CINEFORGE: SELF-IMPROVING AGENTS FOR LONG-HORIZON VIDEO GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-horizon story-driven video generation requires coordinating narrative decomposition, state tracking, shot design, prompting, rendering, and revision across interdependent scenes. Adaptive video systems primarily refine requests or reusable skills, leaving recurring failures disconnected from persistent, stage-targeted improvements across stories. We introduce **CineForge**, a self-evolving video-production agent framework coupling **CineForge-Produce** for generation with **CineForge-Evolve** for cross-story policy evolution. CineForge-Produce structures stories into typed narrative, character, spatial, and cinematic states to coordinate asset and clip generation, recording canonical production trajectories. CineForge-Evolve applies **Case-to-Pattern-to-Policy Evolution (CPPE)** to review trajectory evidence, consolidate recurrent findings into bounded stage-local patches, and deploy validated updates through structural replay and confidence-controlled paired evaluation. To measure complete story realization, **CineScope** combines 100-script **CineScope-Data** with human-aligned, multiscale **CineScope-Metric** covering causal state, directorial orchestration, pacing and resource allocation, and character arc. Across CineScope-Data and two public benchmarks, the evolved policy improves CineScope-Metric from 4.02 to 4.38, outperforms three long-video baselines with consistent gains under ScriptAgent, and reduces review LLM calls by 37.0% on new stories. Results establish production trajectories as actionable experience for cumulative improvement across long-form storytelling.

1 INTRODUCTION

Recent advances in video foundation models have substantially improved visual fidelity, motion realism, instruction following, and temporal extension (Kim et al., 2024; Lu et al., 2024; Guo et al., 2025; Kara et al., 2025). These capabilities are moving video generation toward long-form story-driven production, a long-horizon setting where a complete narrative is realized through interdependent scenes and shots. Success in this setting depends on preserving event coverage, character identity and state, spatial relations, directorial intent, and pacing throughout the full narrative horizon. This paper studies how a persistent video-production agent can learn from completed productions and convert prior experience into systematic improvements for future stories.

Agentic video systems combine hierarchical planning, memory, multi-agent coordination, iterative critique, and consistency control to manage complex long-horizon generation workflows (Lin et al., 2023; Wang et al., 2024; Wu et al., 2025b; Long et al., 2026). Adaptive approaches operate at distinct scopes: GenMAC iteratively repairs the current generation, while VideoWeaver evolves reusable skills from prior videos (Huang et al., 2024a; Wei et al., 2026a). Figure 1 presents these as instance-level repair and skill-level optimization, respectively. Long-form production requires a further policy-level capability: persistent production-policy evolution across stories, since a visible defect may originate in narrative decomposition, state tracking, shot design, prompt rendering, or backend execution. This requires production-stage traceability, cross-case recurrence modeling, stage-targeted policy updates, and deployment gates that preserve reliable behavior.

We introduce **CineForge**, a self-evolving agent framework organizing the full production lifecycle through **CineForge-Produce** and **CineForge-Evolve** for long-horizon story-driven video production and persistent cross-story policy improvement. CineForge-Produce decomposes each source story into checkable narrative units and typed episode, scene, shot, character, and spatial states. These

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

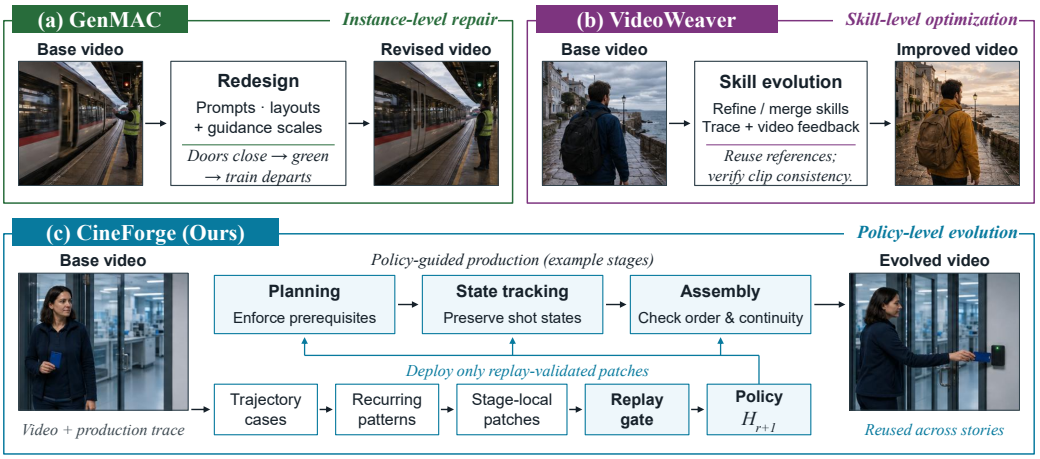

Figure 1: Comparison of adaptation scopes: instance-level repair (GenMAC), skill-level optimization (VideoWeaver), and our replay-gated policy-level evolution (CineForge).

shared states coordinate asset creation, prompt construction, clip generation, composition, and bounded revision to produce complete videos, with intermediate decisions and outcomes recorded as canonical production trajectories. CineForge-Evolve applies **Case-to-Pattern-to-Policy Evolution (CPPE)** to review pipeline artifacts, local clips, and complete episodes, linking failures to the earliest production stages supported by trajectory evidence. Recurrent findings across stories become patterns and bounded stage-targeted policy patches. Each candidate is replayed against the incumbent policy through structural checks and confidence-controlled paired evaluations, admitting validated knowledge into the next policy version to guide subsequent stories.

Reliable progress in long-form video production requires a framework for evaluating story-level quality across complete videos. We therefore introduce **CineScope**, comprising **CineScope-Data** and **CineScope-Metric**. CineScope-Data contains 100 long-form benchmark scripts curated across diverse genres, narrative structures, character configurations, temporal spans, and scene-transition patterns, providing evaluation inputs that exercise long-range story dependencies. CineScope-Metric evaluates complete videos through multiscale video evidence and twenty human-aligned criteria grouped into Causal State, Directorial Orchestration, Pacing and Resource Allocation, and Character Arc. Calibrated against expert judgments and frozen for system evaluation, CineScope supports measuring story realization quality and tracking improvements across evolution rounds.

Evaluation has two stages. We first assess CineScope-Metric against expert judgments. We then freeze it for CineScope-Data, AnimeShooter, and ViStoryBench evaluation (Qiu et al., 2025; Zhuang et al., 2026), complemented by story-level ScriptAgent and perceptual VBench-Long assessments. Spearman correlation reaches 0.710 and Bradley-Terry agreement 0.810. Both exceed 0.400–0.500 plateaus on complex video tasks. The evolved policy reaches 4.38 versus Round 0's 4.02 (+8.80%), outperforming three long-video baselines with consistent ScriptAgent gains. Despite initial-policy plateaus and narrative complexity, this indicates substantial coherence improvement. Evolution sustains gains: Round 3 exceeds Round 0 by 0.355 in CineScope and 0.188 in ScriptAgent. On new stories, scores improve; Round 1 reduces review LLM calls by 23.7% and detected issues by 30.8%. These workload reductions under a frozen audit configuration establish neither lower end-to-end cost nor higher first-pass yield. Results demonstrate cumulative, transferable policy evolution with reduced review workload. Our contributions are:

- We introduce CineForge, which unifies CineForge-Produce and CineForge-Evolve to connect structured long-horizon video production with persistent policy improvement across stories.
- We develop CPPE to transform production-trajectory findings into recurrent patterns and bounded stage-local patches through replay-gated deployment and confidence-controlled analysis.
- We introduce CineScope, combining a 100-script CineScope-Data suite with the human-aligned, multiscale CineScope-Metric for evaluating complete story-driven videos.
- Experiments across CineScope and public benchmarks demonstrate strong video quality, cumulative evolution gains, transfer to new stories, and reduced production-review workload.

2 PRELIMINARIES

Task, Production Trace, and Persistent Policy. Let r index a policy version and collection round, and i index a story. For $x_{r,i} \sim \mathcal{D}$, CineForge-Produce executes H_r with randomness $\zeta_{r,i}$, yielding

$$(V_{r,i}, Z_{r,i}) = \text{CineForge-Produce}(x_{r,i}; H_r, \zeta_{r,i}),$$

where V is the complete video and Z its canonical trajectory. Episodes are internal text/production chapters whose clips form one concatenated video, not separate evaluation units. Collection, fresh admission, and held-out evaluation use protocol-specific samples; $\mathcal{D}$ denotes the target task distribution. The trajectory records stage inputs and outputs, validator and Review-Rewrite outcomes, provenance, and dependencies among plans, states, prompts, assets, and clips.

Persistent policy $H_r = (\mathcal{C}_{stage}, \mathcal{R}_{prompt}, \Theta_{\text{prod}}, \mathcal{R}_{review}, \mathcal{P}_{style})$ comprises stage contracts, prompt-rendering rules, production thresholds and retry budgets, review-routing rules, and bounded style profiles. In-run repair changes artifacts and logs outcomes in Z; admitted patches change H for subsequent stories. Recurrence thresholds, replay construction, confidence budgets, and acceptance margins are fixed controls per certified sequence, not editable production parameters.

Cases, Patterns, and Bounded Updates. A finding ℓ records an observed symptom, its unit, evidence, and candidate causes. Diagnosis produces a standard issue e with the earliest evidence-supported stage σ, normalized failure family g, repair specification, confidence, lineage, and repair status. One production contributes one case κ, containing its story, policy, trajectory, video, issue set, and repair actions and outcomes. Complete record schemas appear in App. B.11.

A pattern $P = (\sigma, g, \chi, E_P)$ groups cases by stage, failure family, and context signature χ, with supporting case set E_P. Eligibility requires a fixed threshold $|E_P| \geq k_g$ over distinct story-level opportunities supported by sufficiently confident, policy-editable issues. An eligible pattern yields a bounded stage-local patch Δ: a typed policy target, allowed edit operation and payload, risk level ρ, and rollback metadata. Applying it gives $H' = H \oplus \Delta$; the rollback metadata identify the incumbent.

Replay Regimes and Evaluation Boundary. Every candidate undergoes structural checks; exact replay is valid only when all affected transformations after a declared cache boundary are deterministic. Upstream artifacts and validators are fixed, while editable rules remain in H. A patch invoking a new LLM, image, or video-generation call additionally requires repeated paired stochastic evaluation. Cache and trace definitions are in Apps. B.9 and B.13.

The structural margin $Y_{H,\Delta}$ combines validity Q, protected-validator regression D, and off-slice change U. These internal admission quantities differ from frozen final-video utility $J_j(H) = \mathbb{E}_{x\sim\mathcal{D},\zeta}[j(V_H(x;\zeta))]$, measured on held-out outputs after the policy is fixed. CineScope-Metric and ScriptAgent are outcome evaluators, not patch-admission signals.

3 CINEFORGE FRAMEWORK

3.1 OVERVIEW

CineForge couples trajectory-aware production with cross-story evolution (Figure 2). Produce returns videos and trajectories; Evolve combines the collection batch D_r with CaseMemory M_r, induces patterns $\mathcal{B}_r$, and compiles a finite candidate set $\mathcal{C}_r$. At most one admitted patch $\Delta_r^\star$ is deployed:

$$H_{r+1} = \begin{cases} H_r \oplus \Delta_r^\star, & \Delta_r^\star \neq \varnothing, \\ H_r, & \text{otherwise.} \end{cases}$$

The full interface is in App. B.11; policy components and fixed evolution controls follow Sec. 2. **An artifact-based walkthrough of production, cross-story rule formation, and later rule reuse appears in App. A.1 (Figure 4).**

3.2 CINEFORGE-PRODUCE: TRAJECTORY-AWARE LONG-FORM PRODUCTION

Given a story x, Produce constructs a typed state Σ from narrative atoms organized into episode, scene, and shot plans. Coverage maps link atoms to shots; character and scene ledgers propagate identity,

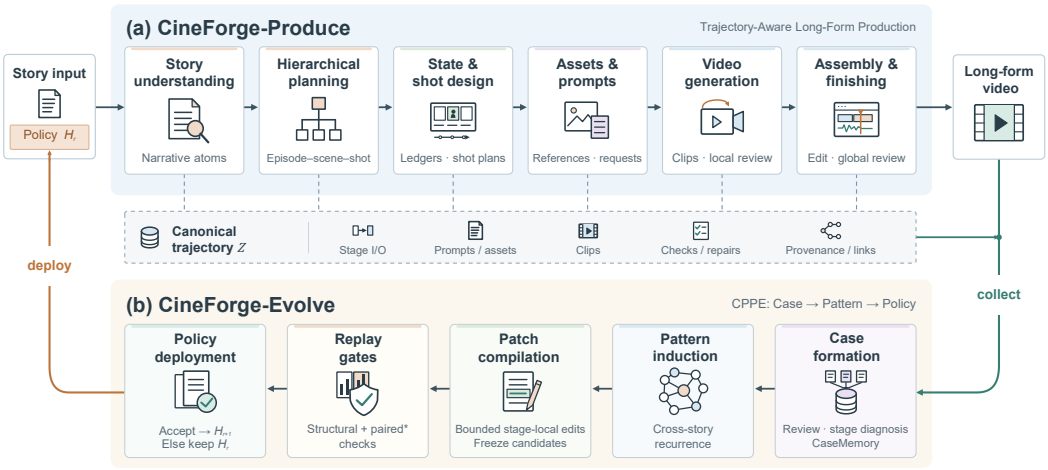

Figure 2: CineForge-Produce generates story videos and canonical trajectories under current policy. CineForge-Evolve converts recurring trajectory evidence into bounded, replay-validated updates for subsequent productions.

spatial state, visual anchors, and transitions. These dependencies support pre-rendering coverage and continuity checks. The policy coordinates reusable assets, provider-specific prompts, clips, anchor propagation, and composition. Deterministic contracts validate schemas, identifiers, coverage, state consistency, durations, and provider constraints. Bounded Review-Rewrite repairs affected artifacts and logs the outcome. The returned canonical trajectory retains stage decisions, artifacts, checks, repairs, provenance, and dependencies for subsequent evidence-supported localization. App. B.11 specifies states and the production graph; App. B.12 gives the procedure.

3.3 CineForge-Evolve: Case-to-Pattern-to-Policy Evolution

Trajectory-grounded case formation. CPPE reviews completed productions through pipeline checks of contracts, semantic propagation, assets, and cross-segment state; local-video checks of prompts and clips; and global-video comparisons of adapted scripts with composed episodes. Following trajectory dependencies, CPPE assigns the earliest evidence-supported stage σ, or `unknown` for ambiguous evidence. Cross-granularity findings are deduplicated and merged before a stage-specific critic normalizes families and proposes bounded repairs with confidence and lineage. CaseMemory retains full cases and repair outcomes. Only sufficiently confident, policy-editable unresolved or `stuck` issues support pattern induction; localization is an auditable hypothesis, not a proven cause.

Cross-story abstraction and bounded patches. CPPE groups supporting cases by (σ, g, χ) and admits a pattern into $\mathcal{B}_r$ when $|E_P| \geq k_g$ across distinct story-level opportunities. Thresholds and opportunity definitions remain fixed; Lemma 1 states the recurrence conditions. Each pattern is compiled into typed, stage-local edits, whose union forms $\mathcal{C}_r$. For example, recurrent spatial inconsistencies induce ledger rules. Patches retain the typed schema of Sec. 2 and an audited dependency slice; component details and examples appear in Apps. B.7 and B.11.

Replay-gated policy versioning. CPPE freezes the complete candidate class, selection rule, and tie-breaker before accessing fresh admission stories R_r. Motivating failures and near misses form the diagnostic set R_r^{diag}; structural checks compare each candidate with H_r on $R_r^{\text{det}} = R_r^{\text{diag}} \cup R_r$. Every protected metric in the risk-level guard set $\mathcal{G}_\rho$ must satisfy

$$S_j(H_r \oplus \Delta; x) \geq S_j(H_r; x) - \epsilon_{j,\rho} \quad \forall x \in R_r^{\text{det}},\ S_j \in \mathcal{G}_\rho.$$

Tolerances are predeclared and zero for low-risk structural updates. The fresh subset separately supplies the statistical structural margin below; diagnostic checks do not count toward its sample size. Stochastic-impact patches also require repeated paired generation and scoring on R_r^{vis}:

$$\widehat{\Delta A}_r \geq m_{V,r} + 2W_{\mathcal{A}}\sqrt{\frac{\log(2(K_r+1)/\delta_r^V)}{2n_{V,r}}},$$

where $K_r = |\mathcal{C}_r|$, $n_{V,r}$ counts independent paired stories, $W_{\mathcal{A}}$ is the admission-score weight sum, and δ_r^V is the stochastic confidence budget. This lower-bound condition concerns CPPE admission, not CineScope quality (Proposition 1). The frozen rule selects at most one candidate passing all applicable checks; otherwise H is unchanged. Each decision and the aggregate gate statistics are recorded in an immutable audit; replay identifiers and artifacts do not enter CaseMemory. Full risk, side-effect, and deployment checks remain those of Apps. B.9–B.12.

3.4 Measurable Evolution and Evaluation Boundary

Let $\widehat{G}_r(\Delta)$ be the fresh-story mean of the risk-adjusted structural margin $Y_{H_r,\Delta} \in [-C, C]$, and $G_r(\Delta)$ its conditional population expectation. With structural welfare $Q \in [0,1]$, normalized guard debt $D \in [0,1]$, off-slice change $U \in [0,1]$, and fixed $\beta, \gamma \geq 0$, the margin is

$$Y_{H,\Delta} = Q(T^d_{H\oplus\Delta}) - Q(T^d_H) - \beta D_{H,\Delta} - \gamma U_{H,\Delta}, \qquad C = 1 + \beta + \gamma,$$

where T^d_H denotes the deterministic trace under a declared fixed replay context (App. B.13). Thus, a structural gain is penalized for protected regressions and changes outside the audited dependency slice. Fixed-item checks compare protected validators on the declared replay items; the fresh-story certificate adds a population statement rather than replacing those checks. Candidate construction may adapt across rounds, but the complete candidate class and its selection rule must be fixed before each fresh admission set is revealed. In addition to fixed-item guards, structural admission requires

$$\widehat{G}_r(\Delta_r^\star) \geq m_r + C\sqrt{\frac{2\log(|\mathcal{C}_r|/\delta_r^R)}{n_{R,r}}},$$

where $n_{R,r}$ counts fresh stories and δ_r^R is the round's structural confidence budget. For a fixed target distribution $\mathcal{D}$, conditionally i.i.d. fresh replay after freezing each finite candidate class, and $\sum_r \delta_r^R \leq \delta_R$, Theorem 1 guarantees $G_r(\Delta_r^\star) \geq m_r$ simultaneously across accepted rounds with probability at least $1 - \delta_R$. Across adaptive rounds, stochastic budgets satisfy $\sum_r \delta_r^V \leq \delta_V$ and $\delta_R + \delta_V \leq \delta$. Apps. B.13 and B.10 give full definitions, theorem and cumulative proof, and stochastic, recurrence, and transfer results. These establish neither causal attribution nor video improvement. Frozen outcome evaluators remain outside updating and selection and measure $J(H)$ (Sec. 5.1).

4 CineScope Benchmark

4.1 CineScope-Data: A Diverse Collection of Long-Form Stories

CineScope combines **CineScope-Data**, a suite of 100 long-form scripts, with **CineScope-Metric**, a frozen final-output measure of story realization. The scripts span genres, narrative structures, character configurations, temporal spans, and scene transitions, exercising cross-scene state continuity, causality, pacing, and character progression. All 100 are used, with 60 for evolution collection and the remaining 40 combined with 40 AnimeShooter and 40 ViStoryBench scripts (Qiu et al., 2025; Zhuang et al., 2026). This non-collection pool is split into disjoint human-alignment, replay-admission, and system-test roles (Sec. 5.1). Replay is revealed only after candidate freezing; system-test stories serve generator, optimizer, and round-wise unseen comparisons. Across sources, the item is a complete script and concatenated video, with the story as the statistical unit. App. B.1.1 details construction, source allocations, identifiers, and normalization; App. A.3 compares existing benchmarks.

4.2 CineScope-Metric: Global Evaluation for Long-Form Video

For source narrative x, permitted task package r, and complete video V, CineScope-Metric evaluates $q = (x, r, V)$ and returns $F_{\text{video}}(q) = (\mathbf{s}^S, \mathbf{s}^G, \mathcal{E}_V)$, where $\mathbf{s}^S \in [0,5]^{20}$ contains twenty criterion scores, $\mathbf{s}^G \in [0,5]^4$ four Global aggregates, and $\mathcal{E}_V$ temporal evidence, eligibility, and confidence. Each Global dimension aggregates five criteria:

- **Causal State (CS):** coherent causes, consequences, and state changes across scenes.
- **Directorial Orchestration (DO):** shots, camera choices, and transitions serving narrative intent.
- **Pacing and Resource Allocation (PR):** screen time, emphasis, and rhythm matching narrative importance.

Table 1: Alignment between CineScope-Metric and held-out expert judgments at the overall and four Global-dimension levels.

CineScope-Metric scope	Spearman ρ [95% CI] ↑	BT pair agreement [95% CI] ↑	W_1 similarity [95% CI] ↑	JS similarity [95% CI] ↑
Overall	0.710 [0.550, 0.826]	0.810 [0.698, 0.905]	0.949 [0.926, 0.967]	0.932 [0.823, 0.958]
Causal State (CS)	0.621 [0.463, 0.726]	0.667 [0.524, 0.794]	0.910 [0.883, 0.936]	0.887 [0.797, 0.922]
Directorial Orchestration (DO)	0.522 [0.342, 0.674]	0.698 [0.571, 0.810]	0.955 [0.921, 0.972]	0.957 [0.859, 0.972]
Pacing and Resource Allocation (PR)	0.607 [0.432, 0.742]	0.730 [0.635, 0.825]	0.916 [0.885, 0.941]	0.863 [0.744, 0.901]
Character Arc (CA)	0.680 [0.512, 0.806]	0.730 [0.619, 0.841]	0.925 [0.892, 0.947]	0.902 [0.813, 0.937]

- **Character Arc (CA):** intelligible identities, goals, emotions, relationships, and change across the complete video.

These dimensions require global evidence that independent clip averages miss. Human raters and the metric use the same anchored 0–5 rubric; the complete twenty-item inventory appears in App. B.6.

Scores combine full-video, 150-second, and 30-second evidence under a frozen criterion-by-scale contract. Before window scoring, a score-blind eligibility decision checks source-text alignment and assessability at that scope, independently of generator identity or score. Ineligible windows are excluded rather than scored as failures. Eligible-window scores are duration-weighted within each scale; temporal weights are renormalized over available scales, and within-Global weights aggregate the five criteria. Both non-overlapping window grids start at the beginning of the final video and do not restart at chapter boundaries; a final shorter window is retained and duration-weighted when eligible. Internal chapters are not separately scored or averaged. Long-range criteria such as causal closure and arc resolution rely primarily on full-video evidence unless an aligned window contains a relevant milestone; a locally absent ending is not treated as a global failure. App. B.6 specifies exact equations, evidence format, confidence and abstention policies, and retry rules. Criterion-by-scale inclusion, exclusion, and abstention rates accompany the scores to reveal selective coverage.

Evaluator-optimizer separation. Eligibility rules, temporal and within-Global weights, and prompts are calibrated only on development data and frozen before held-out scores are inspected. The common final-output interface receives neither production trajectories nor CPPE state. CineScope-Metric never enters case formation, pattern induction, patch compilation, admission, checkpoint selection, or stopping. CPPE uses a separate trajectory reviewer with independent prompts, model configuration, schema, and gate.

5 Experiments

5.1 Experimental Setup and Evaluation Protocol

Experiments use disjoint splits of 60 collection, 30 human-alignment, 45 replay-admission, and 45 system-test scripts. Replay comprises three disjoint 15-story sets (five per source), revealed after candidate freezing and excluded from collection and compilation; system-test never enters evolution. Systems share text and video backbones, fixed comparison settings and generation budgets, and one frozen CineScope VLM (App. B.1). Each story-system pair has two independent generations and two independent scores per video/evaluator, averaged equally. Checkpoints share one globally frozen reviewer and write nothing back; later evolution may change operational thresholds and routes. Source-stratified story-bootstrap 95% CIs are conditional on policies, treating stories as independent units and excluding evolution-seed uncertainty (App. B.2). Admission records and margins are specified in App. B.9.

5.2 CineScope-Metric Human Alignment

Each source supplies five calibration and five held-out alignment stories, used exclusively for these roles. Produce, MovieAgent, and AniMaker each generate every story twice (180 videos total). Ten trained university raters provide at least five valid ratings per video using the common anchored 0–5, twenty-criterion rubric (App. B.1). Inter-rater correlation is 0.789 (mean pairwise Spearman; 95% CI [0.640,0.860]). After replicate aggregation, Table 1 reports held-out alignment: overall Spearman 0.710, BT agreement 0.810, and W_1/JS similarities 0.949/0.932. Global correlations range from 0.522 to 0.680, with the highest for Character Arc. W_1/JS measure marginal distribution similarity,

not item-wise agreement. The results support a human-aligned outcome measure that remains frozen and separate from CPPE optimization, preventing evaluator feedback.

5.3 Comparison of Video Generation Agents

We compare Produce with ViMax (Huang et al., 2026), AniMaker (Shi et al., 2025), and MovieAgent (Wu et al., 2025b) under the common system-test protocol. Table 2 shows that Produce achieves the highest overall CineScope score (4.02) and ScriptAgent score (3.91). It also leads all four CineScope Global dimensions, supporting its advantage in long-form story realization.

Table 2: Generator comparison on the system-test split: (a) CineScope-Metric, (b) ScriptAgent CriticAgent, and (c) VBench-Long. Scores are story-level means, with 95% story-bootstrap CIs in brackets. Bold marks the highest mean in each column.

(a) CineScope-Metric

Generator	Weighted avg. ↑ [95% CI]	CS ↑ [95% CI]	DO ↑ [95% CI]	PR ↑ [95% CI]	CA ↑ [95% CI]
ViMax (Huang et al., 2026)	3.74 [3.66,3.82]	3.78 [3.68,3.87]	4.00 [3.92,4.07]	3.39 [3.30,3.49]	3.81 [3.69,3.91]
AniMaker (Shi et al., 2025)	3.84 [3.76,3.92]	3.88 [3.79,3.97]	3.92 [3.84,4.00]	3.53 [3.44,3.62]	4.01 [3.91,4.11]
MovieAgent (Wu et al., 2025b)	3.60 [3.52,3.69]	3.53 [3.43,3.64]	3.91 [3.83,3.98]	3.53 [3.44,3.61]	3.45 [3.32,3.58]
CineForge-Produce	**4.02 [3.95,4.10]**	**4.05 [3.96,4.13]**	**4.01 [3.94,4.08]**	**3.70 [3.61,3.78]**	**4.34 [4.25,4.43]**

(b) ScriptAgent CriticAgent

Generator	Critic avg. ↑ [95% CI]	CCA ↑ [95% CI]	KBL ↑ [95% CI]	VDF ↑ [95% CI]	EAM ↑ [95% CI]	NPT ↑ [95% CI]
ViMax (Huang et al., 2026)	3.34 [3.27,3.42]	3.12 [3.03,3.21]	3.03 [2.92,3.14]	3.79 [3.70,3.87]	3.32 [3.22,3.42]	3.47 [3.38,3.56]
AniMaker (Shi et al., 2025)	3.38 [3.31,3.45]	3.08 [2.98,3.17]	2.89 [2.78,3.00]	3.90 [3.81,3.98]	3.43 [3.33,3.53]	3.60 [3.51,3.69]
MovieAgent (Wu et al., 2025b)	3.79 [3.71,3.86]	3.58 [3.48,3.66]	**3.66 [3.56,3.77]**	3.88 [3.79,3.96]	**4.09 [4.00,4.18]**	3.72 [3.63,3.81]
CineForge-Produce	**3.91 [3.84,3.98]**	**3.82 [3.73,3.91]**	3.48 [3.37,3.59]	**4.39 [4.30,4.47]**	3.93 [3.84,4.02]	**3.94 [3.85,4.02]**

(c) VBench-Long

Generator	SC ↑ [95% CI]	BC ↑ [95% CI]	MS ↑ [95% CI]	DD ↑ [95% CI]	AQ ↑ [95% CI]	IQ ↑ [95% CI]
ViMax (Huang et al., 2026)	0.954 [0.945,0.962]	**0.964 [0.957,0.970]**	**0.989 [0.986,0.992]**	0.391 [0.361,0.422]	0.614 [0.592,0.636]	0.757 [0.739,0.776]
AniMaker (Shi et al., 2025)	**0.955 [0.946,0.963]**	0.958 [0.951,0.964]	0.982 [0.978,0.986]	**0.705 [0.677,0.733]**	0.624 [0.603,0.645]	**0.775 [0.756,0.792]**
MovieAgent (Wu et al., 2025b)	0.915 [0.904,0.925]	0.952 [0.945,0.959]	0.988 [0.984,0.991]	0.637 [0.607,0.666]	0.592 [0.570,0.614]	0.706 [0.685,0.726]
CineForge-Produce	0.924 [0.914,0.934]	0.953 [0.946,0.960]	0.989 [0.985,0.992]	0.419 [0.389,0.449]	**0.650 [0.628,0.670]**	0.747 [0.727,0.765]

5.4 Comparison of Self-Evolution Strategies

GenMAC (Huang et al., 2024a), VideoWeaver (Wei et al., 2026a), and CineForge-Evolve optimize the same Produce checkpoint on the collection split. The baseline variants use adapted optimizers, not their original generators; backbones, generation budgets, aggregation, and testing are shared. Table 3 reports end-to-end optimization-chain gains: GenMAC reaches 4.12, VideoWeaver 4.18, and Full (Round 3) 4.38, leading in all four CineScope Global dimensions. Source/checkpoint results and resource accounting are in Apps. B.2 and B.4; shared generation budgets do not imply equal total optimization expenditure.

External outcome measures. ScriptAgent and VBench-Long use matched final videos, replicates, and source inputs with frozen sampling, partitioning, aggregation, retries, and versions. Neither informs optimization or selection. VBench-Long provides complementary perceptual evidence: Full improves narrative scores but has lower BC/MS/IQ means than Round 0 (Table 3(c)). Unsupported dimensions are reported as N/A; missing supported scores are not replaced by zero or another dimension (App. B.3).

5.5 Self-Evolution Scaling

Three disjoint 20-story collection batches produce Rounds 1–3 from the unevolved Round 0 through admitted updates. Each round's *seen* measurement is a fresh rollout on its collection subset; *unseen* uses fixed system-test stories excluded from evolution. Round 0 uses the same read-only protocol

Table 3: Optimization methods applied to the common CineForge-Produce baseline on the system-test split: (a) CineScope-Metric, (b) ScriptAgent CriticAgent, and (c) VBench-Long. Scores are story-level means, with 95% story-bootstrap CIs in brackets. Round 0 is CineForge-Produce; other + rows denote adapted optimizers.

(a) CineScope-Metric

System	Weighted avg. ↑ [95% CI]	CS ↑ [95% CI]	DO ↑ [95% CI]	PR ↑ [95% CI]	CA ↑ [95% CI]
CineForge-Produce	4.02 [3.95,4.10]	4.05 [3.96,4.13]	4.01 [3.94,4.08]	3.70 [3.61,3.78]	4.34 [4.25,4.43]
CineForge-Produce + GenMAC (Huang et al., 2024a)	4.12 [4.04,4.19]	4.17 [4.08,4.26]	4.08 [4.01,4.15]	3.79 [3.71,3.88]	4.43 [4.34,4.51]
CineForge-Produce + VideoWeaver (Wei et al., 2026a)	4.18 [4.11,4.26]	4.23 [4.15,4.31]	4.18 [4.11,4.26]	3.86 [3.78,3.94]	4.46 [4.38,4.54]
CineForge (Full)	**4.38 [4.31,4.45]**	**4.54 [4.46,4.62]**	**4.37 [4.30,4.44]**	**3.95 [3.87,4.03]**	**4.65 [4.57,4.72]**

(b) ScriptAgent CriticAgent

Policy	Overall ↑ [95% CI]	CCA ↑ [95% CI]	KBL ↑ [95% CI]	VDF ↑ [95% CI]	EAM ↑ [95% CI]	NPT ↑ [95% CI]
Round 0	3.91 [3.84,3.98]	3.82 [3.73,3.91]	3.48 [3.37,3.59]	4.39 [4.30,4.47]	3.93 [3.84,4.02]	3.94 [3.85,4.02]
+ GenMAC	3.97 [3.90,4.04]	3.90 [3.82,3.99]	3.58 [3.47,3.68]	4.42 [4.34,4.50]	3.99 [3.90,4.07]	3.98 [3.89,4.06]
+ VideoWeaver	4.04 [3.97,4.10]	3.97 [3.88,4.05]	3.65 [3.55,3.76]	4.47 [4.39,4.55]	4.04 [3.96,4.13]	4.05 [3.97,4.13]
Full (Round 3)	**4.10 [4.03,4.17]**	**4.08 [3.99,4.15]**	**3.76 [3.66,3.86]**	**4.52 [4.44,4.59]**	**4.08 [4.00,4.15]**	**4.08 [3.99,4.16]**

(c) VBench-Long

Policy	SC ↑ [95% CI]	BC ↑ [95% CI]	MS ↑ [95% CI]	DD ↑ [95% CI]	AQ ↑ [95% CI]	IQ ↑ [95% CI]
Round 0	0.924 [0.914,0.934]	0.953 [0.946,0.960]	0.989 [0.985,0.992]	0.419 [0.389,0.449]	0.650 [0.628,0.670]	0.747 [0.727,0.765]
+ GenMAC	**0.948 [0.938,0.957]**	0.936 [0.929,0.944]	0.974 [0.970,0.978]	**0.449 [0.419,0.478]**	**0.677 [0.656,0.697]**	0.722 [0.703,0.741]
+ VideoWeaver	0.914 [0.903,0.924]	**0.969 [0.962,0.975]**	**0.996 [0.993,0.998]**	0.411 [0.382,0.441]	0.643 [0.621,0.664]	**0.760 [0.741,0.778]**
Full (Round 3)	0.940 [0.930,0.949]	0.946 [0.938,0.953]	0.981 [0.977,0.985]	0.441 [0.411,0.470]	0.669 [0.648,0.689]	0.735 [0.716,0.754]

before updating. All evaluation rollouts are discarded and cannot affect CaseMemory, CPPE, policy, checkpoint selection, or stopping; they share the frozen reviewer, outcome evaluators, budgets, and replication protocol (Apps. B.1.1 and B.1.5).

On unseen stories, Round 3 gains 0.355 in CineScope and 0.188 in ScriptAgent over Round 0 after rounding (Figure 3; exact checkpoints in Table 5). By Round 1, review calls fall 46.7→35.6 (23.7%) and issues 104→71.9 (30.8%), both per internal chapter. These are fixed-set point estimates; changing seen subsets permit descriptive comparisons only. Counts measure frozen-audit workload, not total production and evolution cost (Apps. B.4 and B.5).

On 45 matched system-test stories, human overall scores are 3.86 [3.80,3.92], 4.059, 4.10 [4.04,4.16], and 4.181 for Rounds 0–3, respectively (available 95% CIs in brackets). The Round 3 minus Round 0 mean gain is 0.321. The human-evaluation protocol is given in App. B.1.3. **App. A.2 illustrates how an evolved rule is applied in video production, with a before-and-after comparison (Figure 5).**

Table 4: Core CineForge-Evolve ablation. All rows use the system-test split. Cells report means (95% story-bootstrap CIs). Quality scores evaluate complete videos; review calls and issues are per internal episode.

Variant	CineScope ↑	ScriptAgent ↑	Review calls ↓	Review issues ↓
CineForge-Produce (Round 0)	4.02 [3.95,4.10]	3.91 [3.84,3.98]	46.7 [42.5,50.9]	104 [94.6,113]
CineForge (Full, Round 3)	**4.38 [4.31,4.45]**	**4.10 [4.03,4.17]**	**29.4 [26.2,32.7]**	**54.6 [48.3,61.0]**
Evolve: w/o trajectory-grounded localization	3.98 [3.90,4.06]	3.88 [3.81,3.95]	49.9 [45.4,54.4]	111 [101,122]
Evolve: w/o cross-story recurrence	4.04 [3.96,4.11]	3.91 [3.85,3.99]	45.9 [41.9,50.1]	104 [92.7,110]
Evolve: free-form policy lessons	4.21 [4.13,4.28]	3.97 [3.90,4.03]	36.8 [33.3,40.5]	70.9 [63.5,78.6]
Evolve: w/o fresh structural replay	4.17 [4.10,4.25]	3.94 [3.87,4.01]	39.3 [35.5,43.2]	76.9 [69.0,85.0]

5.6 Ablation Study

Each evolution ablation starts at Round 0, uses the same three collection batches, separate replay admission where applicable, and the fixed system-test split without selection feedback. Shared settings follow Sec. 5.5. Variants remove trajectory-grounded localization, allow single-case triggers without cross-story recurrence, replace typed bounded patches with free-form lessons, or remove

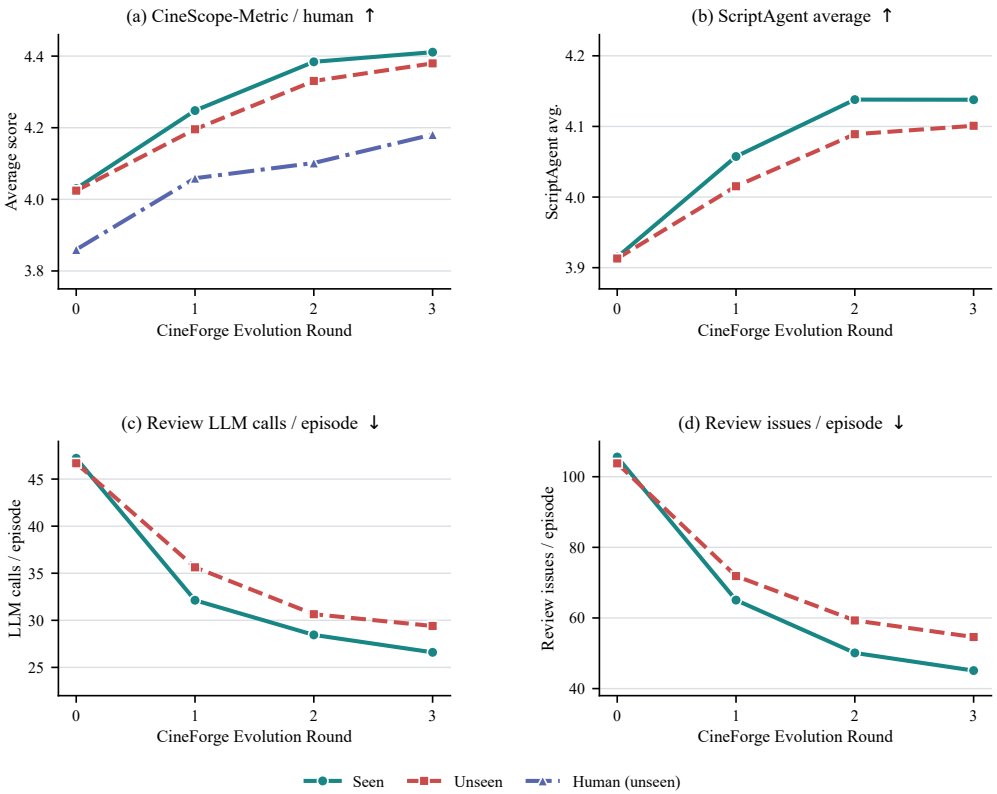

Figure 3: Round-wise self-evolution from Round 0 through Round 3 for (a) CineScope-Metric, (b) ScriptAgent, (c) Review LLM calls per episode, and (d) Review issues per episode. Here, episode denotes an internal production chapter; quality scores evaluate the complete concatenated video. Seen results use the corresponding round's collection subset; unseen results use the fixed system-test split, which is excluded from evolution. Round 0 is the pre-evolution reference. All checkpoint tests use the same globally frozen reviewer configuration and write nothing back. Panel (a) additionally shows human overall scores on the same 45 system-test stories.

fresh structural replay. The last variant runs only in an isolated branch; stochastic-impact patches still require paired admission (App. B.1.6).

Full has the highest means on both outcome metrics and the lowest review workload (Table 4). Without localization, mean scores fall below Round 0; without recurrence, they remain near it. The variants using free-form lessons and omitting fresh structural replay retain gains but underperform Full, consistent with benefits from bounded updates and replay screening. Comparisons are descriptive; statistics and workload definitions follow Apps. B.2 and B.4.

6 Conclusion

We presented CineForge, a self-evolving framework coupling CineForge-Produce and CineForge-Evolve for long-horizon video generation and persistent cross-story policy improvement. Produce organizes narrative and cinematic states into canonical production trajectories, while Case-to-Pattern-to-Policy Evolution (CPPE) transforms recurring trajectory evidence into bounded, stage-local policy updates. Fresh structural replay and confidence-controlled paired evaluation govern deployment, with conditional guarantees for structural improvement and internal admission scores. We also introduced CineScope, combining a 100-script benchmark with a human-aligned, multiscale metric for complete-video story realization, kept independent of policy optimization. Experiments across three data sources show improved story-level quality, cumulative gains on unseen stories, and reduced review workload, supported by human evaluation and component ablations. These findings establish production trajectories as reusable evidence for cumulative agent improvement, while distinguishing current-output repair, persistent policy evolution, and independent outcome evaluation.

AI USE STATEMENT

Generative AI tools assisted with modifying research code and drafting and revising the manuscript, including its organization, wording, and LaTeX formatting. AI models are also components of the evaluated system: large language models support agent planning, production decisions, review, and policy-update proposals, while video-generation models render the videos. Model-based evaluators additionally provide the automatic assessments described in the paper. These methodological uses, including model configurations and the separation between optimization and outcome evaluation, are documented in Secs. 3 and 5.1 and App. B.1. The authors retain responsibility for the final code, manuscript, experimental results, and scientific claims, including all AI-assisted content.

ETHICS STATEMENT

Long-form video generation can support creative production but also enable misleading synthetic media, unauthorized depictions, and the reproduction of biases or copyrighted material. The structural replay and admission checks in CineForge address production-policy updates and do not constitute safeguards against all such misuse; deployment requires appropriate content review, disclosure of synthetic provenance, and respect for privacy and intellectual-property rights. The human evaluations use informed consent, voluntary participation, anonymized rater identifiers, and controlled access to annotation records, as described in App. B.1.3. Corpus provenance and release considerations are discussed in App. B.1.1; any redistribution should respect item-level licenses and permissions.

REPRODUCIBILITY STATEMENT

The paper specifies the production and evolution interfaces, typed records, editable policy components, and candidate-admission procedures in Secs. 2–3.4 and Apps. B.11 and B.12. Data construction and split roles, model configurations, repeated generation and evaluation, and human-study protocols are described in App. B.1. Apps. B.6, B.2, and B.1.6 detail metric aggregation, story-level statistical analysis, and ablation settings. The assumptions, replay conditions, and full proofs are provided in Apps. B.9, B.10, and B.13.

REFERENCES

Lakshya A. Agrawal, Shangyin Tan, Dilara Soylu, Noah Ziems, Rishi Khare, Krista Opsahl-Ong, Arnav Singhvi, Herumb Shandilya, Michael J. Ryan, Meng Jiang, Christopher Potts, Koushik Sen, Alexandros G. Dimakis, Ion Stoica, Dan Klein, Matei Zaharia, and Omar Khattab. GEPA: Reflective prompt evolution can outperform reinforcement learning. In *International Conference on Learning Representations*, 2026. URL https://iclr.cc/virtual/2026/poster/10009493.

Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. *Concentration Inequalities: A Nonasymptotic Theory of Independence*. Oxford University Press, 2013.

Emanuele Bugliarello, Hernan Moraldo, Ruben Villegas, Mohammad Babaeizadeh, Mohammad Taghi Saffar, Han Zhang, Dumitru Erhan, Vittorio Ferrari, Pieter-Jan Kindermans, and Paul Voigtlaender. Storybench: A multifaceted benchmark for continuous story visualization. In *Advances in Neural Information Processing Systems*, volume 36, pp. 78095–78125, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/f63f5fbed1a4ef08c857c5f377b5d33a-Abstract-Datasets_and_Benchmarks.html.

Lirong Che, Yuzhe Yang, Peiwen Lin, Chuang Wang, Xueqian Wang, and Jian Su. Demoevolve: Overcoming sparse feedback in agentic harness evolution with demonstrations. *arXiv preprint arXiv:2605.24539*, 2026. URL https://arxiv.org/abs/2605.24539.

Jiamin Chen, Qianben Chen, Jiawen Zhang, Yidi Wu, Yuchen Li, Xiaokun Zhang, Wangchunshu Zhou, and Chen Ma. Directorbench: Diagnosing long-form video generation with personalized multi-agent evaluation. *arXiv preprint arXiv:2605.30090*, 2026a. URL https://arxiv.org/abs/2605.30090.

Mengzhuo Chen, Junjie Wang, Zhe Liu, Yawen Wang, Haiming Zheng, and Qing Wang. From failed trajectories to reliable llm agents: Diagnosing and repairing harness flaws. *arXiv preprint arXiv:2606.06324*, 2026b. URL https://arxiv.org/abs/2606.06324.

Xiaokun Feng, Haiming Yu, Meiqi Wu, Shiyu Hu, Jintao Chen, Chen Zhu, Jiahong Wu, Xiangxiang Chu, and Kaiqi Huang. NarrLV: Towards a comprehensive narrative-centric evaluation for long video generation. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=Qh3CQBTB1g.

Yuan Gong, Youxin Pang, Xiaodong Cun, Menghan Xia, Yingqing He, Haoxin Chen, Longyue Wang, Yong Zhang, Xintao Wang, Ying Shan, and Yujiu Yang. Talecrafter: Interactive story visualization with multiple characters. *ACM Transactions on Graphics*, 42(6), 2023. doi: 10.1145/3610548.3618184. URL https://arxiv.org/abs/2305.18247.

Yuwei Guo, Ceyuan Yang, Ziyan Yang, Zhibei Ma, Zhijie Lin, Zhenheng Yang, Dahua Lin, and Lu Jiang. Long context tuning for video generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 17281–17291, 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Guo_Long_Context_Tuning_for_Video_Generation_ICCV_2025_paper.html.

Yichen Han, Bojun Liu, Zhengpeng Zhou, Guanyu Zhou, Zeng Zhang, Yang Yang, Wenli Wang, Isaac N. Shi, Yunyan, Lewei He, and Tianyu Shi. Mapgd: Multi-agent prompt gradient descent for collaborative prompt optimization. *arXiv preprint arXiv:2509.11361*, 2025. URL https://arxiv.org/abs/2509.11361.

Ruozhen He, Meng Wei, Ziyan Yang, and Vicente Ordonez. EntityBench: Towards entity-consistent long-range multi-shot video generation. *arXiv preprint arXiv:2605.15199*, 2026. URL https://arxiv.org/abs/2605.15199.

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. Metagpt: Meta programming for a multi-agent collaborative framework. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=VtmBAGCN7o.

Shengran Hu, Cong Lu, and Jeff Clune. Automated design of agentic systems. *arXiv preprint arXiv:2408.08435*, 2024. URL https://arxiv.org/abs/2408.08435.

Kaiyi Huang, Yukun Huang, Xuefei Ning, Zinan Lin, Yu Wang, and Xihui Liu. Genmac: Compositional text-to-video generation with multi-agent collaboration. *arXiv preprint arXiv:2412.04440*, 2024a. URL https://arxiv.org/abs/2412.04440.

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

Qingqiu Huang, Yu Xiong, Anyi Rao, Jiaze Wang, and Dahua Lin. Movienet: A holistic dataset for movie understanding. In *European Conference on Computer Vision*, pp. 709–727, 2020.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. Vbench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024b. URL https://arxiv.org/abs/2311.17982.

Ziqi Huang, Fan Zhang, Xiaojie Xu, Yinan He, Jiashuo Yu, Ziyue Dong, Qianli Ma, Nattapol Chanpaisit, Chenyang Si, Yuming Jiang, et al. VBench++: Comprehensive and versatile benchmark suite for video generative models. *arXiv preprint arXiv:2411.13503*, 2024c. URL https://arxiv.org/abs/2411.13503.

Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *International Conference on Machine Learning*, pp. 267–274, 2002.

Ozgur Kara, Krishna Kumar Singh, Feng Liu, Duygu Ceylan, James M. Rehg, and Tobias Hinz. Shotadapter: Text-to-multi-shot video generation with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 28405–28415, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Kara_ShotAdapter_Text-to-Multi-Shot_Video_Generation_with_Diffusion_Models_CVPR_2025_paper.html.

Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, Hanna Moazam, et al. Dspy: Compiling declarative language model calls into self-improving pipelines. *arXiv preprint arXiv:2310.03714*, 2023. URL https://arxiv.org/abs/2310.03714.

Jihwan Kim, Junoh Kang, Jinyoung Choi, and Bohyung Han. FIFO-Diffusion: Generating infinite videos from text without training. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-2853. URL https://arxiv.org/abs/2405.11473.

Changxin Lao, Fei Pan, Guozhuang Ma, Han Li, Huihuang Lin, Jijun Shi, Kangzhi Zhao, Kun Gai, Mo Zhou, Qinqin Zhou, Quan Chen, Ruochen Yang, Shifu Bie, Shijie Yi, Shuang Yang, Shuo Yang, Wenhao Li, Wentao Xie, Xiao Lv, Xuming Wang, Yijun Wang, Yiming Chen, Yusheng Huang, Zhongyuan Wang, Zibo Zhao, Zijie Zhuang, et al. Agentx: Towards agent-driven self-iteration of industrial recommender systems. *arXiv preprint arXiv:2606.26859*, 2026. URL https://arxiv.org/abs/2606.26859.

Romain Laroche, Paul Trichelair, and Remi Tachet Des Combes. Safe policy improvement with baseline bootstrapping. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3652–3661, 2019. URL https://proceedings.mlr.press/v97/laroche19a.html.

Claire Le Goues, Michael Pradel, and Abhik Roychoudhury. Automated program repair. *Communications of the ACM*, 62(12):56–65, 2019.

Yitong Li, Zhe Gan, Yelong Shen, Jingjing Liu, Yu Cheng, Yuexin Wu, Lawrence Carin, David Carlson, and Jianfeng Gao. Storygan: A sequential conditional gan for story visualization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6329–6338, 2019. URL https://arxiv.org/abs/1812.02784.

Han Lin, Jaemin Cho, Abhay Zala, and Mohit Bansal. Videodirectorgpt: Consistent multi-scene video generation via llm-guided planning. *arXiv preprint arXiv:2309.15091*, 2023. URL https://arxiv.org/abs/2309.15091.

Jiahang Lin, Shichun Liu, Chengjun Pan, Lizhi Lin, Shihan Dou, Zhiheng Xi, Xuanjing Huang, Hang Yan, Zhenhua Han, Tao Gui, and Yu-Gang Jiang. Agentic harness engineering: Observability-driven automatic evolution of coding-agent harnesses. *arXiv preprint arXiv:2604.25850*, 2026a. URL https://arxiv.org/abs/2604.25850.

Minhua Lin, Juncheng Wu, Zijun Wang, Zhan Shi, Yisi Sang, Bing He, Zewen Liu, Tianxin Wei, Zongyu Wu, Zhiwei Zhang, Dakuo Wang, Xiang Zhang, Benoit Dumoulin, Cihang Xie, Yuyin Zhou, Suhang Wang, and Hanqing Lu. Harness updating is not harness benefit: Disentangling evolution capabilities in self-evolving llm agents. *arXiv preprint arXiv:2605.30621*, 2026b. URL https://arxiv.org/abs/2605.30621.

Jinzhuo Liu, Jiangning Zhang, Wencan Jiang, Yabiao Wang, Dingkang Liang, Zhucun Xue, Ran Yi, and Yong Liu. Advancing narrative long video generation via training-free identity-aware memory. *arXiv preprint arXiv:2605.18733*, 2026a. URL https://arxiv.org/abs/2605.18733.

Tengfei Liu, Yang Shi, Xuanyu Zhu, Jiafu Tang, Liu Yang, Qixun Wang, Zhuoran Zhang, Yuqi Tang, Fengxiang Wang, Yuhao Dong, Xinlong Chen, Bozhou Li, Bohan Zeng, Yue Ding, Xiaohan Zhang, Jialu Chen, Haotian Wang, Yuanxing Zhang, Pengfei Wan, and Leye Wang. LongAV-Compass: Towards unified evaluation of minute-scale audio-visual generation across T2AV, I2AV, and V2AV. *arXiv preprint arXiv:2605.26244*, 2026b. URL https://arxiv.org/abs/2605.26244.

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. 22139–22149, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Liu_EvalCrafter_Benchmarking_and_Evaluating_Large_Video_Generation_Models_CVPR_2024_paper.html.

Do Xuan Long, Xingchen Wan, Hootan Nakhost, Chen-Yu Lee, Tomas Pfister, and Sercan Ö. Arik. Vista: A test-time self-improving video generation agent. *arXiv preprint arXiv:2510.15831*, 2025. URL https://arxiv.org/abs/2510.15831.

Do Xuan Long, Yale Song, Min-Yen Kan, Tomas Pfister, and Long T. Le. A^2rd: Agentic autoregressive diffusion for long video consistency. *arXiv preprint arXiv:2605.06924*, 2026. URL https://arxiv.org/abs/2605.06924.

Xinghua Lou, Miguel Lázaro-Gredilla, Antoine Dedieu, Carter Wendelken, Wolfgang Lehrach, and Kevin P. Murphy. Autoharness: Improving llm agents by automatically synthesizing a code harness. *arXiv preprint arXiv:2603.03329*, 2026. URL https://arxiv.org/abs/2603.03329.

Yu Lu, Yuanzhi Liang, Linchao Zhu, and Yi Yang. Freelong: Training-free long video generation with spectralblend temporal attention. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-4177. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/ed67dff7cb96e7e86c4d91c0d5db49bb-Abstract.html.

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. *arXiv preprint arXiv:2303.17651*, 2023. URL https://arxiv.org/abs/2303.17651.

Ryosuke Matsuda, Keito Kudo, Haruto Yoshida, Nobuyuki Shimizu, and Jun Suzuki. SLVMEval: Synthetic meta evaluation benchmark for text-to-long video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7784–7794, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/papers/Matsuda_SLVMEval_Synthetic_Meta_Evaluation_Benchmark_for_Text-to-Long_Video_Generation_CVPR_2026_paper.pdf.

Chenyu Mu, Xin He, Qu Yang, Wanshun Chen, Jiadi Yao, Huang Liu, Zihao Yi, Bo Zhao, Xingyu Chen, Ruotian Ma, Fanghua Ye, Erkun Yang, Cheng Deng, Zhaopeng Tu, Xiaolong Li, and Linus. The script is all you need: An agentic framework for long-horizon dialogue-to-cinematic video generation. *arXiv preprint arXiv:2601.17737*, 2026. URL https://arxiv.org/abs/2601.17737.

Jun Nie, Yonggang Zhang, Jun Song, Qianshu Cai, Dahai Yu, Yike Guo, Xinmei Tian, and Bo Han. TTHE: Test-time harness evolution. *arXiv preprint arXiv:2607.08124*, 2026. URL https://arxiv.org/abs/2607.08124.

Krista Opsahl-Ong, Michael J. Ryan, Josh Purtell, David Broman, Christopher Potts, Matei Zaharia, and Omar Khattab. Optimizing instructions and demonstrations for multi-stage language model programs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 9340–9366, 2024. doi: 10.18653/v1/2024.emnlp-main.525. URL https://aclanthology.org/2024.emnlp-main.525/.

Linyue Pan, Lexiao Zou, Shuo Guo, Jingchen Ni, and Hai-Tao Zheng. Natural-language agent harnesses. *arXiv preprint arXiv:2603.25723*, 2026a. URL https://arxiv.org/abs/2603.25723.

Wenbo Pan, Shujie Liu, Chin-Yew Lin, Jingying Zeng, Xianfeng Tang, Xiangyang Zhou, Yan Lu, and Xiaohua Jia. Evolving agents in the dark: Retrospective harness optimization via self-preference. *arXiv preprint arXiv:2606.05922*, 2026b. URL https://arxiv.org/abs/2606.05922.

Matteo Pirotta, Marcello Restelli, Alessio Pecorino, and Daniele Calandriello. Safe policy iteration. In *International Conference on Machine Learning*, pp. 307–315, 2013.

Reid Pryzant, Dan Iter, Jerry Li, Yin Tat Lee, Chenguang Zhu, and Michael Zeng. Automatic prompt optimization with "gradient descent" and beam search. *arXiv preprint arXiv:2305.03495*, 2023. URL https://arxiv.org/abs/2305.03495.

Lu Qiu, Yizhuo Li, Yuying Ge, Yixiao Ge, Ying Shan, and Xihui Liu. Animeshooter: A multi-shot animation dataset for reference-guided video generation. *arXiv preprint arXiv:2506.03126*, 2025. URL https://arxiv.org/abs/2506.03126.

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*, 2023. URL https://arxiv.org/abs/2211.13319.

Haoyuan Shi, Yunxin Li, Xinyu Chen, Longyue Wang, Baotian Hu, and Min Zhang. AniMaker: Multi-agent animated storytelling with MCTS-driven clip generation. *arXiv preprint arXiv:2506.10540*, 2025. URL https://arxiv.org/abs/2506.10540.

Haoyuan Shi, Yunxin Li, Nanhao Deng, Zhenran Xu, Xinyu Chen, Longyue Wang, Baotian Hu, and Min Zhang. Msvbench: Towards human-level evaluation of multi-shot video generation. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 24034–24058. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.findings-acl.1203. URL https://aclanthology.org/2026.findings-acl.1203/.

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*, 2024.

Anubhav Shrimal, Stanley Kanagaraj, Kriti Biswas, Swarnalatha Raghuraman, Anish Nediyanchath, Yi Zhang, and Promod Yenigalla. MARCO: Multi-agent real-time chat orchestration. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pp. 1381–1392, 2024. doi: 10.18653/v1/2024.emnlp-industry.102. URL https://aclanthology.org/2024.emnlp-industry.102/.

Achint Soni, Sreyas Venkataraman, Abhranil Chandra, Sebastian Fischmeister, Percy Liang, Bo Dai, and Sherry Yang. Videoagent: Self-improving video generation. *arXiv preprint arXiv:2410.10076*, 2024. URL https://arxiv.org/abs/2410.10076.

Makarand Tapaswi, Martin Bäuml, and Rainer Stiefelhagen. Book2movie: Aligning video scenes with book chapters. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1827–1835, 2015. doi: 10.1109/CVPR.2015.7298792. URL https://openaccess.thecvf.com/content_cvpr_2015/html/Tapaswi_Book2Movie_Aligning_Video_2015_CVPR_paper.html.

Philip Thomas, Georgios Theocharous, and Mohammad Ghavamzadeh. High confidence policy improvement. In *International Conference on Machine Learning*, pp. 2380–2388, 2015.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023. URL https://arxiv.org/abs/2305.16291.

Jiuniu Wang, Zehua Du, Yuyuan Zhao, Bo Yuan, Kexiang Wang, Jian Liang, Yaxi Zhao, Yihen Lu, Gengliang Li, Junlong Gao, Xin Tu, and Zhenyu Guo. Aesopagent: Agent-driven evolutionary system on story-to-video production. *arXiv preprint arXiv:2403.07952*, 2024. URL https://arxiv.org/abs/2403.07952.

Shengyi Wang, Niantong Li, Guangzheng Hu, Hong Qi, Fei Ding, Weixu Qiao, Jinlin Wang, Xiaotong Lv, Peng Han, Zimeng Li, Fanshu Ding, Yushu Wang, Han Wu, Jingjing Chen, Chongxiao Wang, Yanhao Wu, Chenglong Huang, Xiaoqian Zhu, Jie Tian, Hua Li, Jingjing Fan, Mingshuang Tang, Zhong Li, Hengxia Qiang, Weibin Chen, Jinyang Zhen, Bing Zhao, Lin Qu, Jing Li, and Hu Wei. FilmBench: A film-grade benchmark for cinematic video generation. *arXiv preprint arXiv:2607.24241*, 2026a. URL https://arxiv.org/abs/2607.24241.

Yike Wang, Huaisheng Zhu, Zhengyu Hu, Yige Yuan, Zhengyu Chen, Shakti Senthil, Hannaneh Hajishirzi, Yulia Tsvetkov, Pradeep Dasigi, and Teng Xiao. Rethinking the evaluation of harness evolution for agents. *arXiv preprint arXiv:2607.12227*, 2026b. URL https://arxiv.org/abs/2607.12227.

Yiping Wang, Xuehai He, Kuan Wang, Luyao Ma, Jianwei Yang, Shuohang Wang, Simon Shaolei Du, and Yelong Shen. Is your world simulator a good story presenter? a consecutive events-based benchmark for future long video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13629–13638, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Wang_Is_Your_World_Simulator_a_Good_Story_Presenter_A_Consecutive_CVPR_2025_paper.html.

Jianhui Wei, Jie Tan, Hengchuan Zhu, Xiaotian Zhang, Yan Zhang, Ziyi Chen, Daoan Zhang, Wei Xu, and Zuozhu Liu. Videoweaver: Evaluating and evolving skills for agentic long video generation. *arXiv preprint arXiv:2606.08091*, 2026a. URL https://arxiv.org/abs/2606.08091.

Yujie Wei, Yujin Han, Zhekai Chen, Yongming Li, Kaixun Jiang, et al. Msavbench: Towards comprehensive and reliable evaluation of multi-shot audio-video generation. *arXiv preprint arXiv:2605.20183*, 2026b. URL https://arxiv.org/abs/2605.20183.

Weijia Wu, Mingyu Liu, Zeyu Zhu, Xi Xia, Haoen Feng, Wen Wang, Kevin Qinghong Lin, Chunhua Shen, and Mike Zheng Shou. Moviebench: A hierarchical movie level dataset for long video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 28984–28994, 2025a. URL https://openaccess.thecvf.com/content/CVPR2025/html/Wu_MovieBench_A_Hierarchical_Movie_Level_Dataset_for_Long_Video_Generation_CVPR_2025_paper.html.

Weijia Wu, Zeyu Zhu, and Mike Zheng Shou. Automated movie generation via multi-agent cot planning. *arXiv preprint arXiv:2503.07314*, 2025b. URL https://arxiv.org/abs/2503.07314.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V. Le, Denny Zhou, and Xinyun Chen. Large language models as optimizers. In *International Conference on Learning Representations*, 2024a. URL https://arxiv.org/abs/2309.03409.

Shuai Yang, Yuying Ge, Yang Li, Yukang Chen, Yixiao Ge, Ying Shan, and Yingcong Chen. Seed-story: Multimodal long story generation with large language model. *arXiv preprint arXiv:2407.08683*, 2024b. URL https://arxiv.org/abs/2407.08683.

Yifan Yang, Ziyang Gong, Weiquan Huang, Qihao Yang, Ziwei Zhou, Zisu Huang, Yan Li, Xuemei Gao, Qi Dai, Bei Liu, Kai Qiu, Yuqing Yang, Dongdong Chen, Xue Yang, and Chong Luo. Skillopt: Executive strategy for self-evolving agent skills. *arXiv preprint arXiv:2605.23904*, 2026. URL https://arxiv.org/abs/2605.23904.

Mert Yuksekgonul, Federico Bianchi, Joseph Boen, Sheng Liu, Zhi Huang, Carlos Guestrin, and James Zou. Textgrad: Automatic "differentiation" via text. *arXiv preprint arXiv:2406.07496*, 2024. URL https://arxiv.org/abs/2406.07496.

Jiayi Zhang, Jinyu Xiang, Zhaoyang Yu, Fengwei Teng, Xionghui Chen, Jiaqi Chen, Mingchen Zhuge, Xin Cheng, Sirui Hong, Jinlin Wang, Bingnan Zheng, Bang Liu, Yuyu Luo, and Chenglin Wu. Aflow: Automating agentic workflow generation. *arXiv preprint arXiv:2410.10762*, 2024. URL https://arxiv.org/abs/2410.10762.

Qizheng Zhang, Changran Hu, Shubhangi Upasani, Boyuan Ma, Fenglu Hong, Vamsidhar Kamanuru, Jay Rainton, Chen Wu, Mengmeng Ji, Hanchen Li, Urmish Thakker, James Zou, and Kunle Olukotun. Agentic context engineering: Evolving contexts for self-improving language models. In *International Conference on Learning Representations*, 2026. URL https://iclr.cc/virtual/2026/poster/10008343.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. Expel: Llm agents are experiential learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 19632–19642, 2024. doi: 10.1609/aaai.v38i17.29936. URL https://ojs.aaai.org/index.php/AAAI/article/view/29936.

Xiangqing Zheng, Chengyue Wu, Kehai Chen, and Min Zhang. LoCoT2V-Bench: Benchmarking long-form and complex text-to-video generation. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2510.26412.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. Large language models are human-level prompt engineers. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2211.01910.

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. URL https://arxiv.org/abs/2405.01434.

Cailin Zhuang, Ailin Huang, Yaoqi Hu, Jingwei Wu, Wei Cheng, Jiaqi Liao, Hongyuan Wang, Xinyao Liao, Weiwei Cai, Hengyuan Xu, Xuanyang Zhang, Xianfang Zeng, Zhewei Huang, Gang Yu, and Chi Zhang. Vistorybench: Comprehensive benchmark suite for story visualization. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026. URL https://arxiv.org/abs/2505.24862.

A Qualitative Examples and Related Work

A.1 Production-to-Evolution Workflow Example

Figure 4 shows one recorded video production and the earlier rule it reuses. Read the upper row left to right and the lower row right to left.

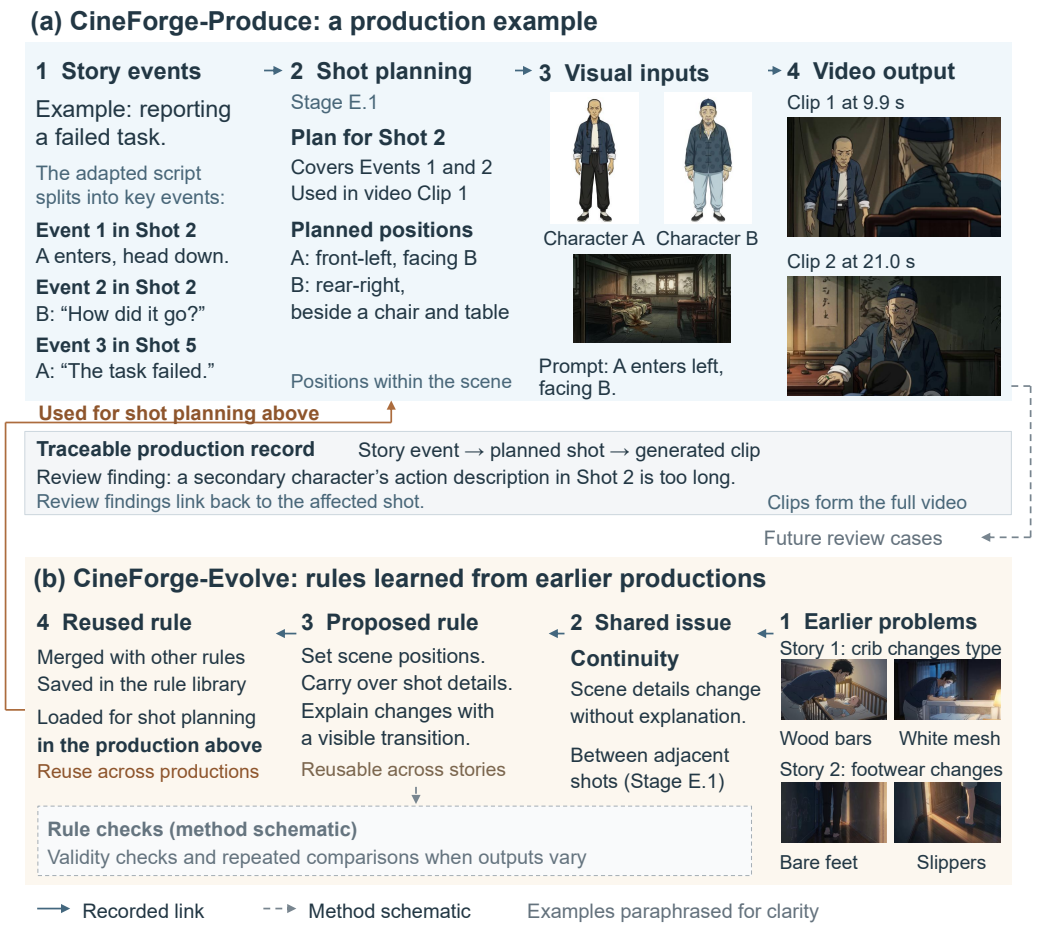

Figure 4: A real production and the earlier rule it reuses. (a) Story events guide shot plans, visual inputs, and video generation. A linked record preserves these steps and review findings. (b) Earlier continuity problems motivate a planning rule, which enters later production through a recorded merge with other rules. Solid arrows show recorded links; the gray box summarizes rule-validation checks, and dashed arrows indicate schematic links for rule checking and future case collection.

In production, CineForge splits the adapted script into key events (narrative atoms), assigns them to shots, and specifies character positions. Reference images and prompts guide clip generation. A linked record connects the clips to their inputs and review findings. Review-Rewrite revises current artifacts, while persistent rules guide later productions.

For evolution, earlier stories show unexplained changes between shots. These cases motivate a Stage E.1 rule that specifies scene positions, carries over shot details, and requires transitions for changes. The archive records a merge with other rules and subsequent loading in the displayed production. The dashed gray box summarizes the method's rule-validation checks; the dashed feedback arrow indicates future case collection. Together, the panels illustrate how production artifacts support rule formation across stories and reuse in later productions.

A.2 Cross-Story Policy Evolution Example

Figure 5 compares the same narrative segment before and after policy evolution. The Stage B rule below specifies how to distinguish characters visually and maintain their identities across shots.

> *At Stage B, establish a distinct visual identity for each principal character through facial features, hairstyle, hair color, physique, and signature clothing, and maintain these features across shots. Characters with different narrative roles must remain visually distinguishable rather than share an identical appearance. Define story-supported variations, such as costume changes or aging, in advance. Prevent unexplained changes in appearance, while allowing transformations, disguises, and hallucinations explicitly supported by the source. Preserve the character's identity, gender, species, and other core facts specified by the story, and keep all character descriptions and visual representations consistent with these constraints.*



Figure 5: Character-identity comparison for the same 60–90 s video segment. The top row (Base) shows outputs before evolution; the bottom row (Iterate) shows outputs after evolution. Distinct white and red costumes make the two characters easier to identify across shots. The case's CA1 character-identity recognizability score increases from 2/5 to 5/5.

In the baseline, the two female characters wear similar red costumes, making their identities difficult to distinguish; an unexplained clothing change adds further ambiguity. With the updated policy, the restrained character wears white while the other character retains a distinctive red costume. This stable visual contrast makes their identities easier to follow across shots. The case's CA1 character-identity recognizability score increases from 2/5 to 5/5, illustrating improved character recognizability after policy evolution.

A.3 Detailed Related Work

A.3.1 Long-Horizon Story and Video Generation

Recent video-generation systems increasingly use language models or agents to decompose a high-level description into scenes, shots, camera plans, or multi-agent production roles. Methods such as VideoDirectorGPT, MovieAgent, GenMAC, ScriptAgent, StoryDiffusion, and SEED-Story show the value of planning, memory, or consistency mechanisms for multi-scene visual generation (Lin et al., 2023; Wu et al., 2025b; Huang et al., 2024a; Mu et al., 2026; Zhou et al., 2024; Yang et al., 2024b; Hong et al., 2024; Shrimal et al., 2024). These systems motivate our setting, but they optimize a different object. Their main goal is to produce better plans or videos for the current input; our focus is on making repeated production failures measurable and using them to update the persistent rule layer of the workflow. The distinction matters for long-horizon production: a system that repairs one failed clip is useful, but a system that prevents the same structural failure from recurring across later stories changes the behavior of the production process itself.

Earlier story-visualization systems established sequential generation, visual memory, and multi-character consistency as central requirements for narrative visual synthesis (Li et al., 2019; Rahman et al., 2023; Gong et al., 2023). Long-video methods further extend temporal context or introduce shot-aware adaptation, but do not by themselves define how recurring cross-stage production failures should update a persistent workflow (Kim et al., 2024; Lu et al., 2024; Guo et al., 2025; Kara et al., 2025).

A.3.2 Long-Video and Narrative Evaluation

Rendered-output suites such as VBench, EvalCrafter, and VBench-Long measure perceptual quality, prompt alignment, motion, and long-video consistency (Huang et al., 2024b; Liu et al., 2024; Huang et al., 2024c). StoryEval, StoryBench, and ViStoryBench add event completion, sequential tasks, or story-visualization criteria (Wang et al., 2025; Bugliarello et al., 2023; Zhuang et al., 2026), while NarrLV and LoCoT2V-Bench explicitly evaluate narrative richness, event-level alignment, temporal consistency, and higher-level narrative realization (Feng et al., 2026; Zheng et al., 2026). MSVBench, EntityBench, and NarraStream-Bench specialize in multi-shot coherence, long-range entity consistency, or continuity across evolving prompts (Shi et al., 2026; He et al., 2026; Liu et al., 2026a). MSAVBench and LongAV-Compass broaden evaluation to audio-visual generation and synchronization (Wei et al., 2026b; Liu et al., 2026b); DirectorBench and FilmBench provide directorial, personalized, or professional cinematic diagnostics (Chen et al., 2026a; Wang et al., 2026a). SLVMEval instead meta-evaluates whether long-video evaluators detect controlled degradations, indicating that evaluator reliability must itself be tested (Matsuda et al., 2026).

These benchmarks are complementary rather than interchangeable. CineScope-Metric does not replace their specialized perceptual, audio-visual, entity-consistency, or cinematic taxonomies. It is distinguished by a compact final-output protocol centered on four full-video Global dimensions and twenty human-aligned criteria. Production-trajectory localization belongs to the review module within CPPE and is evaluated separately from CineScope-Metric.

A.3.3 Self-Improving and Agentic Video Generation

Several recent systems explicitly frame video generation as agentic or self-improving. AesopAgent studies an agent-driven evolutionary system for story-to-video production (Wang et al., 2024). VideoAgent emphasizes self-improving video generation through iterative agent feedback (Soni et al., 2024). VISTA performs test-time self-improvement by refining prompts with self-evaluation and multi-agent critique for a current generation request (Long et al., 2025). A^2RD formulates long-video synthesis as a closed-loop agentic autoregressive diffusion process with consistency enforcement (Long et al., 2026), while VideoWeaver evaluates and evolves reusable skills for agentic long-video generation (Wei et al., 2026a). These works are closest in motivation and justify a precise boundary for our contribution. CPPE differs in the optimization object and deployment criterion: it does not primarily optimize a single test-time prompt, autoregressive diffusion trajectory, current story output, or reusable skill alone. Instead, it converts recurring failures across productions into typed persistent policy patches, admits them through fresh paired replay and risk accounting, and evaluates transfer on unseen stories under a frozen policy.

A.3.4 Self-Refinement, Reflection, and Agent Memory

Self-refinement and reflection methods let language agents critique their own outputs, store feedback, and condition future attempts on past experience. Reflexion, Self-Refine, Voyager-style skill libraries, and related memory-based agents show that textual feedback can improve agent behavior without gradient updates (Shinn et al., 2024; Madaan et al., 2023; Wang et al., 2023; Zhao et al., 2024). Our setting differs in two ways. First, we separate *in-run repair* from *across-run rule evolution*: a fixer may repair the current artifact, but only recurring failures that pass a recurrence test become candidates for persistent rule changes. Second, the write-back object is not an unconstrained memory entry; it is a typed patch over prompt rules, thresholds, genre profiles, or review policies, with risk levels and regression gates.

At the broader agent level, reusable skill, context, workflow, and harness optimization treats persistent external state as an editable object. SkillOpt evolves bounded skill documents, Agentic Context Engineering retains and revises structured context, and AgentX uses trace-derived semantic updates with replay (Yang et al., 2026; Zhang et al., 2026; Lao et al., 2026). ADAS and AFlow automate agent or workflow design (Hu et al., 2024; Zhang et al., 2024), while TTHE, AutoHarness, DemoEvolve, RHO, AHE, NLAH, and HarnessFix further study persistent test-time or harness adaptation (Nie et al., 2026; Lou et al., 2026; Che et al., 2026; Pan et al., 2026b; Lin et al., 2026a; Pan et al., 2026a; Chen et al., 2026b). CPPE applies this specialization to long-horizon video production by tying bounded text-space policy patches to typed production traces, deterministic structural guards, and unseen-story transfer.

A.3.5 PROMPT, PROGRAM, AND TEXT-SPACE OPTIMIZATION

Automatic Prompt Engineer, ProTeGi/APO, OPRO, TextGrad, DSPy, and related systems treat natural-language prompts or LM programs as optimizable objects (Zhou et al., 2023; Pryzant et al., 2023; Yang et al., 2024a; Yuksekgonul et al., 2024; Khattab et al., 2023; Opsahl-Ong et al., 2024; Agrawal et al., 2026; Han et al., 2025). We share their premise that text-space parameters can be improved without updating model weights. The difference is the deployment substrate and safety constraint. We optimize a multi-stage production policy whose failures are observed on explicit intermediate states, not a single prompt scored on a fixed training set. Candidate edits must be bounded, typed, auditable, and regression-tested because a prompt-rule change can affect story decomposition, shot planning, spatial consistency, safety rewriting, and prompt rendering across future productions.

A.3.6 SAFE IMPROVEMENT AND REGRESSION-TESTED REPAIR

Our deployment rule is closer in spirit to safe policy improvement and regression-tested program repair than to unconstrained prompt search (Kakade & Langford, 2002; Pirotta et al., 2013; Thomas et al., 2015; Laroche et al., 2019; Le Goues et al., 2019). A candidate rule patch is accepted only when replayed structural metrics do not regress on a held-out regression set; patches that can affect rendered semantics additionally require stochastic video-level comparison with a confidence margin. Recent analyses further caution that frequent harness updates do not by themselves establish downstream benefit and should be evaluated under fixed budgets and held-out tasks (Lin et al., 2026b; Wang et al., 2026b). RHO (Pan et al., 2026b) is a related but different line of agent optimization: it retrospectively optimizes general harness behavior from rollout self-preference rather than targeting long-horizon video generation. Our setting uses localized deterministic failures and paired counterfactual replay because video failures are delayed, multi-stage, and often partially deterministic before rendering. The patched object is not executable code and the environment is not a Markov decision process; it is a text-space policy controlling a compound generative workflow.

A.3.7 BOUNDARY OF NOVELTY

We do not claim a new video backbone, diffusion architecture, generic LLM optimizer, or open-ended self-modifying agent. The novelty lies in the combination required by long-horizon story-to-video production: explicit intermediate states that make structural failures measurable; cross-production case memory that abstracts local failures into recurring patterns; typed persistent patches over a multi-stage text-space policy; and fresh replay gates that turn adaptive self-evolution into a certifiable sequence of risk-adjusted structural improvements. Here, a persistent update changes behavior on future stories rather than only repairing the current output.

B ADDITIONAL EXPERIMENTAL AND METHOD DETAILS

B.1 COMPLETE EXPERIMENTAL SETUP AND PROTOCOLS

B.1.1 DATA CONSTRUCTION AND SPLIT ALLOCATION

We construct **CineScope-Data** to better match long-video production requirements and to provide evaluation-ready source material in which cross-scene dependencies, pacing variation, character development, and state continuity can be meaningfully tested. The suite contains 100 long-form benchmark scripts covering diverse genres, narrative organizations, character configurations, temporal spans, and scene-transition patterns. Each item contains a complete script or script-like narrative and a stable identifier from which task packages are derived. All 100 CineScope-Data scripts are used: 60 belong to the evolution-collection split, and the remaining 40 join 40 AnimeShooter and 40 ViStoryBench scripts to form a 120-script non-collection pool. Ten scripts from each source form the 30-script human-alignment split, 15 from each form the 45-script replay-admission split, and the remaining 15 from each form the disjoint 45-script system-test split.

Classification design. Figure 6 summarizes the corpus construction plan along four dimensions: length tier, input format, genre, and visual style. The length tiers are Short, Standard, and Stress; input formats distinguish narrative prose from script or dialogue. Genres span mystery and investigation,

family and relationships, workplace and social conflict, rebirth and revenge, historical intrigue, science fiction and fantasy, survival and action, and time and memory. Visual styles range from cinematic realism and animation to period-drama, fantasy, and painterly or retro aesthetics. Genre describes story content, whereas visual style describes its intended presentation, so a genre is not tied to a single visual style. These assignments guide story selection or writing and are distinct from the experimental splits. Each ring shows a separate marginal allocation; angular alignment across rings does not indicate shared case membership. This design varies both narrative material and presentation requirements to support evaluation of continuity, causal development, pacing, and character progression across different production settings.



Figure 6: CineScope-Data classification design across input format, length tier, visual style, and genre, shown from the innermost to the outermost ring.

Corpus provenance and release. CineScope-Data draws on domestic and international long-form novels, stories, and screenplays. Specific platforms and item-level provenance, license, and permission details are withheld during anonymous review. Upon acceptance or public release, a dataset card and corpus manifest will disclose each item's source, language, license or permission status, and permitted disclosure and redistribution scope, alongside inclusion/exclusion decisions, deduplication and normalization procedures, and split identifiers. Public accessibility does not imply redistribution rights; where source texts cannot be redistributed, release is limited to identifiers, metadata, derived annotations, or other information permitted by the applicable terms.

Data sources and split roles. We map CineScope-Data, AnimeShooter (Qiu et al., 2025), and ViStoryBench (Zhuang et al., 2026) to a common task-package and trace schema. The source allocation above is accompanied by a released manifest recording the exact story identifiers, the three disjoint 20-script round subsets that exhaust the CineScope-Data evolution-collection split, the 30 human-alignment identifiers and their development/held-out roles, the three disjoint 15-script replay-admission sets, and the fixed 45-script system-test identifiers. The evolution-collection, human-alignment, replay-admission, and system-test splits are mutually disjoint. Within each experimental track, all outputs sharing a `story_id` remain within the same declared split. Shot- or scene-level evidence is aggregated within its parent story.

Experimental tracks. CineScope-Metric alignment uses the dedicated 30-script human-alignment split (10 per source; five development and five held-out stories per source) and renders each script

twice with CineForge-Produce, MovieAgent, and AniMaker, yielding 180 human-study videos; development ratings select the metric, and held-out stories evaluate the frozen configuration. Both alignment roles are system-unseen and disjoint from CineForge policy development, CPPE collection, replay admission, stopping, and all subsequent system comparisons. Their generation and evaluation are performed at read-only checkpoints, and no resulting artifact is written to the evolving system. The generator comparison evaluates CineForge-Produce, ViMax, AniMaker, and MovieAgent on the fixed 45-script system-test split. The controlled optimization comparison uses the same 60-script evolution-collection split for each optimization method and tests the resulting systems on that 45-script system-test split; CineForge-Evolve additionally uses the separate 45-script replay-admission split solely through its declared candidate gate. The round-wise self-evolution experiment uses three disjoint 20-script subsets that exhaust the CineScope-Data evolution-collection split; at each checkpoint, seen measurements use the corresponding 20-script round subset, whereas unseen measurements use the fixed 45-script system-test split. AnimeShooter and ViStoryBench do not enter collection or seen measurements, but 15 scripts from each contribute to every unseen measurement through the system-test split. All human-alignment and system-test identifiers and standardized evaluation outputs are excluded from collection, CaseMemory, patch compilation, replay admission, checkpoint selection, and stopping. Replay-admission identifiers and artifacts are excluded from collection, CaseMemory, pattern induction, patch compilation, final system testing, and stopping; they are consumed only by the predeclared candidate-admission gate.

B.1.2 Unified Video-Production and Automatic-Evaluation Protocol

Shared video-production and video-evaluation configuration. All compared production systems, including CineForge and all baseline or adapted variants, use the same text and video backbones: GLM 5.1 for text-side production calls and Seedance 2.0 for video generation. CineScope-Metric uses Qwen3.6-Plus for all visual-language evaluation calls (API model ID: `qwen3.6-plus-2026-04-02`); this VLM configuration is frozen across all system comparisons. The production systems receive the same source material, generation budget, output specification, and concatenation/post-processing policy, so each comparison isolates the production framework or optimization mechanism rather than the underlying backbones. The shared configuration uses temperature 0.7, top-p 0.9, an 8192-token maximum output length, at most three retries, and a concurrency cap of four; exposed API seeds are recorded. Video requests use a 16:9 aspect ratio, 1080p resolution, and 24 fps. The run manifest records these settings and any interface-specific fields exposed by each provider.

Repeated-generation and repeated-evaluation protocol. For every story-system pair in the automatic-evaluation tracks, we generate two videos with independent generation randomness under identical model, budget, and input settings. Each generated video is evaluated twice independently by each automatic evaluator, producing $2 \times 2 = 4$ call-level scores per evaluator. For each evaluator separately, these four scores are averaged equally to obtain the unique story-system score used in every subsequent comparison and statistical analysis under that evaluator. Review LLM calls and Review issues from the globally frozen test reviewer are averaged over the two production runs. Operational deployment-cost measurements form a separate track using each policy's actual review configuration; they are not substituted for the frozen-review audit. Replicate identifiers, exposed seeds, evaluator-call identifiers, invalid-run handling, and the four-to-one aggregation are retained in the run manifest; statistical analyses treat the story, not an individual video or evaluator call, as the independent unit.

B.1.3 Human-Alignment Protocol

Human-study protocol. Ten trained evaluators recruited from a university score complete videos using the same 20 definitions and 0-5 anchors as CineScope-Metric; each of the two generated videos per story-generator pair receives at least five independent ratings balanced by source, generator, and generation replicate. Human labels validate CineScope-Metric and are not collected for VBench-Long or ScriptAgent. Ratings are first aggregated per video and then across the two generation replicates, while the complete story remains the independent statistical unit.

Human-evaluation configuration. We recruited ten evaluators from a university and trained them on the narrative-video rubric and qualification examples before formal scoring. Before participation, evaluators received a description of the research purpose, tasks, expected time, compensation, and

data use and provided informed consent; participation was voluntary and withdrawal was permitted at any time. Assignments were balanced and randomized, and system identities were hidden. Records use anonymous rater identifiers, omit public personally identifying information, and are stored under access control. Missing responses, failed attention checks, and qualification failures are handled by predeclared exclusion and replacement rules rather than post-hoc score inspection.

Evaluator calibration. Development-set human ratings are used to select temporal-scale weights, within-Global aggregation weights, confidence and abstention thresholds, and bounded revisions to the evaluator prompts. Selection maximizes development alignment subject to stable score validity and evidence coverage; equal weights and the previous prompt version are retained when a candidate does not improve the preregistered development objective. All weights, gates, prompts, retry behavior, and model versions are frozen before any held-out score is inspected, and held-out ratings never trigger further evaluator changes.

Human-rater audit. The released annotation audit records rater qualifications, the number of raters assigned to each video, repeated assignments, valid-rating coverage, inter-rater reliability, adjudication rate, invalid responses, and median annotation time per complete video. Qualification requires experience with narrative video or script analysis, completion of the CineScope-Metric rubric tutorial, and a held-out calibration batch. CineScope-Metric annotations concern visible final-video evidence; production-stage cause labels are collected separately for CPPE process audits and do not change the human benchmark score.

Alignment statistics. Human consensus is averaged over valid raters per video and subdimension; Global scores are first aggregated within rater and then averaged, and the two generation replicates are averaged for each story-generator pair. For each analogous machine scope or subdimension, the two generation replicates and two evaluator calls per replicate produce four call-level scores, which are averaged equally to obtain one story-generator score before any downstream analysis. Spearman correlation is computed across story-generator units for each subdimension, Global aggregate, and optional overall aggregate. Bradley-Terry agreement uses declared tie handling on within-story comparisons among the three matched generators after replicate aggregation. Scope-level Spearman uses Fisher-z macro-aggregation and BT uses valid-pair weighting. For the held-out results in Table 1, source-stratified cluster bootstrap resamples the 15 held-out `story_id`s with all generation replicates, machine calls, and raw ratings, refitting BT per bootstrap replicate. Dimension-wise tests use Benjamini-Hochberg FDR correction.

Human evaluation of evolution gains. Rounds 0–3 are evaluated on the same complete 45-story system-test split used for CineScope-Metric, with 15 stories each from CineScope-Data, AnimeShooter, and ViStoryBench. Each story has two independently generated videos per round, yielding 90 videos per round. We use the same ten trained university evaluators, qualification and training procedures, consent and data-handling practices, and predeclared exclusion and replacement rules as the human-alignment study above. Each complete concatenated video receives five valid independent ratings (450 per round), with balanced, randomized assignments and hidden system and checkpoint identities. Evaluators score videos individually using the same twenty-criterion, anchored 0–5 rubric and frozen overall aggregation weights.

Scores are aggregated over valid raters per video and then equally over the two generation replicates; the reported mean gives equal weight to each story. Following App. B.2, 95% percentile confidence intervals use 10,000 source-stratified story-bootstrap resamples. All four checkpoints, generation replicates, and associated ratings of a sampled story remain together in each resample; the Round 3 minus Round 0 gain is computed from paired story-level scores. These system-test ratings are separate from the human-alignment development and held-out splits and are used only for final evaluation, never for evaluator calibration, policy updates, checkpoint selection, or stopping.

B.1.4 External-Evaluator Protocol

Because complete videos may exceed the native input horizon of the companion evaluators, VBench-Long and ScriptAgent use frozen input-only long-video wrappers for video decoding, temporal partitioning, coverage tracking, and aggregation of supported evaluation units. These wrappers preserve each evaluator's original metric definitions, judging criteria, and score semantics; wrapper instructions only expose partitioned inputs and add no CineScope-Metric dimensions, production traces, or optimization feedback. VBench-Long otherwise retains its published implementation.

ScriptAgent additionally uses a frozen source/script-and-video field adapter and is excluded from case generation, patch admission, checkpoint selection, and stopping. Every automatic evaluator is called twice independently on each of the two generated videos, and, for each evaluator separately, the resulting four scores are averaged equally to obtain the unique story-system result described above. The run manifest records exact model and prompt versions, input modalities, reference-audio policy, temporal sampling, evaluator-call identifiers, aggregation, retries, seeds when exposed, invalid-output handling, and cost. Each baseline receives only fields supported by its documented interface, and retained and omitted fields are recorded by the frozen adapter. The adapters add neither manually authored semantic annotations nor privileged traces; unavoidable interface mismatches are declared and analyzed separately.

Table 2(b,c) reports ScriptAgent and VBench-Long scores for the same generator comparison and test set. VBench-Long measures complementary perceptual properties, whereas ScriptAgent provides an additional story-level critic view; neither evaluator is combined with CineScope-Metric or used as optimization feedback.

B.1.5 Self-Evolution Evaluation Protocol

Three consecutive evolution rounds update the policy from Round 0 through Round 3. For round g, the multi-story collection set D_g and current memory M_{g-1} induce recurring patterns $\mathcal{B}_g$ and compiled candidates $\mathcal{C}_g$ before fresh replay is revealed:

$$(\mathcal{B}_g, \mathcal{C}_g, M_g) = \mathrm{Compile}\Big(H^{(g-1)}; D_g, M_{g-1}\Big),$$
$$\Delta_g^\star = \mathrm{Admit}\Big(H^{(g-1)}; \mathcal{C}_g, R_g^{\mathrm{det}}, R_g^{\mathrm{vis}}\Big),$$
$$H^{(g)} = \begin{cases} H^{(g-1)} \oplus \Delta_g^\star, & \Delta_g^\star \neq \varnothing, \\ H^{(g-1)}, & \Delta_g^\star = \varnothing. \end{cases}$$

For the reported sequence, replay admission uses source-stratified random sampling from the fixed replay-admission pool. Each round's R_g contains 15 stories: five randomly selected from each of CineScope-Data, AnimeShooter, and ViStoryBench. Sampling within each source is without replacement across Rounds 1–3, so R_1, R_2, R_3 are mutually disjoint. Each set is revealed only after $\mathcal{C}_g$ and its selection rule are frozen. Structural admission uses $R_g^{\mathrm{det}} = R_g^{\mathrm{diag}} \cup R_g$, where diagnostic failures are additional fixed checks and do not count toward the fresh replay sample. For a stochastic-impact candidate, the same story identifiers define $R_g^{\mathrm{vis}} = R_g$, evaluated through fresh independent paired generation and scoring replicates. Neither replay identifiers nor their artifacts can update M_g, induce patterns, compile candidates, enter final system testing, or influence stopping; they are consumed only by the candidate-admission gate.

Once $H^{(g)}$ is immutable, seen and unseen measurements are generated in a stateless, write-blocked copy of that checkpoint. The seen measurement reuses only the corresponding collection story identifiers; its newly generated videos and all measurement artifacts are excluded from M_g and later rounds. The unseen measurement uses the separate fixed 45-script system-test split U. The three-round allocation and stopping rule are hash- and time-stamped before any checkpoint report is unsealed, and unseen reports are revealed only after Round 3 is frozen. No task, video, trajectory, review finding, validator record, CineScope-Metric score, ScriptAgent score, or standardized review-audit output from U, and no post-round measurement artifact from the seen set, is available to any self-evolution component.

Globally frozen review-workload audit. Let M_{audit} denote a reviewer snapshot fixed before Round 0 is evaluated. It includes the reviewer model and prompts, call-trigger thresholds, issue taxonomy and decision thresholds, routing, deduplication, retry policy, and maximum Review-Rewrite attempts. For every seen and unseen checkpoint test, the operational reviewer fields of $H^{(g)}$ are temporarily overridden by M_{audit} inside the write-blocked evaluation copy; the override, generated trajectory, findings, repair records, and scores are discarded after measurement. The next collection/evolution round resumes from the unchanged policy $H^{(g)}$, whose production thresholds and review routes remain editable. Review calls count production-trajectory review and fixer invocations triggered under M_{audit}, and Review issues follow its fixed deduplication policy. Both workload measures are reported per internal production chapter (episode), independently of the complete-video unit used for outcome evaluation. Both exclude initial generation, post-hoc CineScope-

Metric/ScriptAgent evaluation, pattern compilation, and replay. The manifest records the snapshot hash, verifies that it is identical across Round 0–Round 3, and logs that every CaseMemory, policy, threshold, prompt, and replay-pool write interface was disabled during evaluation.

B.1.6 Implementation Details of Ablation Variants

Shared settings and branch isolation. The Round 0 row denotes CineForge-Produce without persistent evolution, and Full is the Round 3 anchor. Full and all four variants start independently from the same H_0 and run three evolution rounds on the same three 20-story collection batches, corresponding replay-admission splits, and fixed 45-story system-test split. Models, generation settings, outcome evaluators, globally frozen review-workload configuration, and two-generation/two-evaluation aggregation are shared. Each branch maintains separate policy versions, CaseMemory, patterns, candidates, admission records, and deployment history; no evolution artifacts or updated states are transferred between branches. System-test is read-only and cannot update memory or affect candidate selection, checkpoint selection, or stopping.

Candidate-proposal/search budgets, per-pattern and per-round candidate caps, selection rules and tie-breakers, and the limit of one deployed patch per round match Full. Actual candidate counts may differ: confidence penalties use each branch's full frozen $K_r = |\mathcal{C}_r|$, not Full's count or the number surviving earlier checks. Applicable stochastic admission uses the same fresh-story count and paired-generation configuration, and final system-test generation budgets are identical. Removing deterministic structural replay does not remove or add final-test video generations. Candidate sets, selection rules, and tie-breakers are frozen before fresh admission data are accessed.

Without trajectory-grounded localization. We disable dependency-based root-cause tracing and reassignment to the earliest evidence-supported stage. Instead, a fixed observation-to-stage mapping targets the stage producing the affected artifact for pipeline/schema/state findings, the corresponding prompt/rendering stage for prompt or local-video findings, and composition or the corresponding global production stage for global-video findings. Issues without an editable target are filtered out. The canonical trajectory, multi-level review, failure evidence, cross-granularity deduplication, lineage, and evidence used for admission/regression checking are retained. Stage-specific critics are routed by the observation stage and retain failure-family normalization, repair proposals, confidence, lineage, and clustering information, but cannot override the stage assignment or reintroduce upstream localization. Pattern induction, typed compilation, risk classification, and all diagnostic, fresh structural, protected-guard, off-slice, structural-margin, and applicable stochastic-admission checks remain unchanged. The changed target is thus an observation-stage proxy rather than a trajectory-localized cause.

Without cross-story recurrence. We replace pattern eligibility $|E_P| \geq k_g$ with $|E_P| \geq 1$: one qualifying story-level production case can trigger candidate compilation. Confidence and policy-editability filters remain unchanged, and issues must retain an unresolved or stuck repair status. Patterns still use (σ, g, χ) and pass through the same case-to-pattern-to-policy abstraction and compilation. Deduplication allows at most one supporting case per story for a given pattern, including repeated findings, repeated locations, and overlapping pipeline/local/global detections. Evaluation and replay replicates neither enter CaseMemory nor add support. CaseMemory retains the same case schema, trajectories, issues, repair actions/outcomes, confidence, lineage, and cross-round accumulation; only its recurrence-based eligibility threshold changes. Candidate budgets, risk/dependency handling, and all applicable admission checks match Full.

Free-form policy lessons. Each eligible recurring pattern produces one short natural-language production rule instead of a typed field-level patch, using the form "[target stage] Lesson: …". For example, a state-propagation lesson instructs the next-shot prompt to preserve a re-entering character's latest position, facing direction, and visual anchor unless the script specifies a change. Localization and recurrence remain unchanged. The lesson is appended only to the target stage's policy-lessons prompt region, not to a global prompt or a separate global experience library. Candidate metadata retain the target stage, source pattern, risk level, and rollback/previous-version information; typed edit operations, field-level payload schemas, and their type/edit-operation checks are removed. The dependency scope conservatively covers the target stage and its declared downstream dependencies rather than a precise field-level slice.

Risk requirements, applicable schema/ID/coverage/state validators, diagnostic and fresh structural replay, structural confidence margins, protected guards, and off-slice checks remain, with side

effects evaluated against the stage-level scope. A lesson requiring fresh LLM, image, or video calls is classified as stochastic-impact and must additionally pass fresh paired stochastic admission; generated outputs are not treated as deterministic exact replay. Each candidate contains one independent rule, under the shared candidate and search caps, without extra optimization calls. No additional per-lesson token-length limit is imposed. The shared generation and candidate-search budgets remain unchanged.

Without fresh structural replay. Deterministic checking uses only R_r^{diag}, containing motivating failures and near-miss cases, instead of Full's $R_r^{\mathrm{det}} = R_r^{\mathrm{diag}} \cup R_r$. Diagnostic schema/ID validity, coverage, duplicate/conflict, state and spatial/facing consistency, duration, and other production constraints remain. Protected hard validators, risk-specific protected-metric regression tolerances, and off-slice side-effect limits are checked only on R_r^{diag}, without fresh-set structural screening. We omit the fresh structural mean $\widehat{G}_r$, its confidence radius, and the corresponding structural-margin gate; the variant therefore does not inherit Full's fresh structural generalization certificate.

Fresh stochastic admission is unchanged: stochastic-impact candidates still use $R_r^{\mathrm{vis}} = R_r$, the same independent paired stories, generation/scoring procedure, $m_{V,r}$, δ_r^V, and $W_{\mathcal{A}}$, with the branch's full frozen K_r in the confidence penalty. Eligibility requires the risk gate, diagnostic hard constraints, protected guards, off-slice checks, and any applicable stochastic gate. The same frozen selection rule and tie-breaker select at most one eligible candidate; if none is eligible, the incumbent policy is retained. This removes fresh deterministic structural screening and its margin, while preserving diagnostic regression protection and fresh stochastic admission.

B.2 Story-Level Statistics and Checkpoint Results

Story-level estimates. For evaluator e, story i, fixed policy a, generation replicate r, and evaluator call q, the story-level score is

$$\bar{y}_{ia}^{e} = \frac{1}{4} \sum_{r=1}^{2} \sum_{q=1}^{2} y_{iarq}^{e}.$$

Each y_{iarq}^{e} is one evaluation of the complete concatenated video for generation replicate r; internal chapters are not separately scored or reweighted. For review workload, each production run contributes its mean calls or issues per internal chapter (episode), and these two run-level values are averaged within the story. Means give equal weight to stories. A source-stratified story bootstrap resamples story identifiers within each source, retaining all matched systems, checkpoints, generation replicates, and evaluator calls. Percentile 95% confidence intervals use the 2.5th and 97.5th percentiles from 10,000 resamples. Intervals are pointwise and are not simultaneous guarantees across all reported means. They quantify story-sampling uncertainty conditional on the evaluated policies, not variation across independently evolved policies. Generation and evaluation repeats do not increase the independent story count.

Fixed-test checkpoint analysis. Table 5 uses the same system-test stories at every frozen checkpoint. All diagnostic rollouts are write-blocked and use the globally frozen review configuration. Figure 3's seen curves instead use round-specific collection subsets and remain descriptive; they are not paired measurements on a fixed collection set.

Table 5: Checkpoint results on the fixed 45-story system-test split. Cells report story-level means with 95% bootstrap CIs. Review calls and issues are per internal chapter (episode) under the frozen audit, rather than per complete video or total production or evolution cost.

Checkpoint	CineScope [CI]	ScriptAgent [CI]	Review calls [CI]	Review issues [CI]
Round 0	4.02 [3.95, 4.10]	3.91 [3.84, 3.98]	46.7 [42.5, 50.9]	104 [94.6, 113]
Round 1	4.20 [4.12, 4.27]	4.02 [3.95, 4.08]	35.6 [32.0, 39.4]	71.9 [64.4, 79.7]
Round 2	4.33 [4.26, 4.40]	4.09 [4.02, 4.16]	30.7 [27.4, 34.1]	59.3 [52.7, 66.1]
Round 3	4.38 [4.31, 4.45]	4.10 [4.03, 4.17]	29.4 [26.2, 32.7]	54.6 [48.3, 61.0]

Source-specific performance. Table 6 separates the three data sources to show whether aggregate performance is consistent across sources. Each source supplies 15 assigned system-test stories; valid counts and exclusions follow the evaluator-specific coverage protocol below. These means are descriptive and do not by themselves establish a difference between sources. Produce is also the Round 0 reference for the adapted optimizers.

Table 6: Source-specific overall score means. CSD, AS, and VSB denote CineScope-Data, AnimeShooter, and ViStoryBench. GenMAC and VideoWeaver are optimizer adaptations on Produce. CineScope and ScriptAgent retain their separate score scales.

Source	Score	Produce	ViMax	AniMaker	MovieAgent	GenMAC	VideoWeaver	Full
CSD	CineScope	4.13	3.84	3.97	3.71	4.24	4.29	4.48
CSD	ScriptAgent	4.02	3.47	3.48	3.92	4.09	4.14	4.23
AS	CineScope	3.95	3.68	3.75	3.53	4.04	4.12	4.30
AS	ScriptAgent	3.84	3.26	3.32	3.69	3.90	3.97	4.01
VSB	CineScope	4.00	3.71	3.79	3.58	4.08	4.14	4.36
VSB	ScriptAgent	3.88	3.30	3.35	3.74	3.93	4.00	4.06

B.3 External-Evaluator Coverage

External scores reuse the final videos and generation-replicate identifiers of the corresponding CineScope comparison. The frozen evaluator manifest contains implementation/version, model and prompt hashes, wrapper hash, clip boundaries, sampling rate, aggregation weights, supported fields and dimensions, and retry limits. Changing a wrapper requires reevaluating every compared policy under that wrapper.

A dimension is unsupported only when the evaluator's documented interface cannot score it; it is then marked N/A. After bounded retries, a supported dimension without all four required valid call-level scores is excluded from that dimension's primary story mean and flagged as an evaluation failure. Coverage is valid stories divided by assigned stories; the failure rate uses stories with exhausted evaluation failures. Unsupported inputs and generation failures are recorded separately and never silently become zero scores. The evaluation report must state valid counts and exclusions alongside the scores; when coverage differs across policies or dimensions, the affected counts and rates must be disclosed. ScriptAgent Overall and its five dimensions remain separate from VBench-Long's six dimensions, with no cross-evaluator aggregate.

B.4 Review Workload and Resource Protocol

The efficiency evidence concerns review calls and detected issues under the globally frozen audit configuration. These counts exclude initial generation, pattern compilation, replay admission, and post-hoc outcome evaluation. They therefore do not establish lower end-to-end cost, latency, a monetary break-even point, or a higher first-pass yield. Operational production uses each deployed policy's actual review settings and is distinct from this audit.

For reproducibility, the run manifest records model and evaluator versions, generation and optimization budget limits, actual data access, candidate counts, retries, and concurrency. Resource logs distinguish collection generation, trajectory review, candidate search, replay/admission, operational production, and scientific evaluation. Reused outputs are not counted twice, and per-request refinement belongs to production rather than a reusable one-time optimization cost. Limits and actual consumption are recorded separately; the common generation protocol alone does not imply equal total optimization expenditure.

B.5 Realized Production Scale

Table 7 reports total delivered duration and the numbers of episodes, segments, and shots per story-generation attempt on system-test. Duration is measured on the final concatenated video timeline. An episode is an internal text/production chapter, counted from the production manifest rather than as a separately delivered video; a segment is a contiguous production unit within an episode containing one or more shots; a shot is a constituent cinematic shot in the final composition. Counts are taken from composition manifests and include only units represented in playable output, excluding unused plans, discarded candidates, and retries. Segment counts require a consistent mapping to this definition across systems; an unavailable count is marked N/A rather than inferred from the number of generation calls.

Duration and counts are summarized by the median and observed range over attempts with playable output, including incomplete deliveries. These statistics describe production scale, not quality

rankings; larger values are not inherently better. Long-form production claims are restricted to the realized range reported here; these summaries do not establish quality robustness at untested lengths.

Table 7: Realized production scale for all systems in Tables 2 and 3 on the common system-test split. Produce is the Round 0 reference; GenMAC and VideoWeaver are optimizer adaptations on Produce, and Full uses the Round 3 policy. Duration, episode, segment, and shot cells report median [minimum, maximum] per story-generation attempt with playable output. Duration measures the final concatenated video; Episodes counts internal production chapters, while all counts include only units contributing to playable output.

System	Duration (min)	Episodes	Segments	Shots
Produce (Round 0)	12.0 [2.10,28.5]	3 [1,8]	72 [13,174]	170 [25,370]
ViMax	3.52 [1.27,10.2]	1 [1,1]	41.5 [15,120]	41.5 [15,120]
AniMaker	2.18 [0.670,4.05]	1 [1,1]	26 [8,48]	26 [8,48]
MovieAgent	9.76 [1.78,21.6]	1 [1,1]	111 [21,262]	111 [21,262]
Produce + GenMAC	11.6 [2.05,27.9]	3 [1,8]	70 [12,171]	166 [24,363]
Produce + VideoWeaver	12.2 [2.18,29.0]	3 [1,8]	74 [14,178]	175 [27,378]
Full	11.9 [2.12,28.7]	3 [1,8]	73 [13,176]	173 [26,374]

B.6 Detailed CineScope-Metric Protocol

Benchmark item. Let x be the source narrative, r the permitted task package, and V the complete video obtained by concatenating the generated clips from all internal chapters. A CineScope-Metric item is $q = (x, r, V)$. The metric receives the same permitted source materials and final rendered output for every generator; it does not receive the private production trajectory Z, CPPE review findings, evidence memory, candidate rules, or replay decisions. This input boundary makes CineScope-Metric independent of the optimization mechanism.

Score and eligibility contract. Human raters and the VLM use the same integer scale $\{0, 1, \ldots, 5\}$ for every subdimension. Zero denotes complete failure or contradiction only when the subdimension is eligible at the current temporal scope; the absence of assessable evidence in an ineligible window produces exclusion or abstention, not a zero. Five denotes fully consistent support, with subdimension-specific anchors for levels 1-4. Before emitting a score for a 150-second or 30-second window, the evaluator first returns an eligibility decision based on (i) whether the source-text evidence aligns with that video window and (ii) whether the current subdimension can be judged from the available scope. Only eligible windows are scored. The eligibility contract, prompts, and thresholds are frozen before held-out evaluation and are independent of generator identity and the eventual score. For every scored subdimension and temporal scope, the evaluator returns a concise description, supporting or conflicting video timestamps, confidence, and abstention. An evaluation that yields a score without video-grounded evidence is invalid and is rerun under the frozen retry policy.

Twenty-item rubric inventory. Each Global dimension aggregates five directly scored subdimensions:

- **Causal State (CS):** key-event visibility; temporal-order correctness; causal-chain readability; state-change continuity; consequence follow-through and closure.
- **Directorial Orchestration (DO):** shot-function match; visual-focus clarity; plot-serving shot scale and camera movement; spatial-blocking continuity; visual emphasis and beat hierarchy.
- **Pacing and Resource Allocation (PR):** importance-aware duration allocation; key-beat elaboration; information-density comfort; tension curve and editing rhythm; closure and breathing room.
- **Character Arc (CA):** character-identity recognizability; goal and motivation traceability; behavioral and emotional progression; relationship-dynamics progression; arc resolution and individuation.

Multiscale aggregation. CineScope-Metric scores the complete concatenated video and applies the eligibility gate above to aligned, non-overlapping 150-second windows $\mathcal{W}_{150}$ and 30-second windows $\mathcal{W}_{30}$; a final shorter window is retained and duration-weighted when eligible. Both window grids start at the beginning of the final video and do not restart at internal chapter boundaries. The full-video scope, denoted by the subscript full, and these windows produce one score per evaluation call; no chapter-level quality scores are averaged. For criterion k, window w, and scale $t \in \{150, 30\}$,

let $a_{k,w} \in \{0,1\}$ be the frozen eligibility decision and d_w the window duration. Define

$$\bar{s}_{k,t} = \frac{\sum_{w \in \mathcal{W}_t} a_{k,w} d_w s_{k,w}}{\sum_{w \in \mathcal{W}_t} a_{k,w} d_w}, \qquad I_{k,t} = \mathbf{1}\left\{ \sum_{w \in \mathcal{W}_t} a_{k,w} d_w > 0 \right\},$$

with $\bar{s}_{k,t}$ omitted when $I_{k,t} = 0$. The final score renormalizes the frozen temporal weights over available evidence:

$$s_k = \frac{\alpha_{\text{full}} s_{k,\text{full}} + \sum_{t \in \{150,30\}} \alpha_t I_{k,t} \bar{s}_{k,t}}{\alpha_{\text{full}} + \sum_{t \in \{150,30\}} \alpha_t I_{k,t}}, \qquad \sum_t \alpha_t = 1,$$

$$s_g = \sum_{k \in K_g} \beta_{gk} s_k, \qquad \sum_{k \in K_g} \beta_{gk} = 1,$$

where K_g contains the five subdimensions of Global dimension g. Long-range criteria such as arc resolution and causal closure therefore rely primarily on full-video evidence unless an aligned window contains a criterion-relevant milestone; a locally absent ending is never treated as a global failure. Temporal weights α, within-Global weights β, eligibility rules, prompts, abstention behavior, and retries are calibrated only on evaluator-development stories and then frozen. Equal weights are the preregistered fallback when calibration does not improve development alignment. All four Global scores and all 20 subdimension scores, together with per-system criterion-by-scale inclusion, exclusion, and abstention rates, are released; their overall average is used only as a compact summary.

Metric ownership. VBench-Long independently reports subject consistency, background consistency, motion smoothness, dynamic degree, aesthetic quality, and imaging quality (Huang et al., 2024c). These remain attributed to VBench-Long, are not passed through the CineScope-Metric judge, and are never averaged with CineScope-Metric or ScriptAgent. Unsupported dimensions are reported as N/A rather than zero.

B.7 Detailed Production-Policy Components and Risk Levels

This subsection specifies which production-policy components CineForge-Evolve may update and the evidence required for deployment. Evolution is restricted to bounded, stage-local targets; higher-risk or system-level changes require stricter replay, human approval, or remain outside the automatic update boundary.

Editable components. CineForge-Evolve may revise stage prompts and contracts, prompt-rendering templates, and production thresholds such as duplicate-dialogue limits or retry settings. Prompt and contract edits require structural checks on replayed states; configuration edits additionally retain the configuration diff and paired metric changes. Review-rubric or repair-routing edits require localized issue evidence and old/new replay, while genre/style edits require held-out transfer and side-effect checks. Typed state schemas and model/API backends remain fixed. Changes to deterministic validators require manual revision and regression tests; changes to the global stage graph or mandatory checkpoints require manual revision, integration tests, and human approval. Recurrence thresholds, opportunity and replay definitions, admission margins, and confidence budgets remain frozen within a certified sequence; human-approved recalibration starts a new sequence with a new identifier.

Risk levels. L0 covers soft prompt phrasing, such as narration wording; L1 covers production thresholds and configuration; L2 covers stage contracts, genre profiles, and review rubrics; L3 covers safety rules or validator semantics. L0 and L1 require structural replay, with a configuration diff for L1 and an additional stochastic gate when flagged as affecting rendered output. L2 additionally requires visual comparison and impact review. L3 requires human approval. The replay and audit conditions in App. B.9 govern deployment.

B.8 CPPE Structural Validators and Synthetic Replay Support

Deterministic structural metrics. CPPE validators operate separately from rendered-outcome evaluators. For atom set $\mathcal{A}$, coverage map ϕ, and scene blueprint $\mathcal{B}_s$, the fixed replay audit includes

$$\text{Cov} = \frac{|\{a : \exists s, \phi(a) = s\}|}{|\mathcal{A}|}, \quad \text{Dup} = \frac{\sum_a \max(0, |\phi^{-1}(a)| - 1)}{|\mathcal{A}|}, \quad \text{JSONValid} = \frac{\#\text{schema-valid units}}{\#\text{units}},$$

$$\text{HardPass} = \frac{\#\text{units passing all hard rules}}{\#\text{units}}, \qquad \text{Spatial/Facing} = \frac{\#\text{shots consistent with } \mathcal{B}_s}{\#\text{shots}},$$
$$\text{BadCase} = \frac{\#\text{units with a critical/major issue}}{\#\text{units}}.$$

These checker outputs are exactly reproducible only when the affected path after the frozen cache boundary is deterministic, as specified in Sec. 2; otherwise the patch is routed to stochastic admission. MovieNet (Huang et al., 2020), MovieBench (Wu et al., 2025a), Book2Movie (Tapaswi et al., 2015), and a Chinese public-domain set may support supplementary structural transfer analysis but are not members of the paired CineScope-Metric human subset.

Synthetic replay support. A synthetic CPPE review/replay suite provides generated stories, known atom sets, injected omissions, duplications, schema errors, spatial contradictions, and ground-truth structural labels. It tests trajectory localization and recurrence behavior rather than CineScope-Metric human alignment.

B.9 Replay Certificate and Audit Conditions

Replay assessment. Candidate updates are assessed on replay-admission stories that are separate from the final system-test split. The incumbent and candidate policies use the same story inputs, but their productions are generated anew rather than reusing intermediate plans, images, or videos. Structural checks assess whether the resulting artifacts satisfy production constraints, while reviewer-detected issues provide evidence for judging the candidate update. Accepted rules are retained for subsequent productions. Replay-admission stories and artifacts do not enter case memory, pattern induction, or patch compilation.

Deterministic checks and stochastic production. A fixed structural checker can return a reproducible result for a given artifact even when that artifact was generated stochastically. This property does not make a comparison between newly generated productions an exact deterministic replay. Exact replay in Sec. 2 and App. B.13 applies only to affected transformations that are deterministic after a declared cache boundary, with editable rules remaining outside the frozen context. A path requiring a fresh LLM, image, or video call falls outside that exact-replay scope, even when provider seeds are available. Issue changes observed after regeneration are empirical replay evidence and do not alone establish a deterministic structural gain.

Scope of the guarantees. Theorem 1 and Proposition 1 are conditional guarantees. Theorem 1 provides a population guarantee under its stated conditionally i.i.d. sampling assumptions. The reported benchmark instead uses source-stratified sampling without replacement for balanced source coverage, and is therefore evaluated empirically on the declared finite replay pool. Applying them to a deployed update requires the stated sampling assumptions, a complete candidate class and selection rule frozen before fresh admission data are revealed, and the corresponding measured margins. Structural certification additionally requires a valid deterministic trace comparison. Validator definitions, metric weights, protected tolerances, and confidence allocations must be fixed for the certified sequence; a material change in these controls or the target distribution requires reassessing the certificate. Passing artifact constraints or observing fewer issues does not, by itself, establish that every numerical gate in Algorithm 3 has been satisfied.

Requirements for numerical admission verification. For a nonempty candidate class frozen before replay, let $K_g = |\mathcal{C}_g|$ count all candidates, including those later rejected by other checks. The declared replay allocation supplies 15 fresh stories per transition; diagnostic cases and within-story generation or rating repeats do not increase the number of distinct replay stories. When the respective certificate assumptions hold, the confidence penalties and candidate-specific slacks are

$$b_g^R = C\sqrt{\frac{2\log(K_g/\delta_g^R)}{n_{R,g}}}, \qquad b_g^V = 2W_{\mathcal{A}}\sqrt{\frac{\log(2(K_g+1)/\delta_g^V)}{2n_{V,g}}},$$
$$\eta_g^R(\Delta) = \widehat{G}_g(\Delta) - m_g - b_g^R, \qquad \eta_g^V(\Delta) = \widehat{\Delta A}_g(\Delta) - m_{V,g} - b_g^V.$$

Here, $\widehat{G}_g$ must be computed from the defined structural margin Y, and $\widehat{\Delta A}_g$ from paired admission scores under an explicitly specified metric and aggregation rule. Issue counts cannot be substituted for either quantity without defining their mapping, normalization, and bounds. Nonnegative slack verifies only the corresponding numerical margin; protected guards, risk requirements, and off-slice

conditions must be checked separately. An inapplicable gate is N/A, and no logarithmic radius is evaluated for an empty candidate class.

A complete verification requires the actual definitions and weights of Q and the admission score, the frozen configuration, per-story paired measurements, each protected-guard result, off-slice changes, and the candidate and deployment decisions. The aggregate quality and workload results reported in this paper do not establish this complete per-candidate numerical audit. The empirical evidence therefore supports the reported performance of the evaluated policies, without asserting that every deployed update has been numerically certified under all of the above conditions.

B.10 Stochastic, Recurrence, and Transfer Guarantees

The fixed-item structural gate in Sec. 3 is exact only on deterministic replay paths; Theorem 1 separately provides a high-probability population statement under its stated conditionally i.i.d. sampling assumptions. The following results cover bounded stochastic admission, recurrence, and held-out transfer.

Proposition 1 (Sequential high-confidence gate for stochastic admission)**.** *At round t, assume each CPPE stochastic-admission metric is bounded in $[0,1]$, the sum of nonnegative weights is $W_{\mathcal{A}}$, and every incumbent/candidate score is estimated from $n_{V,t}$ independent story-level paired evaluations. Within each story and policy, the two generation replicates and two evaluation calls per replicate are averaged equally before the candidate-incumbent difference is formed. If K_t candidates are compared with one incumbent and $\delta_t^V > 0$, define*

$$s_t = W_{\mathcal{A}} \sqrt{\frac{\log(2(K_t+1)/\delta_t^V)}{2n_{V,t}}}.$$

If the candidate satisfies $\widehat{\Delta A}_t \geq m_{V,t} + 2s_t$, then with probability at least $1-\delta_t^V$ its expected admission-score improvement is at least $m_{V,t}$. Across adaptive rounds, if $\sum_t \delta_t^V \leq \delta_V$, the statement holds simultaneously for every accepted stochastic-impact patch with probability at least $1-\delta_V$. This proposition concerns CPPE admission only and makes no CineScope-Metric claim.

Lemma 1 supplies the recurrence result under conditionally independent case opportunities; a union bound controls family-wise false triggers across monitored pattern families. Correlated operational streams may be calibrated empirically on a held-out no-update stream, but that procedure is not covered by the i.i.d. bound.

Proposition 2 (Complete operational cause-level transfer decomposition)**.** *Let $\bar{\mathcal{C}} = \mathcal{C} \cup \{\texttt{unknown}\}$ be the frozen operational cause set. Under matched pre/post evaluation, let r_c be the pre-update error mass of cause c removed after the update and a_c the post-update error mass of cause c that was absent before the update. Then the exact held-out transfer gain is*

$$\Delta_{\text{transfer}} = \sum_{c \in \bar{\mathcal{C}}} (r_c - a_c) = (r_{c^*} - a_{c^*}) + \sum_{c \neq c^*} (r_c - a_c).$$

Consequently, transfer is positive if and only if total removed error mass exceeds total newly introduced error mass. The earlier expression $\rho_{c^*}\eta_{c^*} - \sum_{c \neq c^*} \gamma_{c \leftarrow c^*}$ is a special case requiring no new target errors, no removed non-target errors, and unchanged unknown mass. An empirical claim additionally requires the lower endpoint of the story-bootstrap 95% confidence interval for Δ_{transfer} to exceed zero.

Evidence for the guarantees. Dependency confinement requires recorded slice support and off-slice changes. Replay and sequential-improvement claims require candidate-class sizes, guard deltas, acceptance margins, and policy-version and accepted/rejected-patch records across rounds. Stochastic admission additionally requires repeated paired scores and its allocated confidence radius. Recurrence control requires opportunity and trigger counts, with no-update data for empirical calibration; cause-level transfer requires removed and newly introduced errors for every cause, including `unknown`. None of these structural guarantees implies improved rendered-video quality: that conclusion depends separately on human alignment and the held-out CineScope-Metric and ScriptAgent results.

B.11 Detailed Method Notes

Production records and update schema. A finding records a production symptom as $\ell = (d, u, o, \mathcal{E}, \mathcal{C}_0)$, where d identifies the discovery stream, such as pipeline, local-video, or global-video review; u is the observation unit, such as an episode, scene, shot, prompt, or clip; o is the symptom, $\mathcal{E}$ is the evidence, and $\mathcal{C}_0$ is the candidate-cause set. Review and diagnosis normalize the finding into a standard issue $e = (\sigma, u, g, o, \mathcal{E}, f, m)$, where σ is the earliest evidence-supported causal stage, g is the normalized failure family, f is the proposed repair specification, and m contains confidence, lineage, operational context, and repair status. Thus, ℓ records what was observed, while e records the diagnosis and its response.

The normalized issues from one production form a production case $\kappa_{r,i} = (x_{r,i}, H_r, Z_{r,i}, V_{r,i}, \mathcal{I}_{r,i}, \mathcal{A}_{r,i})$, where $\mathcal{I}_{r,i}$ is the standard-issue set and $\mathcal{A}_{r,i}$ contains Review-Rewrite actions and outcomes. Validator records, evidence, and provenance remain linked through $Z_{r,i}$.

A recurring pattern is $P = (\sigma, g, \chi, E_P)$, where χ summarizes the conditions under which the failure recurs and E_P is the supporting production-case set. Each production contributes one case, so eligibility requires a predeclared threshold $|E_P| \geq k_g$ over distinct story-level opportunities supported by sufficiently confident, policy-editable issues. An eligible pattern produces a bounded, stage-local policy patch $\Delta = (\texttt{target}, \texttt{edit_type}, \texttt{payload}, \rho, \texttt{rollback})$ and $H' = H \oplus \Delta$, where ρ is the risk level. The typed target belongs to an editable H component, the edit type restricts allowed operations, the payload contains the change, and rollback metadata identifies the incumbent version.

Complete Produce/Evolve interface. For policy version/collection round r and story i, the production and evolution interface is

$$
\begin{aligned}
(V_{r,i}, Z_{r,i}) &= \text{CineForge-Produce}(x_{r,i}; H_r, \zeta_{r,i}), \\
D_r &= \{(x_{r,i}, Z_{r,i}, V_{r,i})\}_{i=1}^{n_{D,r}}, \\
(\mathcal{B}_r, \mathcal{C}_r, \Delta_r^\star, M_{r+1}) &= \text{CineForge-Evolve}\big(H_r; D_r, M_r, R_r^{\text{det}}, R_r^{\text{vis}}\big), \\
H_{r+1} &= \begin{cases} H_r \oplus \Delta_r^\star, & \Delta_r^\star \neq \varnothing, \\ H_r, & \Delta_r^\star = \varnothing, \end{cases}
\end{aligned}
$$

where D_r is the collection batch, M_r is CaseMemory, $\mathcal{B}_r$ is the recurring-pattern set, and $\mathcal{C}_r$ is the candidate-patch set. The replay pools are accessed after $\mathcal{C}_r$ is frozen. CineForge-Produce makes each production observable, while CineForge-Evolve accumulates cross-story evidence and deploys an admitted patch as the next policy version.

Production state and trajectory construction. Given a story x, CineForge-Produce constructs a typed production state Σ by decomposing source obligations into narrative atoms organized across episode, scene, and shot units with narrative, character, spatial, cinematic, and prompt fields. Coverage maps link atoms to shots, while character and scene ledgers maintain identity, position, facing, gaze, visual anchors, and state transitions across scenes. These dependencies support coverage and continuity checks before rendering.

Using Σ and H_r, the production path coordinates reusable assets and references, provider-specific prompts, clip generation, visual-anchor propagation, and final composition. Deterministic contracts validate schemas, identifiers, atom coverage, state consistency, durations, and provider constraints, while bounded Review-Rewrite repairs affected artifacts and records the results. CineForge-Produce returns the video $V_{r,i}$ with the canonical trajectory $Z_{r,i}$, which retains stage inputs and outputs, prompts, asset and clip identifiers, validator results, repairs, provenance, and dependency links for locating the earliest evidence-supported production stage.

Patch compilation details and examples. For each $P \in \mathcal{B}_r$, CPPE maps the diagnosed failure and repair specification to a finite set of typed, stage-local patches; their union forms $\mathcal{C}_r$. A patch may refine a stage contract, add a required field, adjust a production threshold or review route, strengthen a rendering rule, or modify a bounded style profile. Missing narrative atoms, for example, induce coverage constraints, while recurrent spatial inconsistencies induce ledger rules. Each patch records its target, edit type, payload, risk level, dependency slice, and rollback metadata. CPPE freezes $\mathcal{C}_r$,

the selection rule, and the tie-breaker before fresh admission, providing a fixed candidate class for replay.

CineForge-Produce: information-conserving decomposition. A long story is decomposed into narrative atoms and downstream coverage maps. The structural check is completeness and non-duplication: every checkable atom should be assigned to at least one shot and should not be redundantly covered across incompatible units. This makes narrative preservation observable before video rendering.

CineForge-Produce: deterministic scaffold and creative infill. The production path fixes identifiers, schemas, and structural normalizers. LLMs fill bounded semantic fields such as shot intent, visual focus, and narration, while editable prompt-rendering templates remain part of H. Exact replay begins only after an explicit cache boundary and covers downstream deterministic transformations; any patch that requires creative infill or regeneration is handled by stochastic admission.

CineForge-Produce: single-source-of-truth continuity. Scene blueprints, state ledgers, force-vector-to-facing rules, and first-frame chains carry long-range continuity. When continuity failures recur, CineForge-Evolve targets the stage rule that derives the relevant state rather than patching a single prompt.

CineForge-Produce: concrete production graph. Table 8 summarizes the implemented story-to-video path. Each stage consumes versioned typed records, emits auditable artifacts, and runs a stage-appropriate validator before its outputs become eligible for downstream use.

Table 8: Concrete CineForge-Produce graph. LLM and generative calls fill bounded creative fields; deterministic transforms preserve identifiers, dependencies, and validation records.

Stage / owner	Typed input	Typed output	Operation and validation
Narrative and episode planning / planner	Task package: `story_id`, source text, target budget, policy version	Narrative atoms: `atom_id`, event and state delta, episode assignment; episode plan	An LLM fills bounded semantic fields; schema, identifier, completeness, and non-duplication checks validate the plan.
Scene and state planning / planner	Atoms, episode plan, character and spatial state	Scene blueprints, atom-coverage map, versioned character/spatial state ledgers	LLM planning is followed by deterministic identifier and state propagation; causal, spatial, and coverage validators run at the boundary.
Shot design / director	Scene blueprints, state ledgers, coverage map	Shot contracts and storyboard records: `shot_id`, intent, camera, timing, and dependencies	The director call proposes shot-level creative fields; deterministic normalizers enforce required fields, timing, coverage, and facing constraints.
Character and scene assets / asset manager	Shot contracts and versioned character/scene specifications	Asset records, reference identifiers, and provenance links	Asset calls instantiate reusable visual references; identity, specification, availability, and version checks gate their use.
Prompt rendering / renderer	Shot contracts, asset identifiers, and templates in H	Rendered prompt records and backend requests with reference and budget fields	A deterministic template transform renders each request; contract, required-field, reference, and budget validators run before submission.
Image/reference generation / reference builder	Rendered requests and reusable asset records	Shot-specific first frames or reference images and a propagation manifest	The image/reference backend produces visual anchors; accepted references propagate along the declared shot chain and are checked for identity and availability.
Video generation and local Review-Rewrite / generator and reviewer	Rendered prompts, first-frame/reference chain, and generation settings	Clip artifacts, local findings, repair attempts, and outcome lineage	The video backend renders clips; bounded review and rewrite may revise the current artifact, while validity, event-completion, prompt-fidelity, and continuity checks are logged.
Composition and global review / composer and reviewer	Accepted clips, timing plan, transitions, and episode order	Complete concatenated video, composition manifest, and global-review findings	Deterministic concatenation and post-processing preserve order and timing; coverage, duration, causal, pacing, and long-range continuity checks audit the final video.

All versioned plans, ledgers, assets, prompts, model requests and outputs, validator/reviewer findings, repair attempts, and composition records are persisted in the canonical trajectory Z. A first frame or reference selected for a shot is recorded with its provenance and passed through the explicit shot dependency chain. For a proposed policy patch, the dynamic slice begins at the changed field and expands only through declared downstream dependencies. Paired replay fixes the task input, unaffected upstream artifacts, validator versions, and recorded exogenous conditions at the cache boundary, then recomputes deterministic descendants. Any slice that requires fresh LLM, image, or video generation is excluded from exact replay and evaluated through stochastic admission instead.

CineForge-Evolve: production trajectory to evidence-grounded case. CPPE inspects both the intermediate artifacts in the complete story-to-video trajectory and the rendered video evidence. Its multilevel discovery pass organizes checks into three evidence streams. The pipeline stream

applies deterministic hard rules to schemas, identifiers, required fields, durations, and prompt contracts; checks local semantic propagation along beat-script-storyboard-prompt chains; inspects reusable character and scene assets; and audits cross-segment state, causal, transition, and duplication consistency. The local-video stream jointly reads the prompts and generated clips within a segment to detect prompt-fidelity, visual, motion, temporal, event-completion, local-continuity, and transition failures. The global-video stream compares the adapted episode script with the composed episode to inspect narrative alignment, completeness, coherence, progression and pacing, shot and cinematic organization, and long-range character and scene continuity.

Every detector emits a LocatedIssue ℓ rather than an immediate policy edit. It records what was observed, where it was observed, the supporting intermediate-artifact or video evidence, and a set of possible upstream causes. For video-side symptoms, the review module maps the observation to its clip and segment and traverses the production chain backward through prompt, storyboard/shot design, script, character/scene assets, and beat/episode planning. The assigned root-cause stage is the earliest stage at which the error is supported by the available evidence or at which a required constraint was first absent; a failure supported only by the rendered output remains attributed to the backend/provider, while insufficient or conflicting evidence yields `unknown`. This separation prevents the location where a defect becomes visible from being conflated with the stage that introduced it.

CPPE next matches duplicate LocatedIssues produced at different granularities, merges their pipeline, prompt/script, storyboard, local-video, and global-video evidence, and preserves the contributing records as lineage. The merged finding is routed by root-cause stage to a stage-specific critic, which normalizes the issue type and severity, confirms or abstains on the root cause, proposes the fix type and fix detail, and assigns a cluster key. The resulting standard issue e therefore links an observed failure to evidence, an operational root cause, and a bounded repair proposal. Deterministic violations remain distinguished from reviewer hypotheses; neither type can authorize persistent deployment without the downstream replay gates.

CineForge-Produce/Evolve interface. A standard issue may trigger bounded Review-Rewrite for the current production and is always logged with its repair outcome. Fixed one-off findings remain auditable but do not support persistent evolution. Only unresolved or `stuck`, sufficiently confident, editable issues that recur across conditionally independent stories or predeclared opportunities enter CineForge-Evolve's case-to-pattern-to-policy path. The root-stage label is an auditable diagnostic hypothesis, whereas replayed target and guard changes provide the behavioral evidence for deployment. CineScope-Metric and the other outcome benchmarks are absent from discovery, diagnosis, repair, case accumulation, pattern induction, patch compilation, and admission.

B.12 Algorithmic Details

Algorithm 1 CineForge-Evolve (Case-to-Pattern-to-Policy Evolution). PatternInduction produces recurring-pattern set $\mathcal{B}_r$, CandidateCompilation produces candidate-patch set $\mathcal{C}_r$, the candidate set is frozen before fresh replay, and at most one accepted patch $\Delta_r^\star$ is deployed.

Input: policy H_r, story collection $\mathcal{D}_r$, memory $\mathcal{M}_r$, replay pools, frozen admission configuration Γ_r (Algorithm 3)
Output: updated policy H_{r+1} and memory $\mathcal{M}_{r+1}$

$\mathcal{M}_{r+1} \leftarrow \mathcal{M}_r$
for each story $x \in \mathcal{D}_r$ **do**
  $(V, Z) \leftarrow \textsc{ProduceVideo}(H_r, x)$
  $\mathcal{L} \leftarrow \textsc{MultiLevelTrajectoryReview}(Z, V)$
  $\mathcal{L} \leftarrow \textsc{TraceRootCauses}(\mathcal{L}, Z, V)$
  $\mathcal{L} \leftarrow \textsc{DeduplicateAndMergeEvidence}(\mathcal{L})$
  for each located issue $\ell \in \mathcal{L}$ **do**
    $e \leftarrow \textsc{StageSpecificDiagnose}(\ell, Z, V)$
    $\mathcal{M}_{r+1} \leftarrow \textsc{StoreCase}(\mathcal{M}_{r+1}, e, \mathit{repair_outcome})$
  end for
end for
$\mathcal{B}_r \leftarrow \textsc{PatternInduction}(\mathcal{M}_{r+1})$
$\mathcal{C}_r \leftarrow \textsc{CandidateCompilation}(H_r, \mathcal{B}_r)$
$\textsc{Freeze}(\mathcal{C}_r, \mathit{selection_rule}, \mathit{tie_break})$; $K_r \leftarrow |\mathcal{C}_r|$
$\mathcal{R}_r \leftarrow \textsc{DrawFreshReplayAfterFreeze}$
$\mathcal{R}_{\mathrm{diag}} \leftarrow \textsc{TriggeringAndNearMissReplay}(\mathcal{M}_{r+1})$
$\mathcal{R}_{\mathrm{det}} \leftarrow \mathcal{R}_{\mathrm{diag}} \cup \mathcal{R}_r$
$\mathcal{R}_{\mathrm{vis}} \leftarrow \mathcal{R}_r$ ▷ fresh paired replicates if stochastic-impact
$\mathcal{A}_r \leftarrow \varnothing$
for each $\Delta \in \mathcal{C}_r$ **do**
  $d \leftarrow \textsc{CandidateAdmission}(H_r, \Delta, \mathcal{R}_{\mathrm{det}}, \mathcal{R}_r, \mathcal{R}_{\mathrm{vis}}, K_r, \Gamma_r)$
  if $d = \textsc{Eligible}$ **then**
    $\mathcal{A}_r \leftarrow \mathcal{A}_r \cup \{\Delta\}$
  end if
end for
$\Delta_r^\star \leftarrow \textsc{SelectAtMostOne}(\mathcal{A}_r, \mathit{selection_rule}, \mathit{tie_break})$
if $\Delta_r^\star$ exists **then**
  $H_{r+1} \leftarrow \textsc{ApplyUpdate}(H_r, \Delta_r^\star)$
else
  $H_{r+1} \leftarrow H_r$
end if
$\textsc{RecordRoundDecisionInAudit}(K_r, \mathcal{A}_r, \Delta_r^\star, H_r, H_{r+1})$
return $H_{r+1}, \mathcal{M}_{r+1}$

Algorithm 2 CineForge-Produce (Invariant-Guided Generation). The procedure records coverage, scene state, prompts, and visual-anchor fields in the canonical trajectory Z.

Input: story x, production policy H, previous visual anchors $\mathcal{M}_v$
Output: complete concatenated video V, typed trace Z, and coverage map ϕ

$\mathcal{A} \leftarrow \textsc{ExtractNarrativeAtoms}(x)$
$\mathcal{P} \leftarrow \textsc{BuildEpisodeSceneShotPlan}(x, H)$
$\mathcal{S} \leftarrow \textsc{DeterministicScaffold}(\mathcal{P}, \mathcal{A})$
$\phi \leftarrow \textsc{BuildCoverageMap}(\mathcal{A}, \mathcal{S})$
if $\textsc{CoverageViolation}(\phi)$ **then**
    $\mathcal{S} \leftarrow \textsc{RepairCoverageDeterministically}(\mathcal{S}, \mathcal{A}, \phi)$
end if
for each scene $s \in \mathcal{S}$ **do**
    $B_s \leftarrow \textsc{BuildSceneBlueprint}(s, H)$
    for each shot u in scene s **do**
        $state_u \leftarrow \textsc{DeriveSpatialState}(B_s, u)$
        $text_u \leftarrow \textsc{CreativeInfill}(u, state_u, H)$
        $prompt_u \leftarrow \textsc{RenderPrompt}(text_u, state_u, H)$
        $clip_u \leftarrow \textsc{GenerateVideo}(prompt_u, \mathcal{M}_v)$
        $\mathcal{M}_v \leftarrow \textsc{UpdateVisualAnchor}(clip_u)$
        $\textsc{Add}(Z, u, state_u, prompt_u, \phi)$
    end for
end for
return $\textsc{MergeClips}(\{clip_u\}), Z, \phi$

Algorithm 3 CandidateAdmission: replay-gated eligibility for one frozen candidate. All applicable checks must pass; only Algorithm 1 selects and commits a policy update.

Input: incumbent H, patch Δ, replay sets $\mathcal{R}_{\text{det}}$, $\mathcal{R}_r$ (fresh only), $\mathcal{R}_{\text{vis}}$, full frozen candidate count K_r, configuration Γ_r
Γ_r fixes $C, m_r, \delta_r^R, W_{\mathcal{A}}, m_{V,r}, \delta_r^V$, protected tolerances $\epsilon_{j,\rho}$,
side-effect limits $gate[\rho]$, and the risk requirements in App. B.7.
Output: eligibility decision d; no persistent policy or memory changes

$H_{\text{cand}} \leftarrow$ ApplyUpdate(H, Δ) ▷ local candidate only
$\rho \leftarrow \Delta.risk$; $paired \leftarrow$ N/A
if $\neg$RiskRequirementsMet(Δ, ρ, Γ_r) **then** ▷ includes required human approval
  RecordRejectionInAudit$(\Delta,$ RiskGateFailure$)$
  return RejectedByRiskGate
end if
$old \leftarrow$ RegressionTest$(H, \mathcal{R}_{\text{det}})$
$new \leftarrow$ RegressionTest$(H_{\text{cand}}, \mathcal{R}_{\text{det}})$
if new violates any hard constraint **then**
  RecordRejectionInAudit$(\Delta,$ HardConstraintFailure$)$
  return RejectedByHardRule
end if
if any protected per-item metric change $\delta_j < -\epsilon_{j,\rho}$ **then**
  RecordRejectionInAudit$(\Delta,$ RegressionDrop$)$
  return RejectedByRegression
end if
if OffSliceChange$(H, H_{\text{cand}}, \mathcal{R}_{\text{det}})$ exceeds $gate[\rho]$ **then**
  RecordRejectionInAudit$(\Delta,$ UnattributedSideEffect$)$
  return RejectedBySideEffectGate
end if
$n_{R,r} \leftarrow |\mathcal{R}_r|$; $\widehat{G}_r \leftarrow n_{R,r}^{-1} \sum_{x \in \mathcal{R}_r} Y_{H,\Delta}(x)$
if $\widehat{G}_r < m_r + C\sqrt{2\log(K_r/\delta_r^R)/n_{R,r}}$ **then**
  RecordRejectionInAudit$(\Delta,$ StructuralMarginFailure$, \widehat{G}_r)$
  return RejectedByStructuralMargin
end if
if StochasticImpact(Δ) **then**
  $paired \leftarrow$ RepeatedCPPEAdmission$(H, H_{\text{cand}}, \mathcal{R}_{\text{vis}})$
  $n_{V,r} \leftarrow |\mathcal{R}_{\text{vis}}|$ ▷ independent stories, not replicate calls
  if $paired.\delta < m_{V,r} + 2W_{\mathcal{A}}\sqrt{\log(2(K_r+1)/\delta_r^V)/(2n_{V,r})}$ **then**
    RecordRejectionInAudit$(\Delta,$ StochasticGateFailure$, paired)$
    return RejectedByStochasticGate
  end if
end if
RecordEligibilityInAudit$(\Delta, old, new, \widehat{G}_r, paired)$
return Eligible

B.13 FULL PROOFS

Definitions and assumptions. For a story x and declared cache boundary b, $c_b(x)$ contains fixed upstream artifacts, task inputs, and validator versions. Editable rules and templates remain in H. When all affected downstream transformations are deterministic, $T_H^d(x; c_b)$ denotes their trace. If a patch requires a stochastic model call, it is excluded from exact replay and handled by Proposition 1. The normalized structural welfare $Q(T) \in [0,1]$, validator set, guard weights λ_j, and trace-field weights w_z are frozen before the certified sequence. With $q_j(T) \in [0,1]$ and $\Lambda = \sum_{j \in \mathcal{G}_\rho} \lambda_j$, define normalized guard debt

$$D_{H,\Delta}(x) = \begin{cases} \Lambda^{-1} \sum_{j \in \mathcal{G}_\rho} \lambda_j [q_j(T_H^d) - q_j(T_{H \oplus \Delta}^d)]_+, & \Lambda > 0, \\ 0, & \Lambda = 0. \end{cases}$$

Thus $D_{H,\Delta} \in [0,1]$. The dependency-confinement term $U_{H,\Delta} \in [0,1]$ is defined below. For fixed $\beta, \gamma \geq 0$,

$$Y_{H,\Delta}(x) = Q(T_{H \oplus \Delta}^d(x; c_b)) - Q(T_H^d(x; c_b)) - \beta D_{H,\Delta}(x) - \gamma U_{H,\Delta}(x).$$

Because the three terms are normalized, an explicit uniform bound is

$$C = 1 + \beta + \gamma.$$

If the deployment distribution changes materially, the replay pool and certificate must be rebuilt; Theorem 1 is stated for fixed $\mathcal{D}$.

Cumulative structural-welfare form. Define

$$\Delta Q_{0:K} := \mathbb{E}_{x \sim \mathcal{D}} Q(T_{H_K}^d(x; c_b)) - \mathbb{E}_{x \sim \mathcal{D}} Q(T_{H_0}^d(x; c_b)).$$

On the simultaneous event in Theorem 1, the accepted rounds $\mathcal{A}$ satisfy

$$\Delta Q_{0:K} \geq \sum_{r \in \mathcal{A}} m_r + \beta \sum_{r \in \mathcal{A}} \mathbb{E}_{x \sim \mathcal{D}} D_{H_{r-1}, \Delta_r}(x) + \gamma \sum_{r \in \mathcal{A}} \mathbb{E}_{x \sim \mathcal{D}} U_{H_{r-1}, \Delta_r}(x).$$

Least fixed-point dependency closure. For each replay item x, let V_x be the finite set of typed trace fields. Each deterministic transformation f_i logs input fields $I_i(x) \subseteq V_x$ and output fields $O_i(x) \subseteq V_x$. Let $S_0(\Delta, x)$ be the fields directly written by the audited support of Δ. Define the monotone map

$$\Phi_{x,\Delta}(S) = S \cup \{z \in V_x : \exists i, \exists y \in S \text{ such that } y \in I_i(x),\ z \in O_i(x)\}.$$

Starting from $S^{(0)} = S_0(\Delta, x)$, set $S^{(r+1)} = \Phi_{x,\Delta}(S^{(r)})$. Since V_x is finite and $S^{(r)} \subseteq S^{(r+1)}$, the sequence stabilizes after at most $|V_x|$ iterations at

$$S_{x,\Delta}^\star = \bigcup_{r=0}^{|V_x|} S^{(r)}.$$

The set $S_{x,\Delta}^\star$ is a fixed point. If S is any fixed point containing S_0, induction gives $S^{(r)} \subseteq S$ for all r, hence $S_{x,\Delta}^\star \subseteq S$. Thus $S_{x,\Delta}^\star$ is the unique least fixed point. Define $\mathrm{Slice}_x(\Delta) = S_{x,\Delta}^\star$. If $C_{H,\Delta}(x) \subseteq V_x$ is the set of changed deterministic trace fields and $W_x = \sum_{z \in V_x} w_z$, set

$$U_{H,\Delta}(x) = \begin{cases} W_x^{-1} \sum_{z \in C_{H,\Delta}(x)} w_z \mathbf{1}\{z \notin \mathrm{Slice}_x(\Delta)\}, & W_x > 0, \\ 0, & W_x = 0. \end{cases}$$

Therefore $U_{H,\Delta}(x) = 0$ implies every changed deterministic field lies in the executed dependency closure of the audited patch support. This proves dependency confinement only; it does not identify the cause of an observed failure.

Theorem 1 (Sequential fresh-replay certificate)**.** *Consider adaptive rounds $r = 0, \ldots, K-1$ for a fixed story distribution $\mathcal{D}$. Conditional on pre-admission history $\mathcal{F}_r$, policy H_r and nonempty*

finite candidate set $\mathcal{C}_r$ are frozen before $R_r = \{\widetilde{x}_{r,i}\}_{i=1}^{n_{R,r}}$ is drawn conditionally i.i.d. from $\mathcal{D}$. Let $Y_{H_r,\Delta}(x) \in [-C, C]$, choose $\delta_r^R > 0$ with $\sum_r \delta_r^R \le \delta_R$, and define

$$G_r(\Delta) = \mathbb{E}[Y_{H_r,\Delta}(x) \mid \mathcal{F}_r], \qquad \widehat{G}_r(\Delta) = \frac{1}{n_{R,r}} \sum_{i=1}^{n_{R,r}} Y_{H_r,\Delta}(\widetilde{x}_{r,i}).$$

If CPPE accepts $\Delta_r^\star$ only when

$$\widehat{G}_r(\Delta_r^\star) \ge m_r + C\sqrt{\frac{2\log(|\mathcal{C}_r|/\delta_r^R)}{n_{R,r}}},$$

then, with probability at least $1 - \delta_R$, every accepted round satisfies $G_r(\Delta_r^\star) \ge m_r$.

Proof of Theorem 1. Fix a round r and condition on $\mathcal{F}_{r-1}$. Then H_{r-1}, candidate-patch set $\mathcal{C}_r$, and all candidate definitions are fixed before replay admission. For any $\Delta \in \mathcal{C}_r$, the replay variables $Y_{H_{r-1},\Delta}(x_{r,i})$ are i.i.d. conditional on $\mathcal{F}_{r-1}$, bounded in $[-C, C]$, and have conditional mean $G_r(\Delta)$. Hence, for any $\epsilon > 0$,

$$\Pr\Big(G_r(\Delta) - \widehat{G}_r(\Delta) > \epsilon \mid \mathcal{F}_{r-1}\Big) \le \exp\left(-\frac{n_r\epsilon^2}{2C^2}\right).$$

Set

$$\epsilon_r = C\sqrt{\frac{2\log(|\mathcal{C}_r|/\delta_r^R)}{n_r}}.$$

A union bound over candidates within round r gives

$$\Pr\Big(\exists \Delta \in \mathcal{C}_r : G_r(\Delta) < \widehat{G}_r(\Delta) - \epsilon_r \mid \mathcal{F}_{r-1}\Big) \le |\mathcal{C}_r| \exp\left(-\frac{n_r\epsilon_r^2}{2C^2}\right) = \delta_r^R.$$

Taking expectation over $\mathcal{F}_{r-1}$ preserves the same unconditional bound. Since $\sum_r \delta_r^R \le \delta_R$, a union bound over rounds shows that, with probability at least $1 - \delta_R$, every candidate in every round satisfies

$$G_r(\Delta) \ge \widehat{G}_r(\Delta) - \epsilon_r.$$

Because this event is simultaneous over the entire finite candidate class in each round, it also holds for the candidate Δ_r selected after replay scores are observed. If CPPE accepts Δ_r, then

$$G_r(\Delta_r) \ge \widehat{G}_r(\Delta_r) - \epsilon_r \ge m_r,$$

which proves the main-text claim.

Expanding $G_r(\Delta_r)$ gives

$$\mathbb{E}_{x\sim\mathcal{D}}\big[Q(T^d_{H_r}(x; c_b)) - Q(T^d_{H_{r-1}}(x; c_b))\big] - \beta\mathbb{E}_{x\sim\mathcal{D}} D_{H_{r-1},\Delta_r}(x) - \gamma\mathbb{E}_{x\sim\mathcal{D}} U_{H_{r-1},\Delta_r}(x) \ge m_r.$$

Rearranging yields the displayed per-round structural welfare inequality. Summing this inequality over accepted rounds gives a telescoping sum on the left:

$$\sum_{r\in\mathcal{A}} \mathbb{E}_{x\sim\mathcal{D}}\big[Q(T^d_{H_r}(x; c_b)) - Q(T^d_{H_{r-1}}(x; c_b))\big] = \mathbb{E}_{x\sim\mathcal{D}}\big[Q(T^d_{H_K}(x; c_b)) - Q(T^d_{H_0}(x; c_b))\big],$$

where rejected rounds have $H_r = H_{r-1}$ and contribute zero. This proves the cumulative form above. No step relates Q to rendered-video utility J; the latter is evaluated empirically. □

Proof of Proposition 1. At round t, let $\widehat{A}_t(H)$ be the weighted empirical CPPE admission score and $\overline{A}_t(H)$ its expectation. Hoeffding's inequality (Hoeffding, 1963; Boucheron et al., 2013) gives, for any fixed policy,

$$\Pr\Big(\Big|\widehat{A}_t(H) - \overline{A}_t(H)\Big| > s\Big) \le 2\exp\left(-\frac{2n_{V,t}s^2}{W_{\mathcal{A}}^2}\right).$$

A union bound over the incumbent and K_t candidates gives a simultaneous deviation bound of at most s_t with probability $1 - \delta_t^V$. Hence the error in a candidate-incumbent difference is at most $2s_t$, and $\widehat{\Delta A}_t \ge m_{V,t} + 2s_t$ implies $\overline{\Delta A}_t \ge m_{V,t}$. A second union bound over rounds gives total failure probability at most $\sum_t \delta_t^V \le \delta_V$. The structural gate is checked separately. □

Proof of Proposition 2. Let N_c^{pre} and N_c^{post} be matched pre/post error masses for $c \in \bar{\mathcal{C}}$. Decompose their difference by errors removed and added:

$$N_c^{pre} - N_c^{post} = r_c - a_c.$$

Summing over all target, non-target, and `unknown` causes yields $\Delta_{\text{transfer}} = \sum_c (r_c - a_c)$. Its sign is positive exactly when total removed mass exceeds total added mass. The targeted decomposition follows by separating c^* from the sum. □

Lemma 1 (Recurrence-trigger reliability)**.** *Theorem 1 starts after a finite candidate set has been produced; this lemma describes when local issues may enter candidate generation. Conditional on the fixed monitoring context, let $X_1, \ldots, X_n$ be independent Bernoulli indicators of whether story-level opportunity l supports failure family g. Let $a = k/n$. Suppose non-systematic failures satisfy $\mathbb{E}X_l \le p_0$, while a systematic policy defect satisfies $\mathbb{E}X_l \ge p_1$, where $p_0 < a < p_1$. If CPPE triggers when $\sum_l X_l \ge k$, then*

$$P_{\text{false}} \le \exp\{-nD(a\|p_0)\}, \qquad P_{\text{miss}} \le \exp\{-nD(a\|p_1)\},$$

where $D(a\|p) = a \log \frac{a}{p} + (1-a) \log \frac{1-a}{1-p}$. Consequently, if M pattern families are monitored, the family-wise false-trigger probability is at most $M \exp\{-nD(a\|p_0)\}$. Choosing

$$n \ge \max\left\{ \frac{\log(M/\alpha)}{D(a\|p_0)}, \frac{\log(1/\beta)}{D(a\|p_1)} \right\}$$

controls family-wise false evolution at level α and a missed systematic defect at level β. The result makes no claim for correlated opportunities; held-out no-update calibration for such streams is an empirical procedure outside this lemma.

Proof of Lemma 1. Let $S_n = \sum_{l=1}^n X_l$. Under the non-systematic null, for any $\lambda > 0$, Markov's inequality gives

$$\Pr(S_n \ge na) \le e^{-\lambda na} \prod_{l=1}^n \mathbb{E}e^{\lambda X_l} \le \exp\{-n[\lambda a - \log(1 - p_0 + p_0 e^\lambda)]\}.$$

Optimizing over $\lambda > 0$ gives $\lambda^\star = \log \frac{a(1-p_0)}{p_0(1-a)}$ and yields $\Pr(S_n \ge k) \le \exp\{-nD(a\|p_0)\}$. A union bound over M monitored families gives the stated family-wise false-trigger bound. Under a systematic defect, apply the same Chernoff argument to the lower-tail event $S_n < na$ with $a < p_1$, obtaining $\Pr(S_n < k) \le \exp\{-nD(a\|p_1)\}$. The sample-size condition follows by setting the family-wise false bound below α and the missed-defect bound below β. □