots, registration and per-shot observation, masks, depth and cameras, bodies, lifting to 3D, review and events, and the final checklist of facts with their tolerance windows.

A.2 Task examples

Figure 11 shows one task of each kind, and their prompts are excerpted below.

Script (excerpt). *Suitcase Chase Through the Corner Shop. A courier is chased through a shop into a back alley, where a reversing van traps his accomplice… S02 Shop: north of the street's east part. A west shelf and an east shelf run north–south either side of the aisle from the front door to the back door… P01 person: courier holding O01, west of the planter, facing east. P02 person: pursuer, behind P01, facing east… 0–5 s: P01 runs east with O01, P02 chasing. 5–10.8 s: P01 enters the shop by the front door at 9.5 s; P02 follows at 10.8 s… Shots (lens: wide-angle 24 mm, moderately wide 28 mm, normal 35 mm, slightly long 50 mm)… 5–8 s, hard cut: under the arcade's east end; eye level, normal, facing west; pans right to the door…*

Photos (prompt). *Rebuild the place shown in the attached photo as one static 3D proxy blockout, with one camera. Build the place and everything in it at real-world size, and place the camera so that its view matches the photo: position, direction and lens. Output: 12 frames at 24 fps, 640×426… Place: S01 Informal learning lounge. Tag every piece of the place (walls, floor, furniture, fixtures) by class…*

Continuation (excerpt). *Continue the attached photo as a proxy previs. Frame 1 must reproduce the photo: the camera's position, direction and lens, and every visible subject and object where it is in the photo… Output: 336 frames at 24 fps (14.00 s)… P01 person: the seated woman beside the bed… P02 person: the standing man to her right… 0.0 s: the close eye-level side view of P01 and P02 pulls back in a dolly move. From 0.3 to 2.2 seconds, P01 stands up from the bed… 4.0 s: Cut to a medium eye-level front view of P01 with P02. The camera orbits toward her side… From 6.4 to 7.1 seconds, P01 turns back toward P02…*

Reference video (prompt). *Rebuild the attached reference video as a proxy previs. Match what the video shows: the places and what stands where, which way things face, how every listed subject moves and what it does when, and the camera of every shot (where it stands, how it frames and moves), with the cuts and transitions*

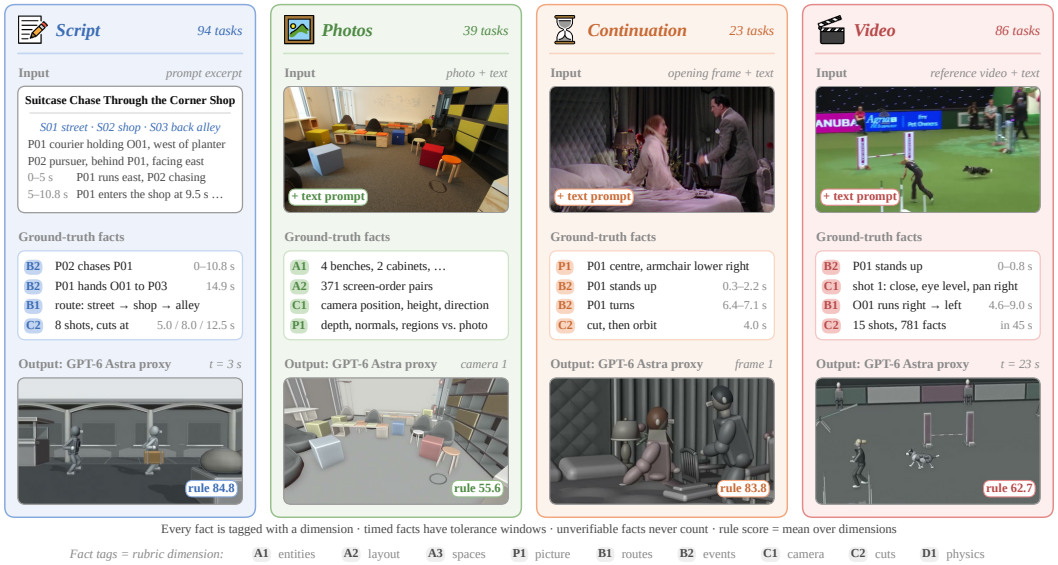


Figure 11. **The four kinds of input.** One task of each kind. *Top:* what the agent receives (a script excerpt, a photo, an opening frame, a reference video); photo, continuation and video tasks also come with a text brief. *Middle:* some of the hidden GT facts for that task, each tagged with its rubric dimension. *Bottom:* a frame of the proxy video delivered by GPT-6 Astra, with its Rule-based Scene Check score. Timed facts carry tolerance windows, and facts the ground truth cannot verify never count.

at the same times as in the video. Start at the video's first frame. Reference video: 45.02 s. Output: 1080 frames at 24 fps (45.00 s), 640×360. Places: S01 indoor agility arena. Subjects: P01 person: blonde woman with her hair tied in a ponytail… O01 dog: medium-sized dog with grey, black, and white fur…

A.3 Task statistics

Scripts come in seven writing styles with 9 to 24 scripts each, and photo tasks give a single photo in 14 cases and three to five photos in the other 25. Outputs run about 40 s for scripts and videos, 12 s for continuations and 10 s for photo tasks. A script stages a median of four people in three places over seven shots, and a video four people over 12 shots, with up to 15 people and 38 shots. Scripts and videos carry about 500 and 420 checks at the median, while photo tasks and continuations carry fewer than 100. Beyond events, camera holds 11% of the checks, routes 8% and cuts 7%, layout, picture structure and entities 4 to 5% each, and spaces and physical validity 2% each.

B Evaluation details

B.1 Rule-based Scene Check dimensions

Table 3. **Rule-based Scene Check dimensions.** The Rule-based Scene Check score is the plain mean of the dimensions a task exercises; P1 applies only to tasks with reference frames (photos, continuations and videos).

Dim.	What is checked against the hidden GT facts
A1 Entities	every listed subject and prop exists, with the right class and count
A2 Layout	relative positions: sides, distances, depth and screen order, against walls
A3 Spaces	places, their extents and how they connect
P1 Picture structure	per-shot depth, normals, segmentation, edges and facing vs. the reference frames
B1 Routes	where each subject is over time, the order of places visited, direction
B2 Events	whether each event happens, when, in which order, with which posture and gait
C1 Camera	per shot: position, height, facing, lens, subjects in frame, camera move
C2 Cuts	number of shots, cut times, transitions, shot order and duration
D1 Physical validity	no interpenetration, floating or sinking, teleporting, implausible sizes

B.2 Rule-based Scene Check scoring

Gates and replay. The three gates run 21 automatic checks: the code runs standalone; the video is rendered in full at the specified resolution, frame rate, duration and codec; and the scene reopens in Blender with the task's settings. Fourteen of the 2,662 runs fail a gate and score 0, two of them by Gemini 4 Argon. The replay dumps the tagged entities and places, their oriented boxes and parts, the cameras and the cuts.

Binding and alignment. Script facts are read in the agent's own frame, whose compass the brief fixes. For media tasks, each place is aligned by a similarity $T(x) = s\,R_\theta x + t$ fitted by weighted least squares to the anchored bodies, with $s \in [0.67, 1.5]$. A place with fewer than three anchors is scored by pairwise relations only.

Fact scores. The 84.4k scorable facts expand into 99.0k checks. A yes-or-no check scores 0 or 1, and most metric checks with error e score $\mathrm{clip}\big((z-e)/(z-f)\big)$, which is full within the registered tolerance f and zero beyond z.

Chance rates and physical validity. The chance rate of a block is $R_b = \frac{1}{200}\sum_{k=1}^{200} P_b(\pi_k S)$, where π_k randomly permutes the agent's static objects within each place. Camera checks, which no permutation moves, take $R_b = 0$. Physical validity is scaled by $\min(1, E/0.25)$, where E is the mean of layout, routes and events, so that a near-empty scene earns no physical-validity points.

B.3 VLM judge rubric

The judge (Gemini 4) watches the proxy video at 2 fps. Its prompt says that the video comes from a simplified previs, with simple shapes and mannequin-like figures in at most flat colors, so this look is not penalized. It never sees the agent's name or the object names.

Turn 1: blind recap. Without the brief, the judge describes only what it sees: in time order, who is where and what each person does, with people referred to by their look and objects by their shape ("unidentified shape" or "unclear movement" otherwise); what kind of place it is; and every object it can identify from its shape. For photo tasks it also watches an evaluation-side orbit of the delivered scene (a 360° look from the room center, then a top-down orbit with the ceiling hidden) and lists places and objects only.

Turn 2: verdicts. The judge then receives the brief (and, for video tasks, the reference video) and checklists generated from the hidden GT facts; items the rules already failed are not asked. It is told that the automatic checks verify presence, positions, layout, camera placement, timing, shot count and physical validity, and that materials, lighting, realism, aesthetics and composition are not judged. Every entry gets *yes*, *partial*, *no* or *n/a* (entity or shot absent, already penalized) with timestamps, and the lower verdict when in doubt.

V1 Recognition of shapes and places (key entities, places). *yes* only if the blind recap already named the category or kind of place; *partial* if it was not named blind but a shape at a plausible spot fits once the brief is known; *no* otherwise.

V2 Action readability (events). Would a viewer who has not read the brief describe the footage as that action? *yes*: action and contact are clear; *partial*: the pose changes but the action or contact is unclear; *no*: miming, standing still or another action. Events whose moment is not visible in V3 are left out of V2.

V3 Visibility at key moments. At each judged event, and in each shot without events, are the people and objects involved clearly visible? *partial*: small, partly blocked or partly out of frame; *no*: blocked, out of frame, too small, a blank or wall-only frame, the camera inside geometry, flicker or leftover helper objects.

V4 Story beats. Does the judge's own blind recap contain each story beat (*yes*: its gist; *partial*: part of it)?

With *yes* $= 1$, *partial* $= \frac{1}{2}$ and *no* $= 0$, each dimension averages its entries (n/a left out), and VLM-based Screen Check is 100 times the mean over the applicable dimensions, which on photo tasks are V1 alone.

B.4 Is the evaluation trustworthy?

Determinism and direction. Rule-based Scene Check is deterministic: each of the 2,786 deliveries that were scored more than once during development received an identical score every time. On 12 high-scoring video deliveries, injected defects lower the Rule-based Scene Check score monotonically and in the intended dimensions: delaying all action by 4 s costs 9.8 points and keeping one person costs 12.1. A 90° camera yaw costs 3.6 points when only the exported cameras are turned, and 21.0 when the ray-cast picture buffers are also re-projected to the turned camera. The physical-validity check misses floating people: lifting them by 1 m costs only 0.8.

Judges. Re-judged with GPT-6.1 Sol and Qwen3.8 Omni Flash, the agents' VLM-based Screen Check scores correlate with Gemini 4's at Spearman 0.94–0.96, though only at 0.73–0.76 per run; Gemini 4 favors its own family by 1.3 points without changing any rank, and the overall ranking is identical whichever judge is used (Kendall $\tau = 1.00$). Table 4 lists every pair and the agreement with humans.

Table 4. **Agreement between judges and with humans.** *Agent level*: correlation of the 11 agents' mean VLM-based Screen Check scores; *run level*: Spearman ρ over single runs. Failed runs and the runs GPT-6.1 Sol declined are excluded. Human rows correlate each agent's win rate in the human study with its mean score from that judge. *Mean*: each judge's mean score over all 2,662 runs, failed runs counting 0.

			Agent level		Run level
Pair		Runs	Spearman ρ	Pearson r	Spearman ρ
Gemini 4	GPT-6.1 Sol	2,628	0.96	0.98	0.76
Gemini 4	Qwen3.8 Omni Flash	2,648	0.94	0.97	0.73
GPT-6.1 Sol	Qwen3.8 Omni Flash	2,628	0.96	0.99	0.79
Human	Gemini 4	242	0.92	0.94	–
Human	GPT-6.1 Sol	242	0.92	0.91	–
Human	Qwen3.8 Omni Flash	242	0.90	0.93	–
Mean: Gemini 4 75.3, GPT-6.1 Sol 61.1, Qwen3.8 Omni Flash 61.0					

Ranking robustness. Agent means are compared by a paired bootstrap over tasks with 10,000 resamples. The 95% interval of the score difference includes zero for five of the ten adjacent pairs: GPT-6.1 Sol and Gemini 4 Argon, Gemini 4 Argon and Opus 5.5, GPT-5.6 Sol and DeepSeek V4.1 Flash, Gemini 3.8 Flash and GLM 5.3 Flash, and Doubao Seed 2.1 Pro and Muse Spark 1.3. The 18 scoring variants combine nine scores, namely the overall score with each of the three judges or their mean, the geometric mean of rules and judge, the rules alone and each judge alone, with two weightings, one per task and one per input kind. Their rankings agree with the main one at Kendall τ of 0.855 or more, and GPT-6 Astra is first under every one of them.

Humans. On 242 pairs of proxy videos, one per task, an annotator picked the one easier to understand; each agent's win rate correlates with its mean score from each judge at Spearman 0.90–0.92.

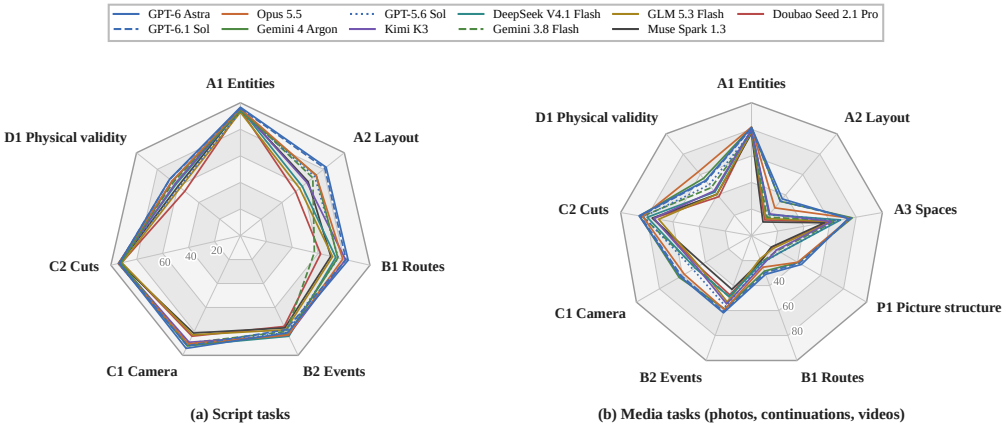
C Additional results

C.1 Ablations and capability profiles

Figure 12 profiles all 11 agents across the nine Rule-based Scene Check dimensions, and Fig. 13 gives the exact values, separately for script and media tasks.

Table 5. **Ablations** on DeepSeek V4.1 Flash. Each row reruns the agent with one change and reports the paired per-task difference to the main run (mean ± s.d.). *Rerun*: the same tasks run again unchanged. *Oracle caption*: video tasks with the video replaced by a text description. *No images*: every image returned to the agent is replaced by a text placeholder. $|\Delta|$: mean absolute change of the overall score per task. Cost: agent $ per task, reference → ablation. †The caption is generated from the hidden GT facts (events, cuts, camera), so this row bounds what a perfect perception step would add.

Change	Tasks	n	ΔScene	ΔScreen	ΔOverall	$\lvert\Delta\rvert$	Cost
Rerun	Script	8	$-2.0_{\pm 5}$	$+2.5_{\pm 14}$	$+0.2_{\pm 7}$	5.5	0.57→0.70
Oracle caption†	Video	5	$+14.0_{\pm 8}$	$+17.2_{\pm 13}$	$+15.6_{\pm 6}$	15.6	1.94→0.47
No images	Script	8	$-0.1_{\pm 4}$	$-1.6_{\pm 16}$	$-0.8_{\pm 6}$	5.5	0.57→0.59



(a) Script tasks (b) Media tasks (photos, continuations, videos)

Figure 12. **Capability profiles of all 11 agents.** Rule-based Scene Check score per dimension (0–100, failed runs count 0) on (a) script tasks and (b) media tasks. Exact values per agent are in Fig. 13.

	A1 Entities [95.0]	A2 Layout [68.1]	B1 Routes [73.5]	B2 Events [80.5]	C1 Camera [89.0]	C2 Cuts [93.2]	D1 Physical validity [62.5]
GPT-6 Astra	**96.6**	**82.5**	**83.2**	81.2	**94.2**	93.6	**68.1**
GPT-6.1 Sol	95.8	81.2	81.6	78.6	92.0	93.6	68.0
Opus 5.5	95.5	72.8	79.0	83.0	90.4	**94.2**	65.9
Gemini 4 Argon	93.8	73.0	75.2	82.5	92.1	92.2	65.6
GPT-5.6 Sol	93.5	71.4	75.5	81.0	91.2	94.2	59.0
Kimi K3	95.9	64.0	81.1	83.2	89.1	93.1	62.8
DeepSeek V4.1 Flash	95.6	59.5	73.6	**84.0**	91.9	94.2	62.0
Gemini 3.8 Flash	95.3	69.7	57.1	80.0	89.7	94.0	64.5
GLM 5.3 Flash	92.8	57.0	71.1	78.6	82.9	91.6	57.9
Muse Spark 1.3	95.6	65.1	69.7	77.0	81.4	93.0	59.9
Doubao Seed 2.1 Pro	94.6	53.1	61.9	75.9	84.0	91.8	53.6

(a) Script tasks

	A1 Entities [79.5]	A2 Layout [23.4]	A3 Spaces [67.2]	P1 Picture structure [28.5]	B1 Routes [23.9]	B2 Events [54.0]	C1 Camera [52.1]	C2 Cuts [80.2]	D1 Physical validity [48.1]
GPT-6 Astra	81.4	**36.1**	73.9	**43.4**	**31.0**	**61.8**	61.8	**85.9**	54.0
GPT-6.1 Sol	80.9	33.9	75.8	41.6	29.6	60.9	60.8	85.0	53.4
Opus 5.5	**81.7**	27.3	75.5	40.1	25.3	59.2	58.3	83.2	**62.5**
Gemini 4 Argon	81.6	33.7	**77.0**	40.5	28.4	58.8	**62.9**	85.5	56.2
GPT-5.6 Sol	77.9	20.7	66.4	25.2	22.1	56.8	50.8	82.4	49.8
Kimi K3	79.5	21.3	61.1	21.7	23.8	54.7	47.5	74.7	43.9
DeepSeek V4.1 Flash	79.7	20.6	68.0	25.1	21.9	48.9	48.9	79.7	42.9
Gemini 3.8 Flash	77.9	18.2	64.7	22.5	21.4	49.0	48.3	83.5	47.2
GLM 5.3 Flash	78.1	16.9	61.2	19.0	18.6	53.2	45.7	70.9	40.6
Muse Spark 1.3	77.0	13.4	55.8	17.2	20.5	43.1	42.3	75.9	40.6
Doubao Seed 2.1 Pro	78.9	15.4	59.4	17.5	20.4	47.5	46.0	75.0	38.2

(b) Media tasks

Figure 13. **Rule-based Scene Check score per dimension** (failed runs count 0; column means in brackets; best per column in bold). (a) Script tasks. (b) Media tasks (photos, continuations, videos). Rows sorted by Rule-based Scene Check score over all tasks.

C.2 Where agents improve, and where they do not

This section details the evidence behind Figs. 8 and 14 and the observations of Sec. 4. Unless stated otherwise, gains use the Rule-based Scene Check score, intervals are 95% paired bootstrap intervals over tasks, and correlations are Spearman ρ over the 11 agents.

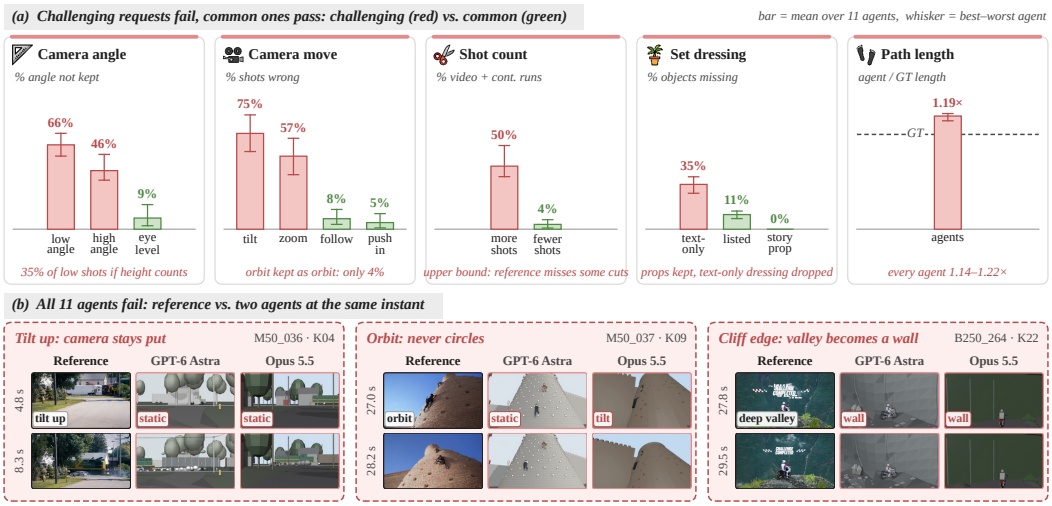


Figure 14. **Shared failure modes.** (a) Error rates on a challenging request (red) and a common one (green), averaged over the 11 agents; whiskers show the best and worst agent. *Camera angle* and *camera move*: shots whose angle or move type differs from the reference. *Shot count*: video and continuation runs with more or fewer shots than the reference, an upper bound since the reference misses some cuts. *Set dressing*: objects missing from the scene. *Path length*: median ratio of agent path to GT path, pooled over all input kinds. (b) Shots that the rules mark wrong for all 11 agents: the reference frame and two agents at the same instant; tags describe what each one shows.

Table 6. **Within-family Rule-based Scene Check gains** per input kind, paired over tasks, with 95% bootstrap intervals. Task counts are given in brackets.

Step	Script (94)	Photo (39)	Cont. (23)	Video (86)	All (242)
GPT-5.6 Sol → GPT-6.1 Sol	+3.6 [2.2, 4.9]	+11.1 [8.8, 13.5]	+5.1 [2.6, 7.6]	+6.9 [5.6, 8.2]	+6.1 [5.2, 7.0]
Gemini 3.8 Flash → Gemini 4 Argon	+3.4 [0.4, 5.8]	+14.1 [11.4, 17.1]	+7.1 [4.1, 10.5]	+9.7 [8.3, 11.0]	+7.7 [6.3, 9.0]
GPT-6.1 Sol → GPT-6 Astra	+1.2 [−0.2, 2.7]	+0.5 [−1.1, 2.1]	+1.7 [−0.2, 3.5]	+0.9 [−0.3, 2.0]	+1.0 [0.3, 1.8]

The spread on visual input is not a difficulty effect. Within each input kind, we split the tasks at the median of their mean Rule-based Scene Check score. On scripts, the best and worst agents are 12.3 points apart on the harder half and 11.8 on the easier half. The two halves give 18.3 and 22.6 on photos, 18.6 and 13.6 on continuations, and 19.7 and 15.3 on videos. A two-way decomposition splits the Rule-based Scene Check score into agent, task and residual. The agent explains 17% of the variance on scripts but 21% on continuations, 28% on photos and 29% on videos. The harder script half (mean 76.8) is still easier than the average video task (48.5), so the split controls difficulty within a kind, not across kinds. The spread is a Rule-based Scene Check result. On photos the judge asks only whether subjects and places are recognizable, where agents score 81–94, so photos have the smallest overall-score spread, 13.3.

Generational gains concentrate on visual input. Table 6 gives the gains behind Fig. 8a. In both families the largest gain on media tasks is in picture structure (P1 +16.4 and +17.9), followed by layout and camera (A2 +13.2 and +15.5; C1 +10.0 and +14.5). The step from GPT-6.1 Sol to GPT-6 Astra is above zero over all tasks but within no single input kind, and every dimension changes by −1.9 to +2.6. Only two family pairs are available, and the Gemini pair also moves from a Flash to an Argon tier. Under the Qwen3.8 Omni Flash judge, Gemini 4 Argon also gains 11.2 VLM-based Screen Check points on scripts, so the

concentration on visual input holds for Rule-based Scene Check.

Picture structure forms two tiers. Per task, the mean P1 of the top four agents exceeds that of the other seven on 39 of 39 photo tasks, by 38.3 points on average (95% CI 34.6–41.8). The same holds on 11 of 11 continuation tasks (25.0 points) and on 70 of 71 video tasks, where every agent stays at 12–26 and the gap is only 9.6. Photo tasks give no camera intrinsics, so agents must estimate position, direction and lens. Entities and layout barely separate the agents on photos (A1 7.6, A2 8.1). No agent exceeds 26 on layout: better agents match the view, while none rebuilds the room behind it. The top four are also the newest and most expensive agents, so the tiers do not isolate which capability crosses the threshold. Split by dimension as weighted in the score, the 15.0-point Rule-based Scene Check gap between the top four and the other seven on photos comes 45% from picture structure, 18% from camera, 14% from physical validity, 13% from spaces, 6% from layout and 3% from entities.

Reading the reference limits video tasks. Figure 15 splits the 86 video tasks by properties of the reference clip. Scores fall with clip length, shots, moving shots and places (Spearman -0.27 to -0.38), but hardly with the number of people or objects (-0.11 each). On shots, moving shots and places, the top four lose 7–8 points from the first to the last bin and the other seven about 5, so complex clips narrow the gap between them rather than widen it. The oracle caption of Tab. 5 raises the score on all five tasks at a quarter of the cost, gaining in routes, events, camera and cuts but not in layout. The caption comes from the hidden GT facts and covers one agent, so it bounds what perfect perception would add.

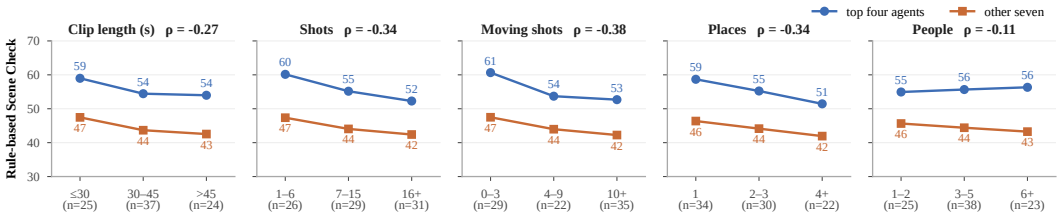


Figure 15. **What makes a reference video hard.** The 86 video tasks fall into three bins along five properties of the reference clip, task counts under each bin. Lines give the mean Rule-based Scene Check score of the top four agents by overall score and of the other seven; ρ is the Spearman correlation between the property and the task's mean score over all 11 agents. Failed runs count 0.

Photos rank the agents differently. The overall-score rankings of scripts, continuations and videos agree at Kendall $\tau = 0.85$–0.96, while photos agree with each of them at only 0.56–0.67. Opus 5.5 leads Rule-based Scene Check on photos, 4.0 points above GPT-6 Astra (95% CI 1.6–6.5) and higher on 28 of 39 tasks. Its lead comes mainly from physical validity (D1 53.6 vs. 41.4) and camera (C1 71.7 vs. 66.3). With only 39 tasks, the photo ranking rests on a small sample.

Error rates. Figure 8c reads errors from the per-check results of Rule-based Scene Check, over delivered runs. A check counts as an error when it scores below 0.5. For each run, a rate is the share of its applicable checks of one type that are errors, and each agent's rate is the mean over its runs. Picture regions and picture depth are the region-layout and depth checks of picture structure. Framing, camera angle, camera move and camera height check each shot's shot size, pitch class, move type and camera height against the hidden GT facts. Missing objects counts the objects in the hidden GT facts that the scene lacks. More shots than ref. is the share of video and continuation runs with more shots than the reference.

Shared errors barely change with capability. In Fig. 8c, errors in reading the picture fall steadily from the weakest to the best agent, while the shared errors stay within a few points. With 11 agents a weak ρ does not prove that no relation exists, so the claim rests on the narrow ranges and on the best and weakest agents being close. Over-cutting depends on the measure. The share of runs with more shots than the reference gives $\rho = -0.18$. The share of video runs with at least one extra cut gives -0.48 (52–86%, highest for GLM 5.3

Flash). Weaker agents thus add more cuts, while the best still adds them in 55% of video runs. Path excess holds on scripts and videos. The median ratio to the ground truth is 1.20 on scripts (7,307 paths; 1.15–1.23 per agent), 0.70 on continuations (138 paths) and 1.16 on videos (1,379 paths; 0.96–1.42 per agent). Scripts hold 83% of the paths, so they set the pooled 1.19 of Fig. 14a. Script paths in the hidden GT facts average 49 samples, and only 4% have two or fewer, so the excess is not an artifact of straight-line references. Direction counts cases whose ratio to the reference exceeds 1.5 or falls below two-thirds: shot counts over 1,191 video and continuation runs, path lengths per path and scene extents per task.

Challenging and common requests. Figure 14a pairs each challenging request with a common counterpart scored the same way. Camera angles use the scorer's reading from camera pitch, as in Rule-based Scene Check; counting camera height as well, 35% of low-angle shots come out at eye level. Orbits are classified by the move type of the agent's camera, which keeps 4% as orbits and turns 47% into sideways moves. The tilt shot of Fig. 14b fails for all 11 agents, and nine of them keep the camera still. The pairs were chosen after the errors were seen, and that agents fall back on common choices because challenging requests are scarce in their training data is our reading of the error direction, not a tested cause.

Looking does not fix. Each agent's runs are split into thirds by the images it views, ties broken by task id, and a run's score is taken relative to the other agents' mean on the same task, which removes task difficulty. Picture structure is treated the same way on the media tasks. Without this control the third that looks most scores 1.5 points lower at the median, since hard tasks draw more looking and score lower. Image counts include every image returned to the agent after its first Blender call, so on photo, continuation and video tasks they include reference frames viewed again. In the no-image ablation of Tab. 5, all eight runs received zero images; it covers one mid-ranked agent and eight script tasks. These comparisons describe associations rather than the effect of looking, since agents may well look more when a task goes wrong.

Verdicts against errors. We take the text an agent writes after each batch of images it views once it has made its first Blender call. A run counts as claiming a match when this text states that the render matches the reference, brief or photo, or that it looks right or correct, by a fixed pattern list. Per-shot error rates come from the Rule-based Scene Check items behind Fig. 8c. A regression with agent and task fixed effects and a bootstrap over tasks compares runs with and without a claim. Camera-angle errors differ by -0.8 points (95% CI -4.0 to $+2.1$), camera moves by $+0.1$ and missing events by -0.6. Framing differs by -2.1 (-3.7 to -0.5) and shot count by -2.5 (-5.7 to $+0.7$). A stricter pattern list and Claude Code runs alone leave camera angle, camera move and events within 1.5 points, while Claude Code runs alone show 4.7 fewer shot-count errors. The Codex CLI keeps only reasoning summaries, so GPT agents claim a match in 123 of 717 runs, and most claims come from the other eight agents. The analysis is correlational, and the pattern list is not hand-checked.

Failed Blender calls. For each Blender call that ends in an error, we take the last line of its traceback. Of 3,796 failed calls, 36% use an attribute, keyword or enum value that the Blender 5.2 API does not have; errors from the GPU-less sandbox's renderer are not counted as such. `Action.fcurves`, which Blender 5 replaced with layered actions, accounts for 23%. A run counts as calling it if any of its calls raises it. With agent and task fixed effects, runs with an API failure score $+0.1$ points (95% CI -0.8 to $+1.0$) against runs without, and runs calling `Action.fcurves` -0.3 (-1.1 to $+0.9$), so later rounds repair these calls.

Build rhythm. Agents spread their Blender calls differently over a run, measured in tool calls and pooled over each agent's runs. Gemini 4 Argon and Opus 5.5 leave 61% and 54% of their Blender calls to the last third, whereas GPT-6 Astra and GLM 5.3 Flash leave only 35% and 37%, and both rhythms reach the top. In Claude Code, Opus 5.5 runs Blender 6 times per task and DeepSeek V4.1 Flash 24, yet Opus 5.5 scores 4.5 points higher.

C.3 Event completion and legibility

Correct is not always legible, and events separate agents most. The VLM judge is asked only about facts the rules accept. Of those, about a third of the actions cannot be read from the render (68% legible on average, from 86% for GPT-6 Astra to 50% for Muse Spark 1.3), and a blind recap retells only 55% of the story. Weaker agents thus err more and also make their correct parts harder to read. Chaining the two scores per event (Fig. 16), only 36% of annotated events are both accepted by the rules and confirmed on screen: about half for the four strongest agents (44–54%) and a fifth for the four weakest, the widest gap of any error type. A further 31% pass the rules but are confirmed only partly (17%) or rejected (14%) by the judge, so the rules alone would overstate event completion. Conversely, the rules miss some posture events such as falls, which lowers the confirmed share.

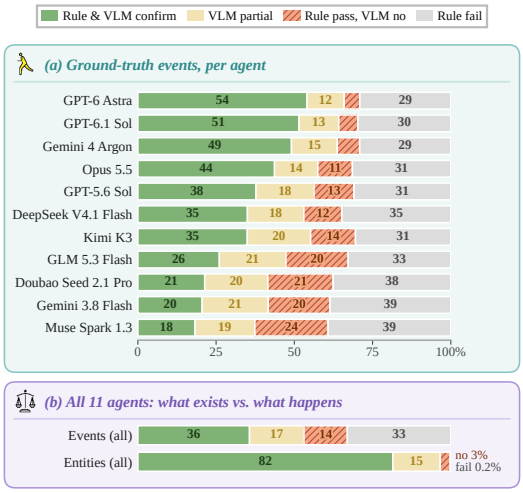


Figure 16. **Event completion.** Each hidden GT fact is triangulated by both scorers: the rules pass it and the VLM confirms it on screen, confirms it partly, or rejects it; or the rules already fail it. Facts without a usable VLM verdict are excluded. (a) Events per agent: confirmed events fall from 54% (GPT-6 Astra) to 18% (Muse Spark 1.3), and weaker agents more often stage an event the rules accept but the screen does not show. (b) Pooled over agents, entities almost never fail, while events do.

C.4 Results by scene domain and screen dimension

Scene domains. Table 7 breaks the overall score down by the seven scene domains. GPT-6 Astra leads six of them and ties GPT-6.1 Sol on Homes at 78.3. Each domain's ranking agrees with the overall ranking at Kendall $\tau = 0.85$–0.96, with the same top four in all seven, so the ranking does not hinge on the domain. Streets are the weakest domain for all 11 agents.

Screen dimensions. Table 8 splits VLM-based Screen Check into its four dimensions. Story is the lowest dimension for all 11 agents, while recognition varies only from 80.9 to 91.7; Appendix C.3 analyzes how legible the actions are.

C.5 Do proxy errors reach the real-world video?

Figure 2 gives the real-world video generation model Seedance 2.5 the proxy videos of a strong and a weak agent on two video tasks, with one prompt shared by both clips of a panel. In both panels, the real-world video generated from the weak agent's proxy keeps its error: the high camera and flat lot in (a), and the distant car in (b). Camera errors can thus reach the real-world video. These pairs show that errors can reach real-world videos, not how often they do. Correlating scores with real-world video quality across all agents is left to future work.

Table 7. **Overall score by scene domain.** Mean of (Scene + Screen)/2 over the tasks of each of the seven domains of Fig. 3, task counts in parentheses. Per column, the best score is in **bold**. ▲ and ▼ mark domains where an agent scores at least 2 points above or below its own *All*.

Agent	Homes (34)	Offices (27)	Streets (42)	Shops (19)	Industry (31)	Sports (46)	Nature (43)	All (242)
GPT-6 Astra	**78.3**	**78.9**	**74.4**▼	**80.7**▲	**77.0**	**79.2**	**77.3**	**77.7**
GPT-6.1 Sol	**78.3**▲	76.5	72.4▼	77.3	76.7	76.5	76.0	76.1
Gemini 4 Argon	75.4	74.9	73.6	76.4	75.1	75.3	75.1	75.0
Opus 5.5	76.6▲	75.9	70.3▼	74.4	75.2	74.4	73.4	74.1
GPT-5.6 Sol	70.4	70.5	66.9▼	72.8▲	70.9	70.5	71.1	70.2
DeepSeek V4.1 Flash	68.4	69.5	65.9▼	71.6	71.6▲	70.8	70.5	69.6
Kimi K3	66.4	66.7	63.6▼	67.1	69.5	69.4	69.3	67.5
Gemini 3.8 Flash	66.1	64.5	59.8▼	66.3	64.1	67.1▲	65.7	64.7
GLM 5.3 Flash	60.5▼	64.6	59.5▼	65.4	61.5▼	67.1▲	67.1▲	63.7
Doubao Seed 2.1 Pro	62.4	63.7▲	57.6▼	63.4▲	62.1	62.8	59.3	61.3
Muse Spark 1.3	61.2	62.9	56.3▼	64.3▲	58.2▼	63.8▲	62.3	61.1

Table 8. **VLM-based Screen Check per dimension** (Gemini 4 judge). Each dimension is the mean over the tasks that exercise it, task counts in parentheses; failed runs count 0. *Screen* is the score of Tab. 2. Per column, the best score is in **bold**. Figure 13 gives the Rule-based Scene Check dimensions.

Agent	V1 Recognition (242)	V2 Action (202)	V3 Visibility (203)	V4 Story (202)	Screen (242)
GPT-6 Astra	**91.7**	**85.6**	**88.7**	**68.6**	**85.4**
GPT-6.1 Sol	91.1	82.3	86.3	64.4	83.1
Gemini 4 Argon	87.9	81.2	83.7	66.8	81.7
Opus 5.5	88.8	76.7	82.9	63.1	79.8
GPT-5.6 Sol	89.9	69.7	80.3	58.1	77.5
DeepSeek V4.1 Flash	88.8	72.1	82.4	57.2	77.9
Kimi K3	85.5	67.5	77.5	55.7	73.5
Gemini 3.8 Flash	83.5	55.6	74.2	46.0	68.7
GLM 5.3 Flash	84.3	57.9	76.0	46.6	69.5
Doubao Seed 2.1 Pro	81.3	54.0	71.0	41.4	66.2
Muse Spark 1.3	80.9	50.2	68.5	41.2	65.0

C.6 Case gallery

Figures 17 and 18 show one task per scene domain, with the proxy video of every agent at the moment of the reference frame.

C.7 Agent statistics and cost

Table 9 gives per-task means of the agents' process and cost over all 242 tasks. Figure 19 relates effort to score: across the 11 agents, more turns go with a lower Rule-based Scene Check score (Spearman $\rho = -0.77$), while output tokens do not track it ($\rho = 0.11$).

Table 9. **Process and cost per task on all 242 tasks.** Means of turns, tool calls, wall-clock minutes, input tokens (millions, cache hits included) and output tokens (thousands, reasoning included); cost in US dollars at the official list prices of late October 2026, computed per request from the API relay logs. *Failed*: runs without a valid scene after re-execution, counted as in Tab. 2. Agents in the order of Tab. 2.

Agent	Turns	Tools	Min.	In (M)	Out (k)	Cost ($)	Failed
GPT-6 Astra	71	70	93	11.2	149	32.59	0
GPT-6.1 Sol	62	60	105	10.7	195	6.71	0
Gemini 4 Argon	49	85	73	5.8	181	4.45	2
Opus 5.5	47	61	54	6.7	137	7.82	0
GPT-5.6 Sol	82	81	39	6.8	54	6.19	0
DeepSeek V4.1 Flash	131	141	91	27.9	261	1.89	0
Kimi K3	106	129	196	20.6	226	17.66	4
Gemini 3.8 Flash	143	141	88	16.0	166	2.73	0
GLM 5.3 Flash	91	116	89	12.7	135	0.73	3
Doubao Seed 2.1 Pro	143	168	116	22.8	143	5.96	1
Muse Spark 1.3	88	130	108	7.8	157	3.84	4

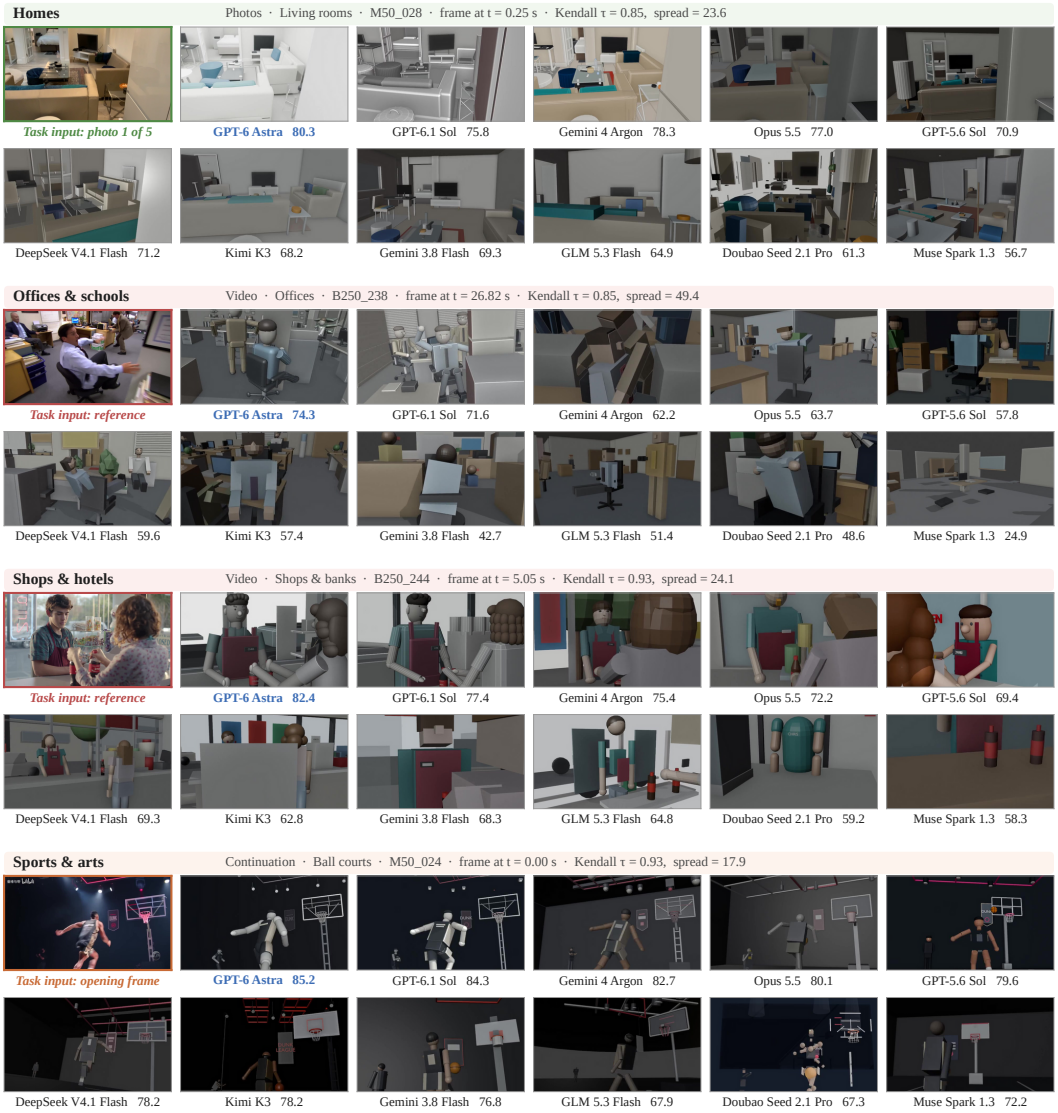


Figure 17. **Case gallery: homes, offices and schools, shops and hotels, sports and arts.** Each domain shows the task input, a reference-video frame, the first of five photos or the opening frame of a continuation, then the frame at the same moment from each agent's proxy video, agents in the order of Tab. 2. Each frame carries the agent's overall score on that task, the best per task in bold. Each task has the highest Kendall τ between its agent ranking and the overall ranking among the media tasks of its domain on which all 11 runs succeed, ties broken by the larger score spread. Frame times are reference keyframes with people or action in view.

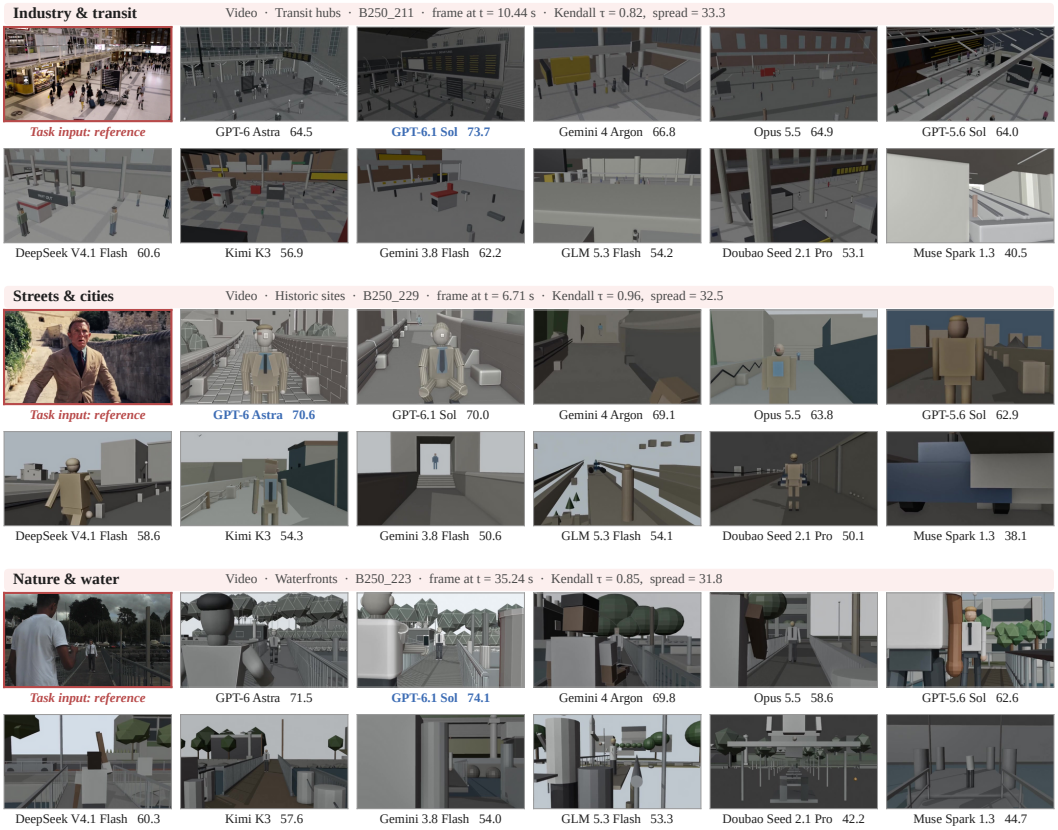


Figure 18. **Case gallery: industry and transit, streets and cities, nature and water.** Layout and task selection as in Fig. 17.

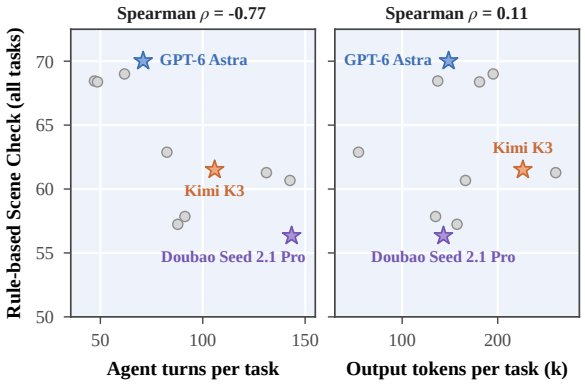


Figure 19. **Effort vs. score.** Mean turns and output tokens per task against the Rule-based Scene Check score (all tasks), one marker per agent (stars: best, median and weakest); Spearman ρ over the 11 agents.