REELONCE: ADAPTIVE REFERENCE POLICY AND EVALUATION-DRIVEN HIERARCHICAL REFINEMENT FOR AGENTIC MULTI-SHOT STORY VIDEO GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-shot story video generation requires shot-specific visual guidance while preserving consistency across characters, scenes, and narrative events, often requiring repeated manual revision. Existing workflows struggle to determine whether a problem originates in story planning, reference construction, or video generation. Fixed-stage retries are costly and may miss the root cause or disrupt content that is already correct. We present ReelOnce, an agentic framework that combines an adaptive reference policy with evaluation-driven hierarchical refinement. It selects an appropriate reference format for each shot, distinguishes the state depicted in a reference from the transition to be generated, and uses evaluation findings to identify the smallest necessary repair scope. Persistent story memory preserves narrative constraints and unaffected decisions throughout refinement. We further introduce Reconstructed Story Semantic Alignment (RSSA), which measures semantic consistency with the source story by reconstructing observable events from generated videos. On Pixar-style videos generated from 20 diverse stories, ReelOnce achieves strong RSSA performance while remaining competitive across VBench quality dimensions, demonstrating that the framework can balance story fidelity with visual quality.

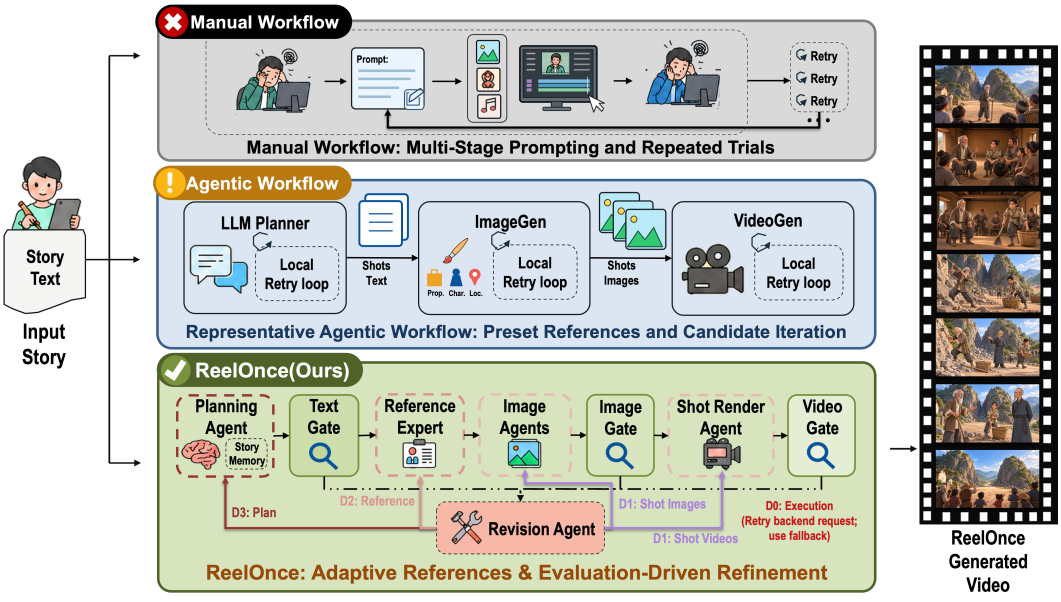

Figure 1: Workflow comparison of manual production, a representative agentic pipeline, and ReelOnce. ReelOnce couples persistent story memory with an adaptive reference policy and evaluation-driven hierarchical refinement to promote a candidate or retain the baseline.

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

1 INTRODUCTION

Text-to-video models can produce visually convincing short clips (Ho et al., 2022; Singer et al., 2023; Blattmann et al., 2023; Kondratyuk et al., 2024; Bar-Tal et al., 2024; Chen et al., 2024; Yang et al., 2025; Kong et al., 2025). Generating a coherent sequence from a story requires control across shots: characters must retain their identities, events must unfold in the intended order, and later shots must reflect earlier changes.

Prior work addresses story decomposition, long-horizon synthesis, visual consistency, and evaluation-guided generation. LLM planners and multi-agent systems organize scenes, shots, and multimodal production (Lin et al., 2024; Xie et al., 2024; Xu et al., 2025; Wu et al., 2026); long-video methods extend temporal context (Qiu et al., 2024; Henschel et al., 2025); and visual references or persistent memory preserve recurring entities (He et al., 2025; Zhou et al., 2024; Zhang et al., 2025; Zhou et al., 2026). AniMaker and ViMax further introduce evaluation or candidate selection into agentic production (Shi et al., 2025; Huang et al., 2026a). We study two coupled decisions that remain central to this workflow: how to condition each shot and which upstream choice to revisit after evaluation.

Visual references serve distinct purposes: identity assets constrain appearance, shot images specify spatial configurations, and storyboard panels organize action progression. Their suitability depends on what a shot must preserve and what it must change. Errors in these early decisions can propagate through image and video generation. Stage-wise evaluation offers opportunities to correct them before further production, while defects visible only in the resulting video may require revisiting an earlier decision. Both reference selection and repair therefore need a shared representation of narrative requirements and dependencies across shots.

We introduce ReelOnce, an agentic framework for multi-shot story video generation (Figure 1). ReelOnce connects planning, reference selection, media realization, evaluation, and revision through persistent story memory. A Planning Agent builds shared story memory and shot contracts, and a Reference Expert selects composition, identity, joint, or panel-based conditioning for each shot. Evaluation Agents inspect text, images, and videos, while a Revision Agent uses their evidence to select the smallest justified intervention level and affected shots. Dependent outputs are regenerated and reevaluated before candidate adoption, allowing video feedback to trigger upstream repair without discarding unaffected decisions.

Our contributions are:

1. **An adaptive reference policy** that matches composition, identity, joint, or panel-based conditioning to shot requirements and separates reference states from intended transitions.
2. **Evaluation-driven hierarchical refinement** that selects repair levels and target shots throughout production, including upstream revisions triggered by video feedback and dependency-bounded regeneration.
3. **A unified agentic framework** that integrates both mechanisms through **persistent story memory**, evaluated through system comparisons, component ablations, and qualitative repair cases.

2 RELATED WORK

Story planning and agentic production. VideoDirectorGPT develops multi-scene plans (Lin et al., 2024); DreamFactory iterates key frames in a multi-agent pipeline (Xie et al., 2024); MM-StoryAgent coordinates multimodal production (Xu et al., 2025); and MovieAgent uses hierarchical agents (Wu et al., 2026). ViMax combines planning, memory, and candidate selection (Huang et al., 2026a). ReelOnce focuses on coordinating shot-specific reference selection with repairs across production stages through shared story constraints.

Long-horizon video generation. FreeNoise reschedules initial noise for longer videos (Qiu et al., 2024), StreamingT2V uses memory across autoregressive chunks (Henschel et al., 2025), and Steins-Gate models causal transitions (Huang et al., 2026b). These methods extend continuity within the

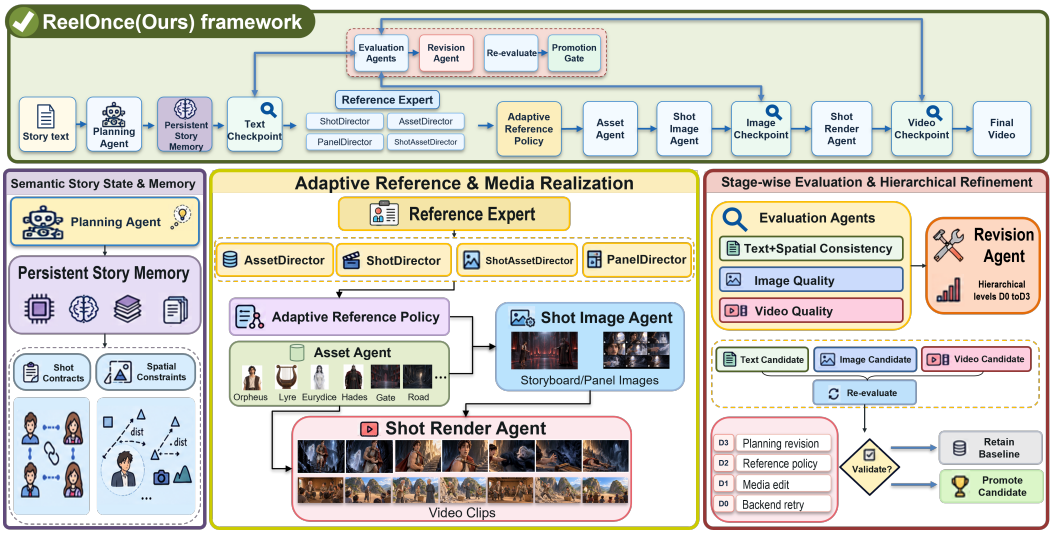

Figure 2: Overview of ReelOnce. Shared story memory connects planning, four reference directors, and media realization. Evaluation operates at text, image, and video stages; the Revision Agent selects a repair level and target shots from the resulting evidence. Upstream changes trigger regeneration of dependent outputs and focused reevaluation. D0–D3 denote intervention levels, available as appropriate throughout the loop.

generator; ReelOnce operates at the production-workflow level, where shots may require different reference formats and repair scopes.

Visual consistency and reference control. Story visualization preserves recurring subjects through memory, shared attention, and reference reuse (Rahman et al., 2023; Zhou et al., 2024; Liu et al., 2025; Singh et al., 2025; Mao et al., 2024). DreamStory, MAGREF, and DanceTogether provide identity and multi-subject controls (He et al., 2025; Deng et al., 2025; Chen et al., 2025); StoryMem and VideoMemory extend explicit memory to multi-shot video (Zhang et al., 2025; Zhou et al., 2026); and LogiStory models roles and causal relations (Meng et al., 2026). ReelOnce treats identity, spatial layout, and action progression as distinct conditioning needs and routes each shot accordingly.

Evaluation and feedback-driven revision. Video benchmarks measure perceptual quality, motion, prompt alignment, compositional relations, and narrative expression (Huang et al., 2024; Liu et al., 2024; He et al., 2024; Sun et al., 2025; Feng et al., 2026). Language-agent methods use feedback for iterative revision (Madaan et al., 2023; Shinn et al., 2023), and AniMaker uses AniEval for candidate search (Shi et al., 2025). ReelOnce adapts image and video evaluation methods from ReelBench (Anonymous, 2026) to identify affected shots, select an intervention level, and assess regenerated outputs.

3 METHOD

Given a story x, ReelOnce constructs an ordered shot plan $S = (s_1, \ldots, s_n)$ and generates a video V. Each shot specifies an inherited state and an intended transition. These requirements guide two coupled decisions: how to condition generation and which decisions to revisit when evaluation finds a discrepancy. Persistent story memory provides the common context for reference selection and hierarchical refinement across the production process. Figure 2 details the agents, evaluation checkpoints, and intervention levels.

3.1 AGENTIC PRODUCTION OVERVIEW

The Planning Agent decomposes the story, expands underspecified actions into observable events, extracts assets, and establishes shot contracts while preserving source events and relations. Text evaluation and revision check this plan before reference construction. The Reference Expert then routes each shot to a director, which constructs static and dynamic descriptions for the required media agents. The Asset Agent, Shot Image Agent, and Shot Render Agent realize identity assets, shot images or panels, and video clips, respectively. Evaluation Agents inspect these artifacts at their corresponding checkpoints, and the Revision Agent uses the resulting evidence and dependency relations to decide whether to repair locally or reopen an upstream decision. The Promotion Gate validates the candidate, promotes it when the targeted constraints improve without regressions, and otherwise retains the baseline.

Persistent story memory provides the shared semantic context through which planning establishes constraints, reference selection instantiates them for each shot, media agents realize them, and refinement revisits only decisions implicated by evidence. Agents pass structured state and confirmed decisions across stages.

3.2 SHARED STORY MEMORY AND SHOT CONTRACTS

Persistent story memory C records the entities, relations, state changes, and shot requirements shared by reference selection and refinement. At step t, the execution state is

$$P_t = (C, A_t, I_t, R_t),$$

where A_t contains selected media and dependencies, I_t contains active issues and evidence, and R_t records execution context and resources. Media-level refinement preserves C; only a semantic change creates a validated successor snapshot.

The memory distinguishes *narrative invariants*, *intended transitions*, and *realization constraints*. Each shot contract specifies visible entities, entry state, required transition, exit state, and spatial relations. Identity bindings preserve who participates; entry and exit states connect events across shots; spatial constraints specify how entities interact. The contract therefore records both what must be inherited when a shot begins and what state must be handed to the next shot when it ends, including the decisions that are already confirmed and should remain protected during repair. Deterministic checks validate identifiers, coverage, inheritance, visibility, and spatial consistency. The same contracts guide generation, evaluation, and revision, so an upstream change can be traced to the shots that depend on it. This lets the Revision Agent separate a local defect from a protected decision that should not be changed incidentally.

For issue q_t and context budget B, ReelOnce compiles $W_t = \mathrm{Compile}(P_t, q_t, B)$ from the relevant shot contracts, continuity dependencies, evidence, and permitted changes. This retains the surrounding constraints without carrying unrelated generation history.

3.3 ADAPTIVE REFERENCE POLICY

Different shots require different conditioning carriers. The Reference Expert therefore chooses among four peers: SHOTDIRECTOR provides a composition image, ASSETDIRECTOR provides identity assets, SHOTASSETDIRECTOR combines both, and PANELDIRECTOR provides ordered action panels. All implement the same reference-plan interface, while backend capabilities constrain the executable choices.

For shot s_i, let $\mathcal{A}_i$ denote its active entities and available visual assets, and let $\mathcal{K}$ denote backend capabilities. The Reference Expert selects a director as follows:

$$d_i = \pi_\theta(C, s_i, \mathcal{A}_i, \mathcal{K}) \in \mathcal{D},$$
$$\mathcal{D} = \{D_{\texttt{shot}}, D_{\texttt{asset}}, D_{\texttt{shot_asset}}, D_{\texttt{panel}}\},$$

where π_θ is a prompted router used without task-specific training. It weighs identity requirements, spatial interactions, action progression, inherited state, and available controls. Identity assets can support a recurring subject, a shot image can establish interaction geometry, and ordered panels can specify successive action states.

The selected director g_{d_i} constructs a structured reference plan z_i:

$$z_i = g_{d_i}(C, s_i, \mathcal{A}_i, \mathcal{K}), \qquad r_i = \text{Compile}(z_i, d_i, \mathcal{K}), \qquad \text{Valid}(r_i, d_i) = 1.$$

The plan z_i records reference roles, static and dynamic descriptions, duration, and requirement coverage. Static descriptions specify the states to depict in an image or panel sheet; dynamic descriptions specify the motion and transitions to generate. Compilation turns these descriptions into generation prompts under the same shot contract and validates the requested mode and input roles. A preceding frame may supply additional continuity evidence alongside the selected primary reference. The Revision Agent can revise this conditioning decision while retaining the narrative requirements of the shot. Director contracts and PANELDIRECTOR timing rules appear in Appendix A and Appendix F.

3.4 MEDIA REALIZATION AGENTS

The Asset Agent creates or retrieves reusable identity assets. When required by the selected mode, the Shot Image Agent generates a composition image or panel sheet from the static prompt. The Shot Render Agent uses the designated references and dynamic prompt to generate the clip. These agents share entity bindings and spatial constraints but have different outputs: assets establish reusable identity, shot images establish composition or ordered states, and rendering converts the selected state into motion. Evaluation checks the available images and then the generated videos against their contracts. Each artifact retains its producing stage and dependencies, allowing a later repair to rebuild the outputs affected by an upstream change.

3.5 STAGE-WISE EVIDENCE EVALUATION AND HIERARCHICAL REFINEMENT

Hierarchical refinement addresses error propagation throughout production. Early feedback supports corrections before further generation, while video feedback can expose problems in previously selected plans or references. The loop therefore separates where a problem is observed from where a change is needed.

Evaluation issue package. Evaluation Agents compare artifacts with the current story memory and shot contracts. Text checks cover story requirements, entity bindings, events, and spatial state. Image checks assess identity, composition, spatial relations, and visual realization. Video checks assess instruction alignment, subject fidelity, technical quality, and physical plausibility, together with visual, cinematic, and narrative continuity between adjacent shots. Image and video evaluation adapt methods from ReelBench (Anonymous, 2026) to the current shot contracts and targeted reevaluation; the online evaluation dimensions, issue-normalization rules, and quality thresholds appear in Appendix A.7. Text checks operate on plans before media generation. RSSA is a separate offline metric.

Each report is converted into an issue package containing the affected shot or shot pair, target constraint, evidence, source/version, confidence, severity, and allowed repair depth. The target constraint states what the artifact is expected to preserve or change; source/version ties the finding to the exact memory or media state that produced it; and allowed depth prevents a weak observation from authorizing an unnecessarily broad rewrite. ReelOnce links each issue to the evaluated artifact and normalizes reports from text, image, and video checks. Full record fields and service settings appear in Appendix A.

Severity and executability. Severity captures impact on story constraints, cross-shot continuity, and candidate usability, with visual salience treated as supporting evidence. Executability records whether the available evidence is sufficiently specific to support a controlled repair. Contract-grounded *hard* issues may support D0–D3, while perceptual *soft* issues remain at D0–D1 unless upstream evidence supports a deeper repair. The Revision Agent checks findings against protected story requirements before selecting an action; incomplete service responses remain execution issues rather than evidence of a media defect. Normalization rules and thresholds appear in Appendix A.7.

Repair-target confirmation. The Revision Agent aligns each issue with story memory, shot contracts, dependency relations, and protected constraints before proposing a revision. It confirms (i)

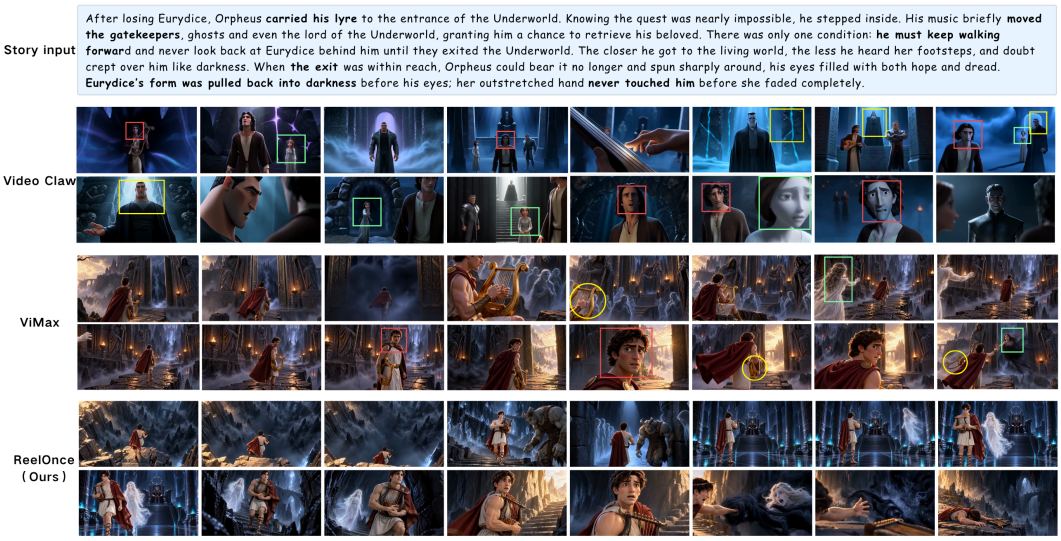

Figure 3: Cross-method comparison on the Orpheus story. Matching colored boxes track the same characters; yellow circles mark the recurring lyre. ReelOnce maintains more stable identities, prop continuity, and narrative progression in this illustrative case.

which constraint was violated, (ii) which shot or shot pair is affected, (iii) which upstream or downstream artifact must be regenerated, (iv) which already-correct results must be preserved, and (v) which checks must be rerun after the repair. It seeks the smallest intervention supported by the evidence, moving upstream only when the discrepancy implicates an earlier decision. Findings without sufficient evidence are deferred. The intervention levels in Figure 2 identify the scope of intervention: D0 retries backend execution, D1 repairs local image or video media, D2 revises reference mode or inputs, and D3 revises story or spatial plans.

For example, a missing action in a video may call for a local prompt and render revision at D1, a reference change at D2, or a plan revision at D3. A text-stage finding can likewise trigger D3 before media generation. After selecting direct targets, the system follows their dependencies to identify outputs that consume a changed artifact or inherit a changed state. It regenerates this affected set in production order, preserving unrelated shots. Thus, video feedback can return to planning or reference construction and then propagate the repair forward through dependent images and clips.

Candidate validation and promotion. Validation checks the declared change scope and reference inputs. Focused reevaluation checks the repair objectives and affected continuity boundaries while protecting established story requirements. The stage-specific selection rule compares the candidate with the baseline; accepted candidates become the selected state, while rejected candidates leave the baseline intact. Remaining video issues can trigger another scoped revision within the available limits. This closes the path from downstream observation through upstream correction to regenerated and reevaluated video.

3.6 Unified Generation and Refinement Loop

ReelOnce interleaves generation, evaluation, and refinement across text, image, and video stages. In the reported full configurations, D1–D3 allow two semantic rounds, one post-media reopening, and two media-repair rounds per shot; exhausted limits retain the selected baseline. D0 retries only transient provider failures, while deterministic failures terminate. The complete procedure and budget accounting appear in Appendices A.1 and A.6.

Table 1: Overall comparison and ablations. RSSA Total is reported as a mean only. LoCoT2V CC: macro story mean with a story-bootstrap 95% confidence interval. Success: percentage of complete, evaluable final videos. Bold and underlined mark the best and second-best available values. The w/o Mem. and w/o Ref. Policy variants also disable repair.

Method	RSSA Total ↑	LoCoT2V CC (95% CI) ↑	Success (%) ↑
VideoClaw	36.03	0.8155 [0.7815, 0.8497]	<u>90.0</u>
ViMax (Pixar)	60.98	0.8227 [0.7747, 0.8648]	**100.0**
ReelOnce w/o Mem.	76.16	0.8478 [0.8246, 0.8713]	85.0
ReelOnce w/o Ref. Policy	75.82	0.8395 [0.8053, 0.8723]	<u>90.0</u>
ReelOnce w/o Repair	76.45	0.8522 [0.8203, 0.8818]	<u>90.0</u>
ReelOnce (Gemini)	<u>77.90</u>	<u>0.8534</u> [0.8261, 0.8793]	**100.0**
ReelOnce (GPT)	**78.70**	**0.8760** [0.8518, 0.9017]	**100.0**

4 EXPERIMENTS

We evaluate story adherence, cross-shot character consistency, completion, visual quality, and the contributions of adaptive references, memory, and refinement. Automatic results appear in Table 1 and Table 2; Figure 3 provides a qualitative cross-method comparison; human judgments appear in Table 3; and Figure 4 examines localized repairs.

4.1 EXPERIMENTAL SETUP

Stories and metrics. We evaluate on 20 narrative stories from ReelBench (Anonymous, 2026), spanning diverse fables, mythology, and multi-character scenarios, using their texts as generation inputs and semantic targets. RSSA reconstructs observable events from four sampled frames per shot without source text and scores seven components; reconstruction prompts, the scoring rubric, and aggregation procedures appear in Appendix C. We report RSSA Total, LoCoT2V-Bench (Zheng et al., 2026) Character Consistency (CC) with a story-bootstrap 95% confidence interval, the final-video success rate, and seven VBench dimensions for visual quality and motion (Huang et al., 2024); temporal style is excluded.

Systems and protocol. We compare ReelOnce (GPT-5.6-luna) and ReelOnce (Gemini-3.8-flash) with ViMax (Huang et al., 2026a) and VideoClaw (HITsz-TMG, 2026). ReelOnce w/o Repair disables online quality checks and method-level revisions, while the other GPT ablations additionally remove memory or use uniform SHOTDIRECTOR; these ablations also disable cross-shot media references and reuse the available asset images from the corresponding configuration (Appendix B). All generations are executed using GPT Image 2 and Seedance 2.0 configured to a 16:9 aspect ratio and 720p resolution with native audio. Human evaluation uses five 1–5 dimensions adapted from MovieAgent (Wu et al., 2026) and pairwise criteria following ViMax (Huang et al., 2026a) (Appendix D).

4.2 QUANTITATIVE ANALYSIS

Overall comparison. Under the primary RSSA evaluator, ReelOnce (GPT-5.6-luna) has the highest mean Total (78.70), followed by ReelOnce (Gemini-3.8-flash) (77.90); both complete all 20 assigned stories (100% success rate). LoCoT2V Character Consistency is highest for GPT (0.8760), followed by Gemini (0.8534), although their bootstrap intervals overlap. On the reported cohorts, both configurations exceed ViMax (60.98) and VideoClaw (36.03) in mean RSSA Total. RSSA evaluations with two alternative VLMs show a consistent overall advantage for the complete ReelOnce configurations (Appendix C.4).

VBench quality. ReelOnce (GPT-5.6-luna) has the highest mean over the seven displayed VBench dimensions (87.90), followed by ReelOnce (Gemini-3.8-flash) (85.81). GPT leads subject consistency, background consistency, aesthetic quality, imaging quality, and dynamic degree; VideoClaw

Table 2: VBench results, scaled by 100. Mean is the unweighted average of the seven displayed dimensions. Bold and underlined mark the highest and second-highest values per column.

Method	VBench (×100)							
	SC	BC	AQ	IQ	TF	MS	DD	Mean
VideoClaw	84.80	92.30	63.82	71.97	**98.68**	**99.29**	55.94	80.97
ViMax (Pixar)	89.66	<u>93.05</u>	<u>69.63</u>	<u>74.92</u>	97.35	98.89	72.22	85.10
ReelOnce w/o Mem.	83.66	90.44	68.05	72.05	96.56	98.24	79.11	84.02
ReelOnce w/o Ref. Policy	85.73	91.08	67.90	73.00	96.97	98.49	59.72	81.84
ReelOnce w/o Repair	85.75	91.22	67.46	72.29	96.80	98.40	79.43	84.48
ReelOnce (Gemini)	<u>90.22</u>	92.60	66.70	73.85	97.06	98.66	<u>81.56</u>	<u>85.81</u>
ReelOnce (GPT)	**92.92**	**93.83**	**70.01**	**75.39**	<u>97.76</u>	<u>99.05</u>	**86.36**	**87.90**

Table 3: Human evaluation: (a) five-dimensional mean ratings (1–5); (b) GPT non-tie win rates (%). CSC/SF/AQ denote cross-scene consistency, semantic following, and aesthetic quality. ReelOnce rows use different backends. Bold and underlined values mark the highest and second-highest values per column.

(a) Mean ratings (1–5, ↑)

Method	Char.	Narr.	Phys.	Script	Visual
ViMax	3.64	2.84	<u>3.22</u>	3.12	3.46
VideoClaw	2.36	2.80	3.16	2.32	3.29
ReelOnce (Gemini)	<u>3.99</u>	**3.38**	3.14	<u>3.71</u>	**3.66**
ReelOnce (GPT)	**4.03**	<u>3.29</u>	**3.24**	**3.72**	<u>3.65</u>

(b) GPT non-tie win rate (%, ↑)

Opponent	CSC	SF	AQ	Mean
ViMax	<u>79.4</u>	<u>84.6</u>	<u>63.0</u>	<u>75.7</u>
VideoClaw	**90.6**	**88.3**	**70.0**	**83.0**
ReelOnce (Gemini)	51.3	52.5	52.9	52.2

leads temporal flickering and motion smoothness. Temporal flickering and motion smoothness remain close across methods, whereas dynamic degree shows much larger variation in motion prevalence.

4.3 Ablation Analysis

Full GPT reaches a mean RSSA Total of 78.70 and completes all 20 stories, compared with 76.45 and 18 completed stories without online evaluation and refinement. Alongside these gains, subject consistency rises from 85.75 to 92.92, indicating stronger visual consistency in the complete configuration. The configuration also lacking persistent memory and cross-shot media references has a mean Total of 76.16 and completes 17 stories. The fixed-reference configuration, which also disables cross-shot media references, has a mean Total of 75.82 versus 76.45 for w/o Repair; its displayed VBench mean (81.84 versus 84.48) and dynamic degree (59.72 versus 79.43) are also lower. The VBench mean difference is concentrated in dynamic degree, which reflects motion prevalence.

4.4 Human Evaluation

In Table 3, ReelOnce (GPT-5.6-luna) leads character consistency, physical-law adherence, and script faithfulness; ReelOnce (Gemini-3.8-flash) leads narrative coherence and visual appeal. GPT has higher non-tie win rates than ViMax and VideoClaw, with a close comparison against Gemini. Both are complete configurations with their respective model backends and online-evaluation settings; coverage appears in Appendix D.

4.5 Qualitative Analysis

In Figure 3, matching boxes track characters and yellow circles mark the lyre. VideoClaw changes character appearance across shots; ViMax maintains scene atmosphere but varies character details

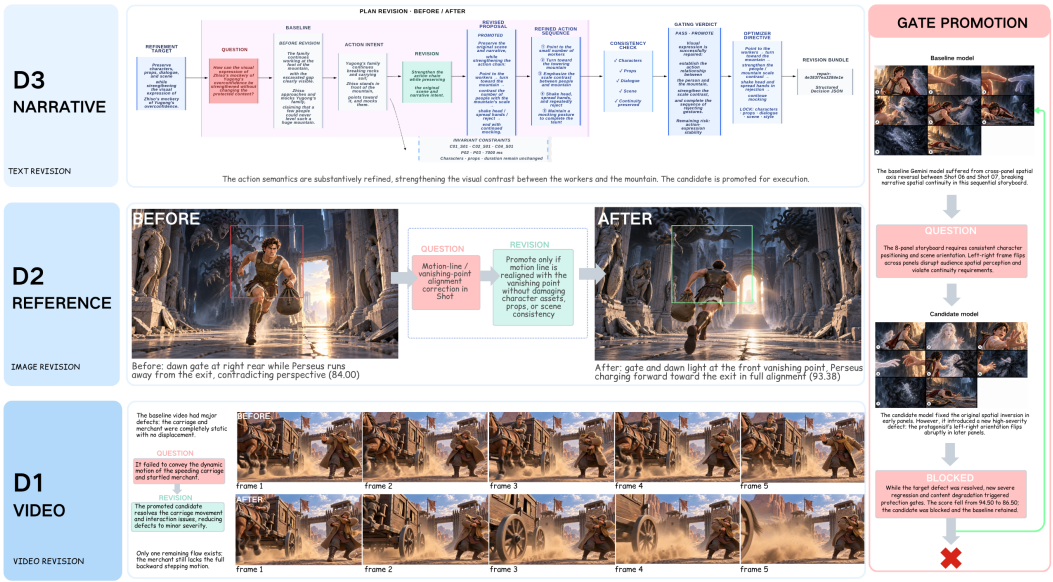

Figure 4: Representative hierarchical repairs: D3 narrative revision, D2 reference and spatial correction, and D1 video refinement (left), followed by candidate validation and rollback after a new cross-panel orientation reversal (right). Additional cases appear in Appendix E.

and lyre presence, position, and form. ReelOnce preserves the Orpheus–Eurydice identities and lyre relationship, with more coherent progression through the Underworld to the final outcome.

In Figure 4, D3 clarifies the contrast between workers and the mountain while preserving story constraints; D2 aligns the gate and forward motion; D1 restores carriage movement, with a residual stepping defect. The Promotion Gate rejects a candidate that corrects an early spatial error but introduces a later orientation reversal, retaining the baseline. These cases connect repairs across production stages with checks of the target issue and protected continuity. Additional cases appear in Appendix E.

5 Discussion and Limitations

ReelOnce turns image and video evaluation feedback into concrete production decisions through shared story state and hierarchical refinement. It provides a practical foundation for automated video production at scale and illustrates how evaluation can guide planning, reference construction, and media generation. Linking evaluation evidence, revision actions, and candidate outcomes makes production decisions traceable and provides a basis for refining the workflow. Frame sampling still provides limited coverage of continuous actions and temporal changes. Improving planning reliability and prioritizing repairs when multiple issues coexist are important directions for improving the quality and efficiency of automated video production.

6 Conclusion

We present ReelOnce, a unified framework for multi-shot story video production that integrates adaptive reference selection and evaluation-driven hierarchical refinement through persistent story memory. The framework organizes visual references according to shot requirements and translates evaluation feedback into targeted production adjustments, connecting planning, generation, and repair in a continuous loop. Automatic metrics, human evaluation, and ablation studies support the effectiveness of the framework. ReelOnce achieves strong results in story fidelity, character consistency, and visual quality, and completes video generation for all test stories with both evaluated model configurations. These results provide a foundation for controllable and scalable automated video production.

AI USE STATEMENT

Generative AI tools were used to assist with language editing and polishing and with literature retrieval and discovery. The authors reviewed all AI-assisted edits and checked retrieved references against their original sources. The authors take responsibility for the final content of the paper, including its technical claims, experiments, and citations.

REPRODUCIBILITY STATEMENT

The appendices document the generation and refinement procedure, reference-mode interfaces, and execution budgets (Appendix A), ablation configurations (Appendix B), RSSA prompts and aggregation rules (Appendix C), and the human-evaluation protocol (Appendix D). An anonymized supplementary package provides code and representative video examples. Model and evaluator configurations are reported alongside evaluation coverage to support reproduction and interpretation of the results.

REFERENCES

Anonymous. Reelbench: Towards systematic evaluation of multi-shot narrative video generation under image reference. Under review, 2026.

Omer Bar-Tal, Hila Chefer, Omer Tov, Charles Herrmann, Roni Paiss, Shiran Zada, Ariel Ephrat, Junhwa Hur, Guanghui Liu, Amit Raj, Yuanzhen Li, Michael Rubinstein, Tomer Michaeli, Oliver Wang, Deqing Sun, Tali Dekel, and Inbar Mosseri. Lumiere: A space-time diffusion model for video generation. *ACM Transactions on Graphics*, 43(6), 2024.

Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023.

Haoxin Chen, Yong Zhang, Xiaodong Cun, Menghan Xia, Xintao Wang, Chao Weng, and Ying Shan. Videocrafter2: Overcoming data limitations for high-quality video diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7310–7320, 2024.

Junhao Chen, Mingjin Chen, Jianjin Xu, Xiang Li, Junting Dong, Mingze Sun, Puhua Jiang, Hongxiang Li, Yuhang Yang, Hao Zhao, Xiaoxiao Long, and Ruqi Huang. Dancetogether! identity-preserving multi-person interactive video generation, 2025. URL https://arxiv.org/abs/2505.18078.

Yufan Deng, Yuanyang Yin, Xun Guo, Yizhi Wang, Jacob Zhiyuan Fang, Shenghai Yuan, Yiding Yang, Angtian Wang, Bo Liu, Haibin Huang, and Chongyang Ma. MAGREF: Masked guidance for any-reference video generation with subject disentanglement, 2025. URL https://arxiv.org/abs/2505.23742.

X. Feng, H. Yu, M. Wu, S. Hu, J. Chen, C. Zhu, J. Wu, X. Chu, and K. Huang. Narrlv: Towards a comprehensive narrative-centric evaluation for long video generation, 2026. URL https://arxiv.org/abs/2507.11245.

Xuan He, Dongfu Jiang, Ge Zhang, Max Ku, Achint Soni, Sherman Siu, Haonan Chen, Abhranil Chandra, Ziyan Jiang, Aaran Arulraj, et al. Videoscore: Building automatic metrics to simulate fine-grained human feedback for video generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 20297–20325, 2024.

Yuchen He, Mengmeng Zhang, Zhentao Dong, Yong Guo, Bo Liu, Ran He, and Shiguang Shan. DreamStory: Open-domain story visualization via multi-subject personalization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(12):11874–11891, 2025. doi: 10.1109/TPAMI.2025.3600149.

Roberto Henschel, Levon Khachatryan, Hayk Poghosyan, Daniil Hayrapetyan, Vahram Tadevosyan, Zhangyang Wang, Shant Navasardyan, and Humphrey Shi. StreamingT2V: Consistent, dynamic, and extendable long video generation from text. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7065–7074, 2025.

HITsz-TMG. VideoClaw: AI automated video generation employee. GitHub repository, 2026. URL https://github.com/HITsz-TMG/VideoClaw.

Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Rui Gao, Alexey Gritsenko, Diederik P. Kingma, Ben Poole, Mohammad Norouzi, David J. Fleet, and Tim Salimans. Video diffusion models. In *Advances in Neural Information Processing Systems*, 2022.

Lingxuan Huang, Sizhe He, Hengji Zhou, Liqiang Nie, Lianghao Xia, and Chao Huang. Vimax: Agentic video generation, 2026a. URL https://arxiv.org/abs/2606.07649.

Yufei Huang, Liangyu Yuan, Changxi Chi, Yunfan Liu, Cheng Tan, Siyuan Li, Jingbo Zhou, Haitao Lin, Chang Yu, and Stan Z. Li. Steinsgate: Adding causality to diffusions for long video generation via path integral. In *International Conference on Learning Representations*, volume 2026, pp. 10378–10396, 2026b. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/117f9f1665afe711eb618f70b3a152a4-Paper-Conference.pdf.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, C. C. Chan, et al. Vbench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Dan Kondratyuk, Lijun Yu, Xiuye Gu, José Lezama, Carl Doersch, Catalin Ionescu, David Feinstein, David Tan, Irfan Essa, et al. Videopoet: A large language model for zero-shot video generation. In *International Conference on Machine Learning*, 2024.

Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, Kathrina Wu, Qin Lin, Junkun Yuan, Yanxin Long, Aladdin Wang, Andong Wang, Changlin Li, Duojun Huang, Fang Yang, Hao Tan, Hongmei Wang, Jacob Song, Jiawang Bai, Jianbing Wu, Jinbao Xue, Joey Wang, Kai Wang, Mengyang Liu, Pengyu Li, Shuai Li, Weiyan Wang, Wenqing Yu, Xinchi Deng, Yang Li, Yi Chen, Yutao Cui, Yuanbo Peng, Zhentao Yu, Zhiyu He, Zhiyong Xu, Zixiang Zhou, Zunnan Xu, Yangyu Tao, Qinglin Lu, Songtao Liu, Dax Zhou, Hongfa Wang, Yong Yang, Di Wang, Yuhong Liu, Jie Jiang, and Caesar Zhong. Hunyuanvideo: A systematic framework for large video generative models, 2025. URL https://arxiv.org/abs/2412.03603.

Han Lin, Abhay Zala, Jaemin Cho, and Mohit Bansal. VideoDirectorGPT: Consistent multi-scene video generation via LLM-guided planning. In *Conference on Language Modeling*, 2024.

Tao Liu, Kai Wang, Senmao Li, Joost van de Weijer, Fahad Khan, Shiqi Yang, Yaxing Wang, Jian Yang, and Ming-Ming Cheng. One-prompt-one-story: Free-lunch consistent text-to-image generation using a single prompt. In *International Conference on Learning Representations*, volume 2025, pp. 24470–24497, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/3d681cc4487b97c08e5aa67224dd74f2-Paper-Conference.pdf.

Yaofang Liu, Xiaodong Cun, Xuebo Liu, Xintao Wang, Yong Zhang, Haoxin Chen, Yang Liu, Tieyong Zeng, Raymond Chan, and Ying Shan. EvalCrafter: Benchmarking and evaluating large video generation models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22254–22264, 2024.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2303.17651.

Jiawei Mao, Xiaoke Huang, Yunfei Xie, Yuanqi Chang, Mude Hui, Bingjie Xu, Zeyu Zheng, Zirui Wang, Cihang Xie, and Yuyin Zhou. Story-iter: A training-free iterative paradigm for long story visualization, 2024. URL https://arxiv.org/abs/2410.06244.

Chutian Meng, Fan Ma, Chi Zhang, Jiaxu Miao, Yi Yang, and Yueting Zhuang. Logistory: A logic-aware framework for multi-image story visualization, 2026. URL https://arxiv.org/abs/2603.28082.

Haonan Qiu, Menghan Xia, Yong Zhang, Yingqing He, Xintao Wang, Ying Shan, and Ziwei Liu. FreeNoise: Tuning-free longer video diffusion via noise rescheduling. In *International Conference on Learning Representations*, 2024.

Tanzila Rahman, Hsin-Ying Lee, Jian Ren, Sergey Tulyakov, Shweta Mahajan, and Leonid Sigal. Make-a-story: Visual memory conditioned consistent story generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2493–2502, 2023.

Min Shi, Feitong Shen, Zhendong Huang, Jinshu Ye, Wei Chen, Baigui Sun, Dahua Lin, and Kai Chen. AniMaker: An interactive multi-agent video generation studio. In *SIGGRAPH Asia 2025 Conference Papers*, pp. 1–11, 2025. doi: 10.1145/3757377.3764009.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2303.11366.

Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, Paridhi Devi, David Gupta, and Yaniv Taigman. Make-a-video: Text-to-video generation without text-video data. In *International Conference on Learning Representations*, 2023.

Jaskirat Singh, Junshen Kevin Chen, Jonas Kohler, and Michael Cohen. Storybooth: Consistent subject and style driven multi-frame image generation. In *International Conference on Learning Representations*, 2025.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8406–8416, 2025.

Weijia Wu, Zeyu Zhu, and Mike Zheng Shou. Automated movie generation via multi-agent CoT planning. In *International Conference on Learning Representations*, 2026.

Zhifei Xie, Daniel Tang, Dingwei Tan, Jacques Klein, Tegawend F. Bissyand, and Saad Ezzini. DreamFactory: Pioneering multi-scene long video generation with a multi-agent framework, 2024. URL https://arxiv.org/abs/2408.11788.

Xuenan Xu, Jiahao Mei, Chenliang Li, Yuning Wu, Ming Yan, Shaopeng Lai, Ji Zhang, and Mengyue Wu. MM-StoryAgent: Immersive narrated storybook video generation with a multi-agent paradigm across text, image and audio, 2025. URL https://arxiv.org/abs/2503.05242.

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, Da Yin, Yuxuan Zhang, Weihan Wang, Yean Cheng, Bin Xu, Xiaotao Gu, Yuxiao Dong, and Jie Tang. Cogvideox: Text-to-video diffusion models with an expert transformer. In *International Conference on Learning Representations*, 2025.

Kaiwen Zhang, Liming Jiang, Angtian Wang, Jacob Zhiyuan Fang, Tiancheng Zhi, Qing Yan, Hao Kang, Xin Lu, and Xingang Pan. Storymem: Multi-shot long video storytelling with memory, 2025. URL https://arxiv.org/abs/2512.19539.

Xiangqing Zheng, Chengyue Wu, Kehai Chen, and Min Zhang. Locot2v-bench: Benchmarking long-form and complex text-to-video generation, 2026. URL https://arxiv.org/abs/2510.26412.

Jinsong Zhou, Yihua Du, Xinli Xu, Luozhou Wang, Zijie Zhuang, Yehang Zhang, Shuaibo Li, Xiaojun Hu, Bolan Su, and Ying cong Chen. Videomemory: Toward consistent video generation via memory integration, 2026. URL https://arxiv.org/abs/2601.03655.

Yupeng Zhou, Daquan Zhou, Ming-Ming Cheng, Jiashi Feng, and Qibin Hou. Storydiffusion: Consistent self-attention for long-range image and video generation. In *Advances in Neural Information Processing Systems*, 2024.

A Implementation and Execution Records

Persistent story memory represents narrative constraints through assets, shots, entity bindings, visibility requirements, state transitions, spatial plans, and rendering specifications. Identity bindings and inherited relations encode invariants; entry and exit conditions encode transitions; visibility, spatial, and rendering fields constrain realization. Source requirements remain separate from compiled rendering fields. Provenance, evaluation evidence, and attempt history are linked to memory and media versions without becoming narrative facts.

Evaluation issue records contain stable identifiers, target shots, evidence, source versions, categories, severities, and repair suggestions. A scoped revision plan specifies the addressed issues, affected shots, continuity successors, invalidated stages, candidate outputs, reevaluation scope, and resource reservation. Together, these records link a proposed intervention to its supporting evidence and allow its declared scope to be checked against execution.

Before executing a revision, the system preserves the selected memory snapshot and media. After an interruption, it resumes from the latest valid checkpoint, reconciles outstanding reservations, and refreshes invalidated checks. Rejected candidates leave the baseline unchanged. Requests accepted by the backend still consume resources; only reservations for unsubmitted requests are released.

Execution manifests record submitted references, backend task identifiers, stage status, selected candidates, input and output fingerprints, evaluator calls, token use, timings, and termination reasons. These records connect policy decisions to execution outcomes and support recovery after interruption.

A.1 Generation and Refinement Algorithm

Algorithm 1 interleaves generation with text, image, and video evaluation. At each checkpoint, the Revision Agent may return to any supported upstream layer for which the evidence justifies a change. A repair rebuilds existing invalidated outputs up to the current checkpoint; later production uses the selected revised state. Thus, a video-stage plan or reference revision traverses the affected image and video stages again. Both accepted and rejected attempts update the execution records and remaining budget.

Algorithm 1: Adaptive reference policy and evaluation-driven refinement

$C \leftarrow$ PlanBindAndValidate(x); initialize $P = (C, A, I, R)$.
for checkpoint p in text, image, video order **do**
  At image: select reference modes and compile static/dynamic prompts; create required assets and images. At video: render clips from selected references.
  Evaluate the current checkpoint under the selected shot contracts; update issues I.
  while an executable issue remains and refinement limits allow **do**
    Select issue q; compile context $W \leftarrow \mathrm{Compile}(P, q, B)$.
    Revision Agent selects the intervention level, target shots, and fields in plan b.
    Expand dependencies; validate scope and reserve required media requests. On failure, record and defer the issue, then continue to the next issue.
    Apply the candidate change at its selected layer, including upstream layers; rebuild invalidated outputs in production order up to p.
    Recheck affected stages, repair objectives, and continuity boundaries.
    Apply the stage-specific selection rule: adopt the candidate or retain the baseline.
    Update issues, attempts, selected state, and remaining budget.
  end while; advance if stage policy permits, otherwise terminate with the current state.
end for; return selected memory, media, and unresolved issues.

The records link each attempted revision to its issue, selected scope, generated candidate, and acceptance decision. They support local decision analysis alongside generation-based comparisons.

A.2 Decision and Record Schemas

A reference decision records the selected mode, input roles, primary source for each generation stage, static and dynamic descriptions, decision rationale, and backend capabilities. Table 4 summarizes this decision and the records used to interpret and execute revisions.

The Evaluation Agent wraps each raw report with its phase, status, overall score, check details, and gate outcome. Normalization links individual findings to an issue package

$$q = (\mathrm{id}, \mathrm{phase}, \mathrm{target}, \mathrm{class}, \mathrm{category}, \mathrm{severity}, \mathrm{evidence}, \mathrm{confidence}, \mathrm{version}, \mathcal{D}_q),$$

where $\mathcal{D}_q$ records admissible repair depths after classification. Problem descriptions and suggested actions remain attached to their source findings. Class, confidence, and admissible depths are framework annotations added after raw evaluation. Stable source identifiers and a canonical issue key link and merge duplicate findings. Records also retain artifact versions, source scope, classification provenance, and resolution status. Context shots are distinguished from mutation targets, and incomplete evaluator responses are classified as service issues. A suggested action supplies evidence for deliberation without certifying a unique root cause.

Table 4: Semantic fields for reference selection, evaluation, and scoped revision.

Record	Required fields	Interpretation
Reference decision	shot identifier, selected mode, reference roles, primary source, required generations, backend capabilities, decision rationale	Specifies how visual evidence is presented to the generator for one shot.
Evaluation issue record	issue identifier, target shots, evidence, source version, category, severity, repair suggestion	Describes a concern for the refinement policy; later adjudication assigns error status and repair scope.
Scoped revision plan	addressed issues, affected outputs, invalidated stages, candidate outputs, reevaluation scope, resource reservation	Specifies an executable candidate revision while the current baseline remains selected.

Issue records describe concerns; revision plans specify actions. The Revision Agent uses shot contracts, continuity dependencies, and available references to select an intervention level. It cites source issues, names the diagnostic and repair stages, identifies target shots and fields, and specifies executing agents and concrete prompt or contract changes. Before submission, it declares the affected outputs and reserves the required replacement requests. Failure to validate or reserve the plan leaves the baseline unchanged. Artifact fingerprints check that execution respects the declared scope, and request validation checks the selected reference mode and input roles.

Illustrative record. A reference-mismatch concern for shot s_i may identify the submitted shot image, cite evaluator evidence, and suggest that the reference be reconsidered. A D1 plan can edit the image while retaining the reference policy. A D2 plan can change the reference mode or input selection and rebuild dependent media; a D3 plan can revise the story binding before rebuilding the affected outputs. The available memory and backend capabilities can lead to different plans for the same issue. The record preserves the evidence without assigning a single root cause.

A.3 Reference-Mode Compatibility and Compilation

Table 5 specifies the input, control target, and required capability for each director. The Reference Expert selects one alternative for each shot. The compiler validates the selected director contract before submission and rejects a request whose declared inputs or roles do not match the corresponding mode.

The four directors share a plan interface but provide different reference inputs: ShotDirector uses a shot image for composition, AssetDirector uses identity assets, ShotAssetDirector combines both, and PanelDirector supplies ordered panels for within-shot progression.

For each selected mode, the compiler separates the designated static state from the intended dynamic progression. It then attaches the inputs permitted by the mode and records the required generation stages. This contract prevents a continuity frame from silently replacing the primary shot image and prevents an identity bundle from being interpreted as an interaction layout. If a backend lacks the capability required by the selected mode, the policy selects an executable alternative or records a fallback.

Table 5: Compatibility contracts for the four mode-specific reference directors.

Director (mode)	Primary input	Main control	Required capability	Typical limitation
SHOTDIRECTOR (`shot`)	Shot image	Composition	Single-image reference	Limited identity cues
ASSETDIRECTOR (`asset`)	Identity bundle	Identity	Asset reference support	Limited action structure
PANELDIRECTOR (`panel`)	Storyboard sheet	Within-shot progression	Panel/image input	Panel-copying artifacts
SHOTASSET DIRECTOR (`shot_asset`)	Shot image + identity bundle	Composition and identity	Multi-reference support	Higher input complexity

A.4 REPRODUCIBILITY AND REPORTING CHECKLIST

For each story–condition pair, the reporting protocol requires the immutable input story, validated memory snapshot, reference decisions, compiled requests, submitted media manifest, evaluator records, revision plans, candidate and baseline fingerprints, and terminal status. Table 6 specifies the metadata to retrieve from execution records. A parent baseline identifier applies only to frozen-artifact diagnostics; generation-based interventions record their own parent baseline.

Table 6: Minimum per-project reporting fields.

Field group	Required fields
Identity	story ID, repeat ID, condition, parent baseline ID, code revision
Configuration	backend/model versions, resolution, duration, audio mode, prompt versions, seed semantics
Execution	actual shot count, stage status, wall time, provider task IDs, warnings, hard failures
Quality	evaluator outputs, issue records, candidate outcome, story-level and per-shot metrics
Resources	image/video requests, evaluator and LLM calls, consumed/released units, cost, peak concurrency
Integrity	input/output fingerprints, baseline-material fingerprint, checkpoint boundary, failure reason

A recorded seed serves as an experiment identifier unless the selected backend explicitly receives and honors it. RSSA and VBench failures remain in the evaluation record; regeneration follows revision findings.

A.5 FAILURE HANDLING AND PROMOTION OUTCOMES

A proposed revision has three possible outcomes. It is promoted when scope validation and the applicable candidate-selection checks favor the revision. It is rejected when the candidate fails those checks; the candidate remains available for inspection and the baseline stays selected. It is skipped when budget, capability, checkpoint, or scope requirements cannot be met. Reporting these outcomes separately distinguishes failed candidate improvements from revisions that never executed.

Candidate selection and absolute quality-gate passage are distinct. Text selection compares structural validity, contract-grounded findings, and scores. Image and video selection rejects newly introduced hard issues and compares blocking-issue profiles, gate outcomes, and scores. A candidate

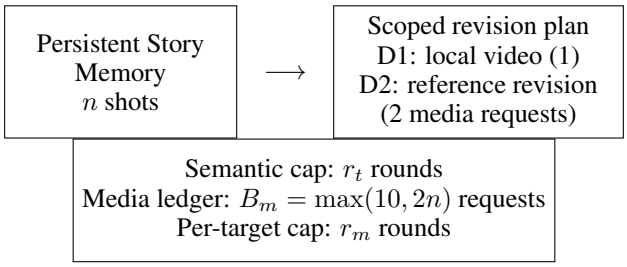

Figure 5: Illustrative scoped repairs and their independent limits. A D2 reference revision may require image and video regeneration. Semantic revisions have a separate round cap, and media revisions reserve their dependent requests from the project ledger. A per-target cap limits repeated repair.

that reduces existing blocking failures without adding a new hard issue may be selected while unresolved findings remain recorded. When no hard issue determines selection, a purely score-based image or video improvement must reach two points. These rules support incremental repair while preserving the best available baseline.

A.6 Media-Request Budget Accounting

For the reported full configurations, the explicit project-level allowance is 10 units; under shot-count scaling, the resulting media budget is $B_m = \max(10, 2n)$, while zero disables media retries. The configurations use $r_t = 2$ semantic-revision rounds, one post-media semantic reopening, and $r_m = 2$ media-repair rounds per shot or asset. A scoped revision plan b requiring $c(b)$ image and video requests must satisfy $c(b) \leq U_m$, where U_m is the remaining allowance (Figure 5). Admission also requires a valid scope, executable actions, and remaining capacity under the independent per-shot or per-asset round cap r_m. The ledger reserves the complete dependency closure before mutation. Submitted requests consume their units even if the candidate is later rejected, while unused reservations are released. Manifests retain consumed, released, and remaining units, skipped revisions, and budget-exhaustion events. Semantic revisions do not consume this media ledger.

A.7 Online Evaluation Protocol

The image and video evaluation methods are adapted from ReelBench. ReelOnce supplies the relevant generation inputs, reference roles, and story or spatial context, links feedback to target artifacts, and requests focused checks of repair objectives during reevaluation. Text evaluation separately checks the story plan and its contracts. RSSA serves as an offline reporting metric, while revisions use online evaluation findings.

For video evaluation, the online loop calls the versioned `video/v1` service. The current service configuration uses `doubao-seed-2-1-pro` with temperature 0.1 and an 8,192-token limit. Each local clip is decoded into eight evenly spaced JPEG frames (quality 85) before being sent to the VLM. The service runs four shot-level checks: instruction alignment (A), character/subject fidelity (B), visual and technical quality (CD), and physical plausibility (E). It also evaluates each adjacent pair for visual continuity (F), cinematic coherence (G), and narrative coherence (H). The raw shot score weights A, B, CD, and E by $(0.15, 0.10, 0.20, 0.15)$; the raw story score additionally weights F, G, and H by $(0.10, 0.15, 0.15)$. Both aggregates divide by the sum of weights for available dimension scores before issue-based adjustments. Cross-shot records are retained separately.

The service returns a 0–100 score and up to three structured issues per dimension. Each issue contains its dimension, severity (critical, major, minor, or info), problem description, evidence, and suggestion. The pipeline maps issue categories to repair actions: reference mismatches request a reference switch and rerender; motion or instruction mismatches request prompt editing and rerendering; severe failures request rerendering; and minor visual defects may request video editing or prompt editing and rerendering. These action proposals are interpreted with the shot contract and story dependencies before a D0–D3 scope is selected.

Severity normalization treats invalid media, missing or identity-breaking primary entities, and violations of required events, prop relations, or spatial continuity as blocking high/critical findings. Material localized failures are medium/major, defects preserving the contract are low/minor, and weak aesthetic observations are informational. Hard/soft classification follows canonical story and media-validity requirements.

For a requested video threshold τ (80 in the standard quality profile), a candidate passes a shot or sequence gate only when its score is at least τ, its evaluator response is complete, and it contains no critical or major issue. Candidate selection can retain an improvement with residual findings under the rules above; gate status and adoption outcome are recorded separately. The active evaluator version, model configuration, threshold, inputs, raw response, and selected candidate are saved in the execution record.

B Ablation Implementation

The reported GPT-5.6-luna ablations share the initial story plan and reuse the corresponding ReelOnce (GPT-5.6-luna) asset images. All three disable online text, image, and video quality checks and method-level revisions, but retain infrastructure retries. ReelOnce w/o Repair retains stateful memory and adaptive references. ReelOnce w/o Mem. additionally disables stateful memory and cross-shot media references. ReelOnce w/o Ref. Policy retains textual state, fixes the reference to a single shot image, and disables cross-shot media references. The latter two conditions are compared with w/o Repair under the same refinement-disabled setting. These comparisons measure complete ablation configurations that include the reused assets.

C RSSA Evaluation Protocol

Reconstructed Story Semantic Alignment (RSSA) measures how faithfully a generated sequence expresses its source story. It first reconstructs visible narrative content without access to the source text, then compares the reconstruction with the original story. A seven-component rubric assesses events, roles, causes, outcomes, and additions. RSSA complements VBench and human evaluation.

Relation to prior evaluation. AniEval combines visual quality, text–video alignment, consistency, and motion measures, including contextual comparisons with neighboring clips (Shi et al., 2025). Its text–story consistency component compares BLIP2-generated descriptions with the input prompt using BLEU. RSSA also uses textual reconstruction as an intermediate representation, but separates observation from story comparison. The first pass reconstructs ordered events and uncertainties without the source story. The second scores this record against the full source story using seven explicit criteria, making both the observed narrative and the evidence for each score available for inspection.

C.1 Inputs and Blind Reconstruction

Each evaluation instance contains the original story and an ordered list of shot identifiers and local video paths. Shots follow narrative playback order. The source must be the story supplied before planning, rather than an expanded script, storyboard description, or generation prompt. Condition labels and descriptive filenames are excluded from the reconstruction prompt. Shot identifiers supplied to reconstruction encode order without descriptive event labels.

For a clip with F_i decoded frames, the default sampler requests four positions,

$$u_{ij} = \mathrm{round}\left(\frac{j(F_i - 1)}{3}\right), \qquad j \in \{0, 1, 2, 3\}. \tag{1}$$

OpenCV seeks to these indices and encodes readable frames as JPEG images at quality 85. Duplicate indices are removed for very short clips. The implementation skips individual frames that cannot be decoded or encoded and fails the story if a clip yields no images. The requested count is therefore four, but the actual count can be smaller. The current runner does not reject a story merely because fewer than four frames are recovered; analyses of short or partially decoded clips must therefore report their realized frame counts separately.

The first model call receives sampled images in clip and frame order, the clip count, and ordered shot identifiers. The source story and all planning and generation text are withheld. The prompt requests a summary, chronological events linked to shots, stable appearance labels for characters, the final observable outcome, and uncertainties. It asks for at least one event per clip and prohibits invented motives or unseen events. The runner checks for a nonempty summary and event list but does not enforce per-shot event coverage. Evaluation uses neither audio nor a continuous video stream.

Reconstruction contract. The released implementation uses a Chinese prompt. Its English structural rendering is:

> You are viewing a multi-shot video sequence. Each shot is represented by 4 uniformly sampled frames. Reconstruct the observable narrative WITHOUT access to the original story or generation prompts.
>
> **Output JSON schema:**

```
{
  "story_summary": "Observable story summary",
  "chronological_events": [
    {
      "order": 1,
      "shot_id": "ordered shot identifier",
      "description":
        "Observable event, action, and outcome",
      "characters": ["Stable appearance labels"],
      "setting": "Observable environment",
      "confidence": "high|medium|low"
    }
  ],
  "characters": [
    {
      "label": "stable appearance label",
      "observed_role": "visible role or relation",
      "observed_actions": ["actions"]
    }
  ],
  "ending": "Final observable state or outcome",
  "uncertainties": [
    "Information not recoverable from the frames"
  ]
}
```

> **Instructions:** Describe at least one chronological-event entry per shot. Use stable appearance labels for recurring characters. Record only visually observable content; do not invent unseen motives, dialogue, or off-screen events. Mark ambiguities explicitly.

C.2 Comparison Rubric and Weighted Score

The second call uses the same model in a separate request containing only the original story and reconstruction JSON. It returns seven component scores with supporting evidence (Table 7). The rubric permits decomposition into shots, visual elaboration, and compatible transitions without penalizing faithful completion that adds no material.

Comparison contract. The second-pass implementation receives the original story and reconstruction JSON. Its English structural rendering is:

> Compare the reconstructed events against the original story across seven dimensions. Each score ranges from 0 to 100 (higher = better adherence).
>
> **Inputs:** Original story text; Reconstruction JSON from first pass.
>
> **Output JSON schema:**

Table 7: RSSA components. Each score lies in $[0, 100]$ and has a positive orientation. The fixed weights prioritize core-event coverage (0.25); character-role fidelity, causal and motivational fidelity, and `no_unsupported_major_additions` each receive 0.15; event order, ending fidelity, and expansion reasonableness each receive 0.10.

Component	Weight	Scoring criterion
Core-event coverage	0.25	Realization of central events without omission, replacement, or reversal.
Event order	0.10	Preservation of the required narrative sequence.
Character-role fidelity	0.15	Preservation of who performs actions and the relevant character relationships.
Causal and motivational fidelity	0.15	Compatibility of observable causes and motives with the source story.
Ending and outcome fidelity	0.10	Preservation of the final state and narrative outcome.
No unsupported major additions	0.15	Avoidance of additions that alter core relationships, causality, theme, or ending.
Expansion reasonableness	0.10	Compatibility and narrative usefulness of elaborations, without penalizing faithful stories that add nothing.

```
{
  "component_scores": {
    "core_event_coverage": 0-100,
    "event_order": 0-100,
    "character_role_fidelity": 0-100,
    "causal_motivational_fidelity": 0-100,
    "ending_outcome_fidelity": 0-100,
    "no_unsupported_major_additions": 0-100,
    "expansion_reasonableness": 0-100
  },
  "component_evidence": {"...": "evidence per component"},
  "matched_core_events": [...],
  "missing_or_changed_core_events": [...],
  "unsupported_major_additions": [...],
  "reasonable_expansions": [...],
  "uncertainty_notes": [...],
  "reasoning_summary": "brief summary"
}
```

Scoring rubric: See Table 7. Faithful completion WITHOUT added material should score high. Compatible elaborations (scene transitions, visual details) are NOT penalized. Uncertain reconstruction details must NOT automatically count as contradictions. Return only component scores; do NOT compute a total (computed programmatically with fixed weights).

Let s_k denote the score for component k on the 0–100 scale and w_k its fixed weight. The reported score is

$$\mathrm{RSSA}(x, V) = \mathrm{round}\left(\sum_{k=1}^{7} w_k s_k,\, 2\right), \qquad \sum_{k=1}^{7} w_k = 1. \tag{2}$$

The implementation requires every component to be numeric and finite, rejects Boolean values, clamps values to $[0, 100]$, and rounds them to two decimal places before aggregation. The model is explicitly instructed not to return a total; the program computes it deterministically. Higher scores indicate greater adherence under the rubric.

The comparison record retains component evidence, matched events, missing or changed events, unsupported additions, reasonable expansions, and uncertainties. Ambiguous reconstruction details remain uncertain unless evidence establishes a contradiction. The two addition-related components

Table 8: RSSA components are reported as mean ± sample standard deviation across three evaluations of fixed generated outputs; Total is reported as a mean only. Cohort sizes are stated in the text; bold and underlined rank the means.

Method	CE	EO	CR	CM
ReelOnce (GPT)	**77.79**±1.29	**90.95**±0.54	<u>81.55</u>±1.62	**57.22**±1.74
ReelOnce (Gemini)	<u>76.52</u>±1.63	<u>90.39</u>±1.02	**81.63**±0.93	<u>57.08</u>±1.85
ReelOnce w/o Repair	76.01±1.04	90.18±2.33	80.66±1.38	54.83±0.78
ReelOnce w/o Ref. Policy	74.97±1.93	88.15±0.74	79.44±2.42	55.88±1.76
ReelOnce w/o Mem.	74.96±1.50	88.37±0.12	81.36±1.05	53.57±0.35
ViMax (Pixar)	60.98±1.64	77.17±0.54	64.85±1.23	40.70±2.51
VideoClaw	40.96±1.72	51.93±0.86	46.95±2.23	22.63±0.88

Method	EF	UA	ER	Total
ReelOnce (GPT)	**78.19**±1.69	**89.34**±3.27	**81.22**±2.58	**78.70**
ReelOnce (Gemini)	<u>77.45</u>±2.39	<u>87.85</u>±1.57	<u>79.98</u>±1.68	<u>77.90</u>
ReelOnce w/o Repair	75.11±2.01	85.15±0.83	78.22±0.25	76.45
ReelOnce w/o Ref. Policy	71.73±1.77	86.24±2.98	78.54±2.61	75.82
ReelOnce w/o Mem.	71.76±1.84	87.84±1.01	79.89±1.23	76.16
ViMax (Pixar)	52.83±2.48	69.70±2.33	64.43±2.71	60.98
VideoClaw	14.76±1.78	34.50±1.54	35.11±1.84	36.03

serve distinct purposes: one penalizes major changes to the story, and the other assesses whether permitted elaborations support the narrative.

C.3 Current Result Coverage

Metric abbreviations. CE, EO, CR, CM, EF, UA, and ER denote core-event coverage, event order, character-role fidelity, causal and motivational fidelity, ending and outcome fidelity, absence of unsupported major additions, and expansion reasonableness. Total is their weighted sum (Table 7). VBench abbreviations SC, BC, AQ, IQ, TF, MS, and DD denote subject consistency, background consistency, aesthetic quality, imaging quality, temporal flickering, motion smoothness, and dynamic degree. Higher TF means less flicker, while higher DD indicates that a larger fraction of videos are classified as dynamic. Bold and underlined mark the highest and second-highest available scores within each column of the relevant comparison table.

Table 8 reports component means and sample standard deviations across three evaluator repeats, with Total shown as a mean only. ReelOnce (GPT-5.6-luna) has the highest Total mean (78.70), followed by ReelOnce (Gemini-3.8-flash) (77.90). The ablation means are 76.45 for w/o Repair, 76.16 for w/o Mem., and 75.82 for w/o Ref. Policy, while ViMax (Pixar) and VideoClaw score 60.98 and 36.03. GPT leads six of seven components, while Gemini narrowly leads character-role fidelity (81.63 versus 81.55). The fixed-reference and repair-disabled configurations differ by 0.63 in Total, but their differing cross-shot media references and cohorts preclude an isolated adaptive-routing estimate. For both ReelOnce backbones, event order is the highest component and causal and motivational fidelity the lowest; sparse frame sampling can miss causal evidence.

C.4 Sensitivity to the VLM Evaluator

We report additional RSSA summaries from a GPT-5.6 Sol evaluator at medium reasoning effort and a Gemini Flash evaluator. The exact Gemini evaluator model version is unavailable in the supplied summary. Table 9 gives the reported means and sample standard deviations for seven components, with Total shown as a mean only. The supplied summaries describe three evaluator repeats, but do not include their individual outputs. The ReelOnce (GPT) and ReelOnce (Gemini) rows use GPT-5.6-luna and Gemini-3.8-flash generator backbones, respectively; evaluator names identify the table

panels. These evaluations use the same 20-story benchmark, but valid scoring coverage varies across configurations and evaluator settings, so the rankings are descriptive.

Comparison across evaluator settings. Both additional summaries place the two complete ReelOnce configurations above the reported ablations and external baselines in mean RSSA Total. With the GPT-5.6 Sol evaluator, the means are 87.63 for ReelOnce (Gemini) and 87.05 for ReelOnce (GPT); with the Gemini Flash evaluator, they are 95.69 and 95.44, respectively. This ordering differs from the primary evaluator, under which GPT scores 78.70 and Gemini scores 77.90. ReelOnce (GPT) covers 18 stories in these additional summaries, compared with 20 in the primary results. The other reported counts are 20 for ReelOnce (Gemini) and fixed references, 18 for w/o Repair and VideoClaw, 20 for ViMax, and 17 for w/o Mem.

C.5 Implementation and Reproducibility

Completion and consistency reporting. Success rate is $100\,N_{\text{complete}}/20$, where N_{complete} counts assigned stories with a complete, evaluable final video. Warnings remain compatible with a complete output. Failed or unfinished stories remain in the denominator but contribute no RSSA score. The reported completion counts are 20 for both ReelOnce configurations and ViMax, 18 for ReelOnce w/o Repair, ReelOnce w/o Ref. Policy, and VideoClaw, and 17 for ReelOnce w/o Mem.

The RSSA summaries cover 20 stories for ReelOnce (GPT-5.6-luna), ReelOnce (Gemini-3.8-flash), and ViMax (Pixar), 18 for ReelOnce w/o Repair, ReelOnce w/o Ref. Policy, and VideoClaw, and 17 for ReelOnce w/o Mem. LoCoT2V CC uses the same 17-story cohort for the two ReelOnce configurations, ViMax, and VideoClaw. Character features are aggregated within each observed role, macro-averaged across roles within a story and then across stories; 95% confidence intervals bootstrap stories as the resampling unit. The corresponding VBench cohorts contain 20 stories and 158 shot clips for ReelOnce (GPT-5.6-luna), 18/141 for ReelOnce w/o Repair, 20/141 for ReelOnce (Gemini-3.8-flash), 17/132 for ReelOnce w/o Mem., 20/126 for ViMax (Pixar), and 18/202 for VideoClaw. VBench story and clip counts for ReelOnce w/o Ref. Policy are unavailable in the supplied summary. The cohorts differ across metric families.

The primary evaluator defaults to `gpt-5.6-luna`, high reasoning effort, and up to 8,192 output tokens per call. It uses an OpenAI-compatible Responses endpoint specified by `OPENAI_BASE_URL`; the endpoint is serialized in the batch output. The same command-line model and reasoning configuration are passed to both reconstruction and comparison calls. Every reported RSSA entry summarizes three evaluator repeats with the video inputs and evaluation configuration held fixed within each evaluator setting. The additional evaluator settings and their separate coverage are reported in Appendix C.4.

For VBench (Huang et al., 2024), we report the seven dimensions separately and define the displayed mean as their unweighted arithmetic mean: $\text{Mean}_7 = \frac{1}{7}\sum_{d\in\{\text{SC,BC,AQ,IQ,TF,MS,DD}\}} v_d$, where v_d is the displayed score on the 0–100 scale. Dynamic degree contributes equally despite measuring motion prevalence rather than stability.

The standalone evaluator loads separate reconstruction and comparison prompt templates and returns one result per story. Its defaults request four frames per clip, high reasoning effort, at most 8,192 output tokens per call, and up to four attempts for request or JSON-parsing failures. Retries use bounded exponential backoff. The batch output records the endpoint, model, reasoning effort, requested frames per video, and weights.

Malformed manifests are rejected before evaluation. Per-story sampling, response-validation, or request failures produce a failed record with a null score and an error description. Failed records remain separate from adherence scores and video regeneration. The runner atomically saves partial results and resumes completed story identifiers. Resumption matches story identifiers rather than input fingerprints. Evaluations with changed videos, prompts, or settings therefore require a new output file.

Aggregation across repeated evaluations. For each component, let z_r denote the batch score from evaluator repeat r, averaged over successfully scored stories. We report the mean and sample

Table 9: RSSA with two additional VLM evaluators. Component cells show mean ± sample standard deviation over evaluator repeats; Total cells show means only. Component abbreviations follow Table 8. Bold and underlined mark the highest and second-highest reported means within each panel and column.

(a) GPT-5.6 Sol evaluator, medium reasoning effort

Method	CE	EO	CR	CM
VideoClaw	48.35±0.39	58.42±0.75	54.73±0.77	33.15±0.67
ViMax (Pixar)	72.30±0.85	85.08±0.79	73.95±1.51	58.73±1.64
ReelOnce w/o Mem.	84.43±0.60	94.45±0.38	90.23±1.46	72.88±2.50
ReelOnce w/o Ref. Policy	**85.03**±1.10	**96.40**±0.23	89.07±0.23	72.43±0.94
ReelOnce w/o Repair	84.02±0.31	95.96±0.31	90.35±0.64	70.98±0.46
ReelOnce (GPT)	84.11±0.82	96.00±0.39	89.44±0.36	72.68±0.71
ReelOnce (Gemini)	83.77±0.96	95.27±0.45	**92.43**±0.38	**73.60**±1.30

(a) GPT-5.6 Sol evaluator, medium reasoning effort

Method	EF	UA	ER	Total
VideoClaw	18.61±0.68	41.13±0.25	42.35±0.95	43.38
ViMax (Pixar)	74.85±1.26	74.32±1.01	74.40±0.28	72.56
ReelOnce w/o Mem.	86.12±1.26	91.35±0.68	89.26±1.22	86.27
ReelOnce w/o Ref. Policy	88.60±2.14	90.03±0.80	87.57±0.55	86.24
ReelOnce w/o Repair	88.31±0.74	91.73±0.98	89.29±0.62	86.32
ReelOnce (GPT)	**91.01**±0.58	93.08±1.86	90.35±1.07	87.05
ReelOnce (Gemini)	88.32±1.30	**95.13**±0.88	**91.50**±0.30	**87.63**

(b) Gemini Flash evaluator

Method	CE	EO	CR	CM
VideoClaw	52.52±0.92	66.06±1.16	58.80±1.51	49.13±1.99
ViMax (Pixar)	77.20±3.05	86.40±1.73	79.28±2.06	74.17±2.88
ReelOnce w/o Mem.	91.31±2.55	97.49±1.60	92.35±2.53	89.13±2.76
ReelOnce w/o Ref. Policy	93.13±1.73	98.37±0.40	94.17±0.60	91.22±1.14
ReelOnce w/o Repair	92.28±0.82	98.61±0.70	94.78±1.57	90.76±1.39
ReelOnce (GPT)	94.65±1.22	**99.46**±0.23	95.00±1.35	**93.28**±1.03
ReelOnce (Gemini)	**94.68**±0.26	99.20±0.38	**96.08**±0.15	92.97±0.18

(b) Gemini Flash evaluator

Method	EF	UA	ER	Total
VideoClaw	39.11±1.10	47.35±1.12	49.53±1.76	51.89
ViMax (Pixar)	79.62±1.68	78.88±2.48	76.55±2.73	78.41
ReelOnce w/o Mem.	93.35±3.50	94.74±2.24	91.62±2.66	92.51
ReelOnce w/o Ref. Policy	95.00±1.71	95.22±0.60	92.78±1.16	93.99
ReelOnce w/o Repair	94.83±1.45	96.08±1.17	93.23±1.57	93.98
ReelOnce (GPT)	96.18±2.47	96.85±0.50	94.37±0.45	95.44
ReelOnce (Gemini)	**96.60**±0.70	**97.33**±0.53	**94.77**±0.08	**95.69**

standard deviation of the three repeat-level scores:

$$\bar{z} = \frac{1}{3}\sum_{r=1}^{3} z_r, \qquad s_z = \sqrt{\frac{1}{2}\sum_{r=1}^{3}(z_r - \bar{z})^2}. \tag{3}$$

Component cells display $\bar{z} \pm s_z$ to two decimal places, while all Total cells display means only. Available component standard deviations quantify evaluator variability on fixed videos; they are neither standard errors nor confidence intervals and do not measure variability across stories or generation runs. Paired comparisons require matched story–repeat outputs and story-level resampling with repeats kept together. Generation and evaluation failures remain separate from low-adherence scores.

C.6 Validity and Limitations

Blind reconstruction limits leakage of intended events into visual observation but remains subject to evaluator bias. Uniform sampling may miss brief actions or distort their apparent order. The runner supplies an ordered shot list and concatenated images without explicit per-image boundary labels, so missing frames can complicate shot attribution. Interpreting event-level evidence therefore requires checking both sampling completeness and reconstructed shot coverage.

Sampled frames provide limited evidence about speech, audio, intent, and implicit causality. Reconstruction errors also affect story comparison. RSSA measures narrative fidelity alongside the visual and motion dimensions of VBench.

D Human Evaluation Protocol

We report five-dimensional ratings adapted from MovieAgent (Wu et al., 2026) and pairwise judgments following the criteria used by ViMax (Huang et al., 2026a). Volunteer raters included students and teachers in artificial intelligence, as well as practitioners in the short-form drama industry. The rating interface uses integer scores from 1 to 5. Table 3 displays the available aggregate means and the GPT-5.6-luna pairwise win rates.

Figure 6 shows the blind rating interface. Reviewers can inspect the source story and four candidate videos labeled A–D, then complete five-dimensional ratings and pairwise preferences without seeing the method identities.

Rating dimensions. *Character consistency* measures the stability of character identity and appearance across shots. *Narrative coherence* measures the internal event flow of the sequence. *Physical law adherence* concerns plausible motion and object interactions. *Script faithfulness* compares visible events, actions, and settings with the source story. *Visual appeal* covers rendering quality and distracting artifacts. The frozen GPT-5.6-luna snapshot uses 153 valid scores per dimension across 20 stories and 15 rater identifiers. The corresponding valid counts are 153–154 for Gemini-3.8-flash and ViMax and 143 for VideoClaw; these counts reflect the frozen live snapshot rather than a balanced fixed panel. The five-dimensional means pool valid individual ratings by method, so stories with more ratings contribute more observations.

Pairwise judgments and coverage. Raters compared candidate videos for cross-scene consistency (CSC), semantic following (SF), and aesthetic quality (AQ). The interface allowed pairwise choices to be inferred from the same five-dimensional ratings and then manually adjusted. The two human summaries can therefore share underlying judgments. The stored summary does not establish how many pairwise entries were entered independently. Table 10 reports the valid judgments in the snapshot. Each criterion cell gives wins, losses, and ties for GPT-5.6-luna against the named opponent.

Win-rate aggregation. For opponent b and criterion c, let $W_{b,c}$, $L_{b,c}$, and $T_{b,c}$ denote wins, losses, and ties. The reported non-tie win rate is $100W_{b,c}/(W_{b,c}+L_{b,c})$. Its denominator excludes ties. The Mean column in Table 3 averages the three criterion-level non-tie win rates before display rounding. Against Gemini-3.8-flash, the current aesthetic-quality comparison contains 45 wins, 40 losses, and 68 ties for GPT-5.6-luna.

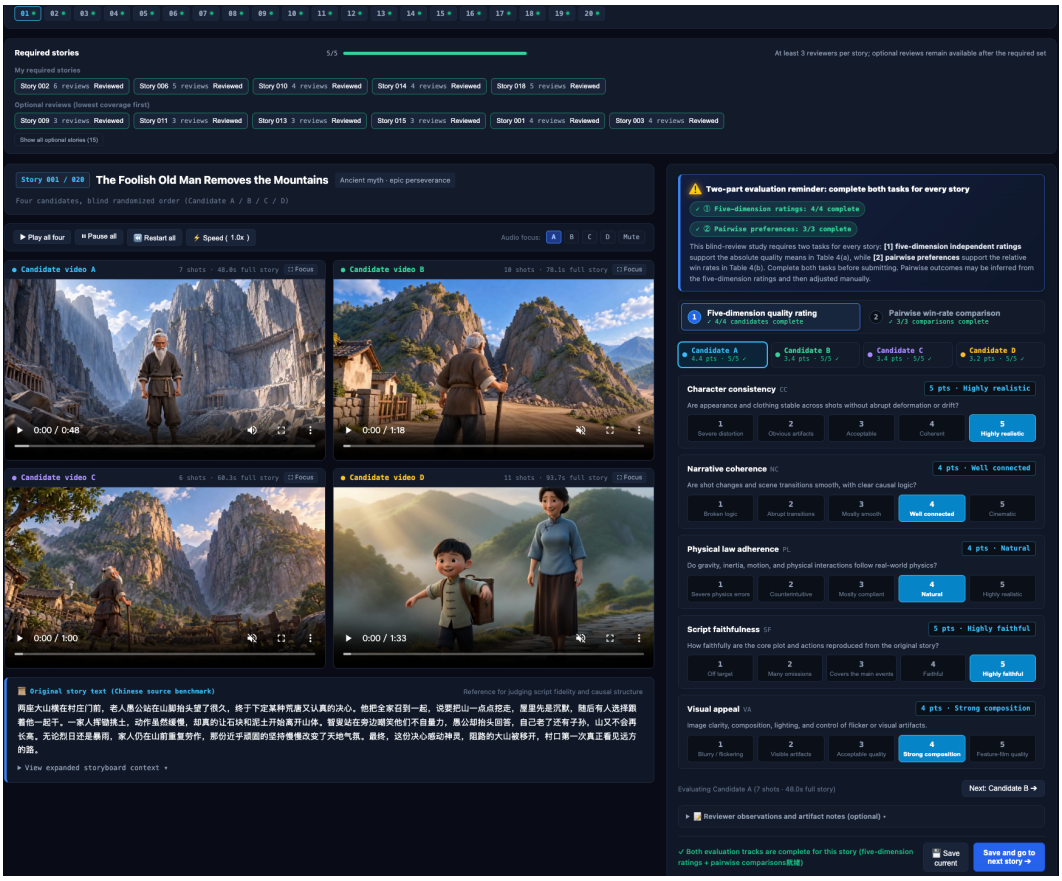

Figure 6: Blind human-evaluation interface. The example view presents four anonymized candidate videos, the source story, five-dimensional rating controls, and access to the pairwise comparison task. Candidate letters do not reveal method identities.

Table 10: GPT-5.6-luna pairwise judgment counts. N is the number of valid judgments per criterion; W/L/T denotes wins, losses, and ties. The three opponents have 14, 11, and 11 valid rater identifiers with at least one criterion entry, respectively.

Opponent	Stories	N	CSC W/L/T	SF W/L/T	AQ W/L/T
Gemini-3.8-flash	20	153	40/38/75	52/47/54	45/40/68
ViMax	20	125	50/13/62	66/12/47	46/27/52
VideoClaw	18	113	96/10/7	91/12/10	49/21/43

E ADDITIONAL QUALITATIVE CASES

Figure 7 shows two text-plan cases, four local image cases, two video cases, and a rejected candidate. These examples illustrate what each evaluation checkpoint can observe and how candidate selection protects previously valid content. The D2 reference-level intervention is illustrated separately in Figure 4; the image rows here show visible before-and-after artifacts, not evidence that a reference mode changed.

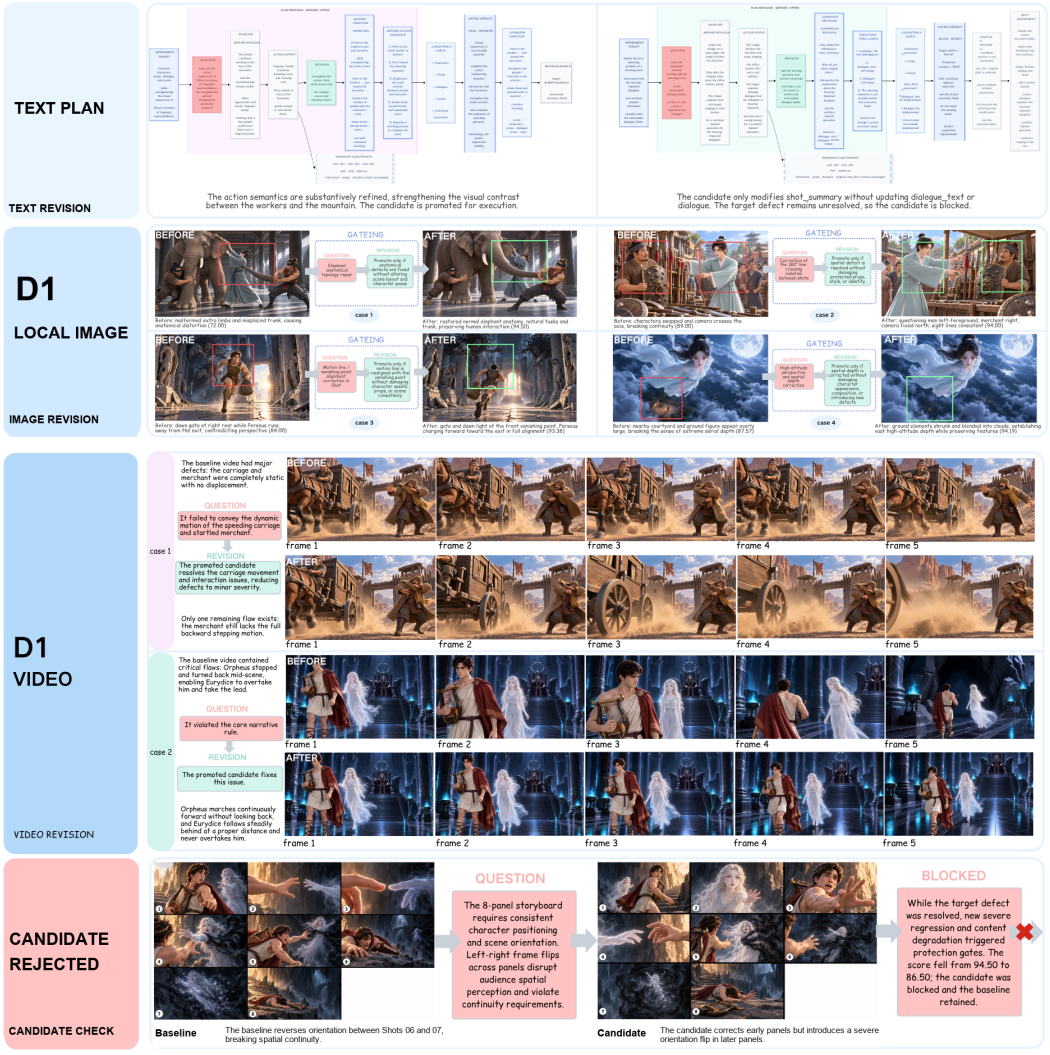

Figure 7: Additional text-plan, local image, and video revision cases, followed by a Promotion Gate rejection. The image rows show corrected artifacts; the related D2 reference-level example is discussed in the main text. Candidate outcomes depend on checks of the target issue and protected content, not on a single visual improvement.

E.1 D3: NARRATIVE AND CONTRACT REVISION

The first text-plan case concerns the contrast between a small group of workers and a mountain that Zhiso mocks them for trying to move. The revision makes that action sequence visually explicit while keeping the characters, props, dialogue, and scene constraints fixed. The revised plan passes the consistency check and is promoted for execution. In the second case, the proposed edit changes the shot summary but leaves the dependent dialogue text and dialogue fields inconsistent with the intended correction. The candidate is blocked at the text stage, before downstream images or video

are generated. Together, the cases show why a planning change must update its dependent contract fields rather than merely improve one description.

E.2 D2: Reference and Spatial Alignment

The Perseus example in Figure 4 illustrates a reference-level spatial correction. In the baseline image, the dawn-lit gate sits behind Perseus even though his action is directed toward the exit. The revised reference aligns the gate, light, and forward motion at the vanishing point; the resulting image also appears in Figure 7. The before-and-after frames show the corrected geometry, while the D2 designation refers to revisiting the upstream reference or spatial input that conditions image generation. A local image comparison by itself does not establish that a reference mode was switched.

E.3 D1: Local Image and Video Correction

At the image checkpoint, the elephant case replaces malformed limbs and a misplaced trunk while preserving the surrounding character poses. The street scene corrects a left–right character swap and camera-axis crossing without changing the protected people and setting. The Perseus image shows the downstream realization of the spatial correction described above. In the aerial case, oversized ground elements are reduced and blended into the clouds to restore the intended high-altitude perspective. These four comparisons show distinct local image defects: anatomy, screen-side continuity, perspective alignment, and scene scale.

At the video checkpoint, the carriage case changes a largely static carriage and merchant into a sequence with visible displacement and interaction. The candidate reduces the main motion defect, although the merchant does not complete the full backward step. In the Orpheus case, the baseline depicts him stopping and turning back, which lets Eurydice overtake him; the revision instead keeps him moving forward with Eurydice following at a distance. The cases distinguish a partial correction with a recorded residual flaw from a correction of the targeted narrative action.

E.4 Promotion Gate: Rejected Candidate

The final panel shows an image-triggered spatial-plan revision for an eight-panel storyboard. The baseline reverses orientation between later shots. A candidate repairs the early panels but introduces a new severe left–right reversal farther along the sequence, and the displayed score falls from 94.50 to 86.50. The Promotion Gate therefore rejects the candidate and retains the baseline. This is a local selection outcome: fixing the original target does not justify promotion when the revision damages a protected continuity requirement.

F PANELDIRECTOR WITHIN THE REFERENCE-DIRECTOR FAMILY

PANELDIRECTOR is one of four peer directors selected by the Reference Expert. All four translate shot contracts into visual inputs and prompts. PANELDIRECTOR additionally divides a continuous shot into timed action beats and selects storyboard density and duration. We analyze archived decisions to characterize their repeatability, output distributions, and agreement across planners. Assessing their effect on video quality requires linked downstream evaluations.

F.1 ARCHIVE AND PROTOCOLS

The main archive contains 140 Chinese shot descriptions spanning fables, mythology, and short narratives. These include 131 distinct raw scripts and 130 after removing leading story-title prefixes and outer whitespace. We report entry-level statistics and check their sensitivity by retaining the first occurrence of each normalized script. Repeated calls measure variation within an entry; they do not increase the number of independent stories.

The joint protocol requests a panel count p, duration d, and rationale, targeting ten samples per entry and model. Gemini 3 Flash Preview, Claude Sonnet 4.6, and GPT-5.5 each have 1,400 samples. Claude Opus 4.7 has 823 samples: 77 entries have ten, 13 have one to eight, and 50 have none. Historical Qwen3.5 4B records cover all 140 entries but fall outside the final selected model set in the archive. Model names identify stored labels rather than verified provider snapshots.

The fixed-duration protocol contains five GPT-5.5 calls per entry, totaling 700 samples. Durations follow the saved GPT recommendation for each entry in the joint protocol: 5, 6, 7, and 8 seconds occur in 3, 20, 113, and 4 entries. An interrupted 87-sample batch is excluded. The archive supports reproducing the reported statistics, but replicating the calls requires the missing decision prompts, temperatures, seeds, and request-level model versions.

F.2 REPEATED-CALL STABILITY AND OUTPUT DISTRIBUTIONS

For entry i, model m, and recorded sample r, write the decision as (p_{imr}, d_{imr}) and the sample count as R_{im}. Modal shares measure concentration:

$$S_{im}^{\text{joint}} = \max_{p,d} \frac{1}{R_{im}} \sum_{r=1}^{R_{im}} \mathbf{1}[(p_{imr}, d_{imr}) = (p, d)], \tag{4}$$

$$S_{im}^{\text{panel}} = \max_{p} \frac{1}{R_{im}} \sum_{r=1}^{R_{im}} \mathbf{1}[p_{imr} = p]. \tag{5}$$

Table 11 averages these shares over entries with all ten samples. Sonnet has the highest repeatability among the three selected models with complete 140-entry coverage. Modal shares quantify repeated-call consistency; evaluating planning accuracy requires independent annotations.

Table 11: Joint-protocol repeatability. Shares are means of entry-level modal frequencies. The final column counts entries whose ten joint decisions all match. Opus uses only its 77 complete entries; Qwen is a historical record.

Archive model label	Complete entries	Joint share	Panel share	All 10 agree
Gemini 3 Flash Preview	140	69.86%	72.29%	13/140
Claude Sonnet 4.6	140	89.29%	94.00%	90/140
GPT-5.5	140	77.43%	81.21%	40/140
Claude Opus 4.7	77	83.51%	88.96%	40/77
Qwen3.5 4B (historical)	140	100.00%	100.00%	140/140

Figure 8 shows different output preferences across models. In the historical Qwen records, all ten samples for each entry have identical panel counts, durations, and rationales. The entire batch contains only three pairs: $(6, 5)$, $(8, 6)$, and $(9, 7)$. Sampling and cache logs are needed to interpret this perfect repeatability.

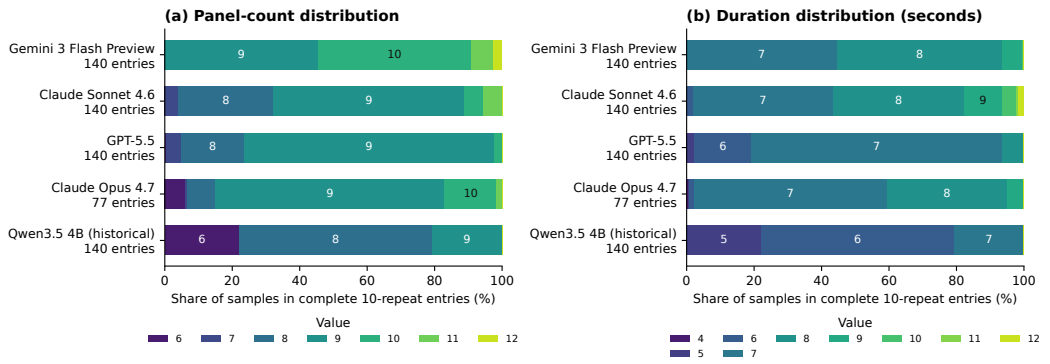

Figure 8: Panel-count and duration distributions from complete ten-repeat entries. Each model contributes 1,400 samples except Opus, which contributes 770. Segment labels identify panel counts or durations, not quality scores.

F.3 Fixed-Duration Decisions

With duration fixed, 109/140 entries (77.86%) return the same panel count in all five rounds. The mean panel-mode share is 93.14%; maximum vote counts of four, three, and two occur in 17, 11, and 3 entries. Across the 700 samples, 6, 7, 8, and 9 panels occur 1, 26, 40, and 633 times. Nine-panel outputs therefore account for 90.43% of calls (Figure 9). Two entries have tied 8/9-panel modes; both retain the archived recommendation of eight.

The panel-mode share rises from 81.21% in the joint protocol to 93.14% with fixed duration. However, the protocols also differ in their prompts and repeat counts, and fixed durations are chosen by the same planner. Because modal share itself depends on repeat count, this difference cannot be attributed to duration alone. The preference for nine panels further motivates testing whether density choices respond to story content.

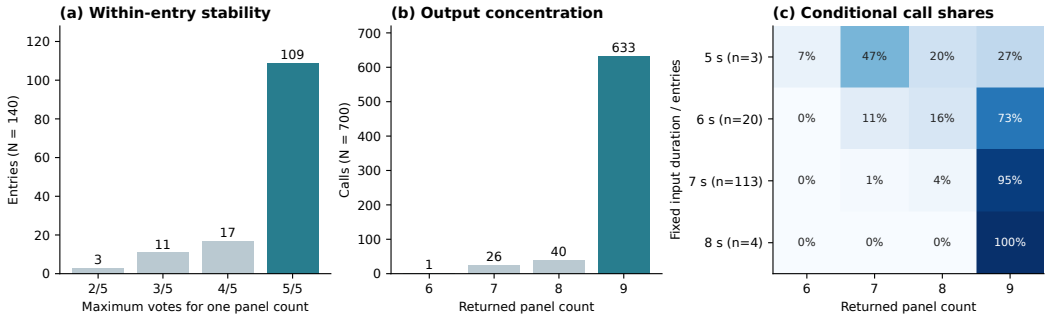

Figure 9: Fixed-duration GPT-5.5 decisions: maximum vote count per five-round entry, panel-count distribution across calls, and within-duration shares. Duration groups contain different scripts and are highly unequal; they are not matched interventions on the same input.

Retaining one entry per normalized script yields a mean modal share of 94.15%, full agreement in 105/130 entries (80.77%), and nine-panel outputs in 595/650 calls (91.54%). Thus, concentration persists after deduplication, though related stories may remain dependent. In each of the three least stable entries, the most frequent count receives only two votes. Their rationales provide material for semantic review, while correctness requires independent labels.

F.4 Agreement Across Planners

We verify that each saved joint recommendation belongs to the modal set of its recorded samples and retain the tie resolution recorded in the archive. Complete entries have tied joint modes for 7 Gemini, 3 Sonnet, 5 GPT, and 1 Opus cases. Across the 140-entry overlap, exact agreement on both variables is 19/140 (13.57%) for Gemini–Sonnet, 41/140 (29.29%) for Gemini–GPT, and 21/140 (15.00%) for Sonnet–GPT. Their mean absolute panel-count differences are 0.84, 0.79, and 0.46; duration differences are 0.64, 0.71, and 0.99 seconds.

Comparisons involving Opus use 77 shared entries with complete samples (Figure 10). All four selected models agree exactly on 1/77 entries (1.30%); including partial entries with available recommendations gives 1/90 (1.11%). Each comparison uses its corresponding shared-entry count as the denominator. We omit archived weighted proximity scores because their implementation and parameters are unavailable.

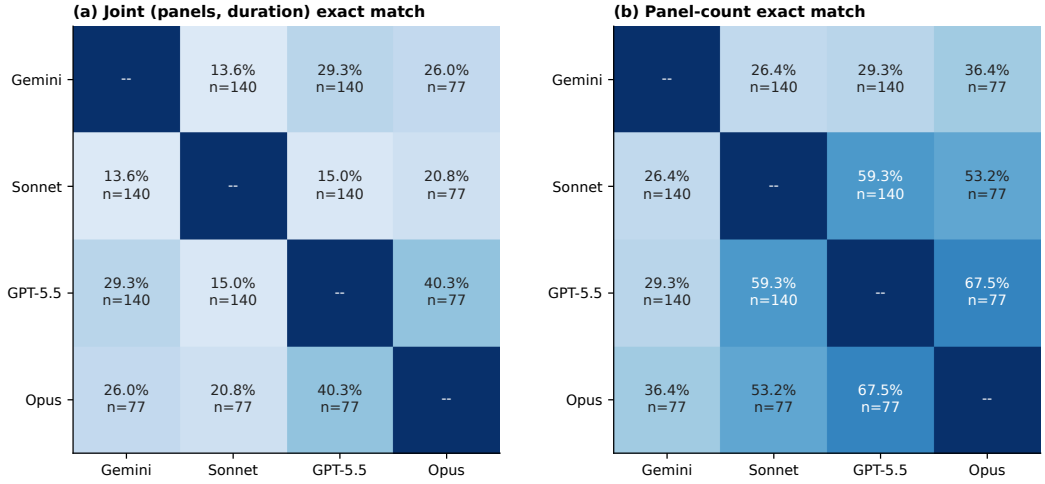

Figure 10: Agreement between saved joint recommendations: exact matches on both variables (left) and on their panel-count components (right). Each off-diagonal cell states its own denominator; both models have ten samples for every included entry. The panel component of a joint mode can differ from the marginal panel mode. Agreement is not a ground-truth quality measure.

F.5 Prompt Compilation and Archived Media

Recovered candidate plans compile a shot summary, reference roles, and a panel plan into generation prompts. Storyboard prompts specify visual states, framing, layout, and numbering; video prompts specify timing, motion, and continuity between beats. The six-panel warehouse example decomposes its action into pressure, attack, block, dodge, counterattack, and retreat. Its compilation structure is summarized below:

> **Storyboard:** shot and style; identity and environment references; panel count, layout, reading order; per-panel beat, visual state, and camera.
> **Video:** the same context; total duration and panel intervals; per-panel motion, continuity from the previous beat, and transition to the next.

These recovered fields document the conversion from plans to image and video prompts, although the original decision prompt is unavailable. Later templates introduce additional layout constraints and require continuous, non-overlapping timing. Their effects remain untested in the archive.

Figure 11 reproduces the archived 6-, 9-, and 15-panel warehouse storyboards. Submitted task records identify GPT Image 2 and Seedance 2.0; all three videos request six seconds and have measured container durations of approximately 6.06 seconds. Media are associated with submitted requests rather than later editable snapshots. For example, a later snapshot of an additional automatic candidate specifies nine panels and eight seconds, while its saved video request uses a different nine-panel plan at six seconds.



Figure 11: Archived 6-, 9-, and 15-panel warehouse-action storyboards, each associated with a requested six-second video. These saved outputs illustrate realized conditions, not a ranking of visual quality or a density–quality curve.

Scope of the analysis. The archive supports analysis of planning decisions and inspection of saved media. Candidate records lack downstream metric vectors and evaluation task identifiers; a separate five-clip evaluation is excluded because it cannot be linked to this batch. Independent annotations and matched video evaluations are needed to assess planning accuracy, density choices, and prompt effects.