OBSERVEACT: VALUING EVIDENCE FOR BUDGETED VIDEO PRODUCTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Video production agents make costly decisions from incomplete evidence about the artifacts they create. A critique of a missing event may justify regeneration, a closer inspection, or retaining a correct clip, and the distinction matters when revision can damage existing content. We introduce ObserveAct, a framework that values evidence through the production decisions it enables. A versioned workspace separates artifacts, requirements, and observations, while outcome models compare concrete revisions by their expected benefit, preservation risk, and cost. An action-contrast router selects observations using the alternatives that remain feasible after inspection. We derive an information-value decomposition that exposes the opportunity cost of spending a repair budget on evidence, and align observation supervision with the downstream selector used at deployment. The framework accommodates bounded query bundles and revalidates affected evidence after asset changes. Exact decision diagnostics and a matched-tool evaluation protocol connect observation quality to action utility, harmful revision, and resource use.

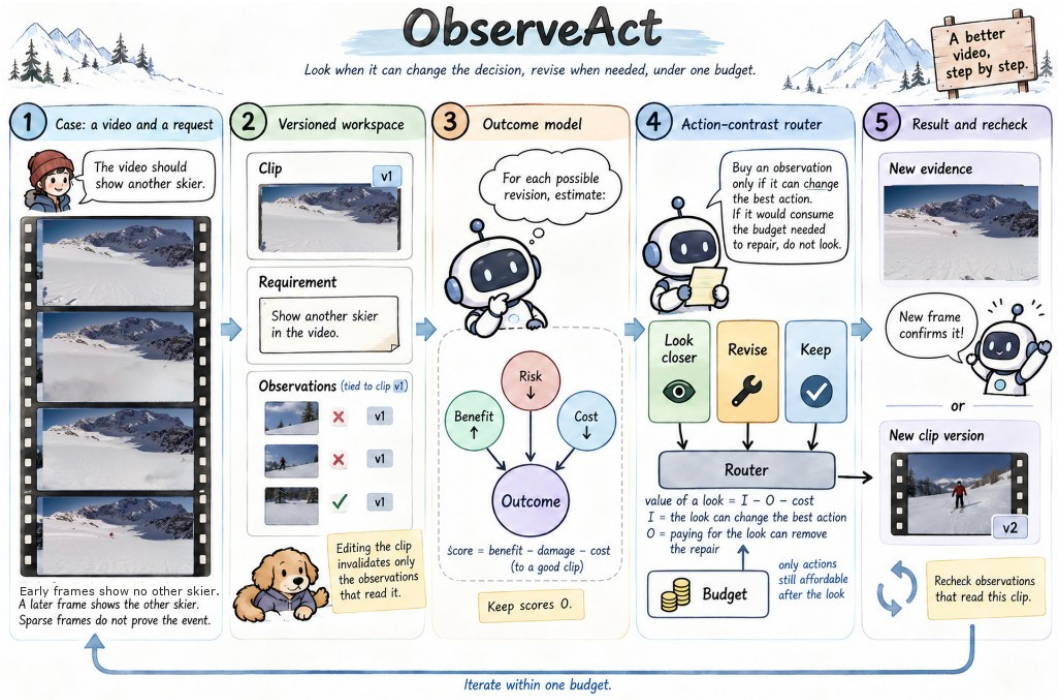


Figure 1: ObserveAct under one budget. The case asks the clip to show another skier. The versioned workspace stores the clip, the requirement, and the observations tied to that version; editing the clip invalidates only the observations that read it. Each revision is scored by benefit, damage, and cost, and keeping the clip scores zero. The router buys a look only when it can change the best action that remains affordable. The value of a look is $I - O - \text{cost}$: I is the change in the selected action, and O is the repair removed by paying for the look. The outcome is new evidence or a new clip version, and observations that read an edited clip are rechecked.

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

1 INTRODUCTION

A video-production agent spends resources both to change artifacts and to understand whether they need changing. Turning a brief into a short film involves a sequence of decisions over scripts, character references, individual shots, sound, and edits, with each decision acting on work that already constrains the next one. Modern generators supply increasingly capable components, and agentic systems organize them into substantial production workflows (Wu et al., 2025; Zhang et al., 2025; Hu et al., 2024; Song et al., 2026; Li et al., 2026). The quality of the delivered film consequently depends on how these components are used, including the occasions on which the most useful action is to inspect an existing asset before revising it.

Consider a request that one character visibly hands a key to another. Sparse frames can show an extended hand and a final possession state without establishing the visible transfer required by the request. A reviewer may need a denser temporal window, a crop of the hands, or a comparison with the input reference. Regenerating immediately spends the same budget that could have resolved this ambiguity and may discard a correct interaction. Conversely, repeatedly inspecting a genuine failure is unproductive when an effective repair is already available. The relevant question is whether a particular observation can change the choice among the actions the agent can still execute.

Existing production agents provide much of the infrastructure needed to ask this question well. MovieAgent and AniME organize hierarchical planning and persistent assets; FilmAgent and Kubrick combine execution with visual feedback; Co-Director searches creative configurations and refines their realization (Wu et al., 2025; Zhang et al., 2025; Xu et al., 2025; He et al., 2024; Song et al., 2026). FrameWorkers adds dynamic task management and descriptor-conditioned routing, while GenMAC, VISTA, and VideoRepair address iterative correction at different granularities (Li et al., 2026; Huang et al., 2024; Do et al., 2026; Lee et al., 2026). These systems motivate a more specific decision problem: given a valid workflow and a fixed set of tools, which evidence is worth acquiring for the next concrete production choice?

We formulate that problem around *decision value*. An observation changes the information available about an asset; a production action changes the asset and may invalidate earlier judgments. Both consume a common resource budget. Classical information value explains why evidence is useful through the decisions it improves (Hay et al., 2012; Callaway et al., 2017), but a production budget introduces an additional coupling: paying for an accurate inspection may remove the very repair that would exploit its answer. We make this lost option value explicit and condition evidence selection on the actions that remain feasible after the query. This connects a video's temporal and spatial ambiguities to the costs and side effects of specific production tools.

ObserveAct implements the connection in a versioned workspace. A lightweight action model predicts the outcome of a fully specified revision, and an evidence router represents each feasible alternative relative to the current incumbent. Its supervision comes from the utility of the actions selected before and after a query, with independent outcome samples and a frozen downstream selector. A query about identity can therefore be valuable when it distinguishes reference repair from resampling, yet unnecessary when neither action is available. The same controller can be invoked at a storyboard, shot, or postproduction node; tool descriptors define the local capabilities, while evidence versions determine which judgments remain usable after a change.

Our evaluation design separates the decision mechanism from the capabilities of the generators. It includes paired production snapshots, matched tools and initial observations, complete cost accounting, and grouped tests of harmful revision and transfer. Exact finite-state diagnostics establish the budget and evidence-complementarity effects. The reported video results are measured outcomes under this protocol.

Our contributions are:

- A production-control formulation that separates versioned artifacts from evidence and values observations against the actions affordable after inspection.
- An action-contrast evidence router with selector-aligned outcome supervision, bounded query bundles, and a common interface for revision and local stopping.
- A budget-aware analysis and reproducible evaluation suite that connect evidence acquisition to decision gain, preservation, and complete-task outcomes.

2 Related Work

Agentic production and refinement. MovieAgent, AniME, and StoryAgent coordinate story, reference, and shot generation, while FilmAgent and Kubrick expose executable production in virtual environments (Wu et al., 2025; Zhang et al., 2025; Hu et al., 2024; Xu et al., 2025; He et al., 2024). Co-Director optimizes creative configurations; FrameWorkers learns routing over a dynamic workspace (Song et al., 2026; Li et al., 2026). GenMAC and VISTA use critique and redesign, VideoRepair localizes changes, SketchVerify evaluates inexpensive motion plans, and Temporal Backtracking Search reuses verified prefixes (Huang et al., 2024; Do et al., 2026; Lee et al., 2026; Huang et al., 2025; Jun et al., 2026). AESR, VideoWeaver, and CineForge further develop repair paths and reusable production experience (Gao et al., 2026; Wei et al., 2026; Liu et al., 2026). ObserveAct builds on these planning, execution, and feedback capabilities and studies the value of acquiring evidence for a particular revision decision. Appendix A maps the relevant interfaces without treating these systems as equivalent implementations.

Information acquisition and decision-dependent prediction. Rational metareasoning evaluates computation by its effect on subsequent decisions (Hay et al., 2012; Callaway et al., 2017); active acquisition methods select informative features or video observations (Ma et al., 2018; 2026). Decision-focused learning aligns prediction with downstream optimization (Donti et al., 2017; Elmachtoub & Grigas, 2017), complementing the reflection and tool-use mechanisms of language agents (Yao et al., 2023; Shinn et al., 2023; Madaan et al., 2023). Our router uses a permutation-invariant set representation (Zaheer et al., 2017; Lee et al., 2019) to connect a query to feasible action contrasts, and its target is the realized utility change of a fixed downstream selector. The production setting additionally requires scoped evidence invalidation, preservation-aware outcomes, and explicit accounting for inspection costs that can change action feasibility.

3 Production State and Decision Objective

The input is a request u, optional user assets, and a budget B_0. A workspace $w = (A, G, \mathcal{K}, E, H, B)$ stores versioned assets A, production dependencies G, requirement contracts $\mathcal{K}$, current evidence E, execution history H, and remaining resources B. The graph records provenance and potential revalidation dependencies. Each requirement retains its source span in u, scope, applicability, and hard or soft status. Evidence records identify the asset versions, temporal or spatial region, and observation procedure supporting a judgment. The complete semantic state z, including whether the artifacts satisfy the request, is only partially observed.

At a production node, the controller considers concrete revisions $a \in \mathcal{A}_B$, observations $q \in \mathcal{Q}_B$, and a local Stay action. Revision descriptors specify parameters, input versions, write scope, validation, and a reserved cost. A revision includes generation, verification, and the declared accept-or-rollback rule; failures retain the previous asset while consuming their actual cost. Observation reads existing assets and produces evidence. A probe that generates a new clip is a production action because it creates content as well as information. A fixed scheduler handles required production stages, invoking the same local controller at each node.

The overall objective is expected terminal requirement quality with preservation penalties under a resource cap. For a local snapshot, we use the incremental utility

$$U_E(a) = \mathbb{E}[\Delta Q(a) - \mu D_{\text{keep}}(a) \mid E] - \lambda c(a), \qquad U_E(\text{Stay}) = 0. \tag{1}$$

Here ΔQ concerns target requirements and D_{keep} concerns separately specified protected content, avoiding an unexplained double penalty. Costs use a single primary unit; time, tokens, and memory are logged separately. Feasibility includes the call and mandatory validation reserve. Local stopping means that no proposed revision has sufficient expected value; successful delivery additionally requires a valid output and all applicable hard requirements to pass. Appendices B and D specify these contracts and outcome denominators.

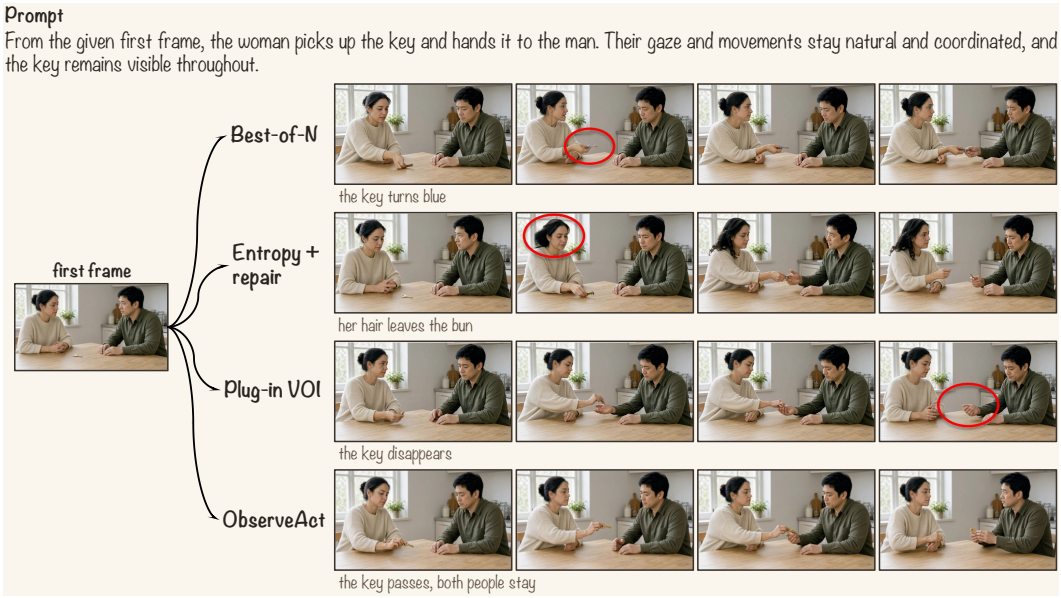


Figure 2: Qualitative comparison on a visible handoff. The shared prompt asks the woman, starting from the given first frame, to pick up the key and hand it to the man, with natural gaze and movement, and with the key visible throughout. Best-of-N turns the key's head blue. Entropy + repair changes her hair before the exchange. Plug-in VOI leaves both hands empty. ObserveAct passes the same brass key from her fingers to his, and both people remain as they were.

4 OBSERVEACT

4.1 VERSIONED EVIDENCE AND CONCRETE ACTION OUTCOMES

A judgment is useful only for the asset and requirement versions it actually addresses. Replacing a character reference does not change the pixels of an old clip, but it can invalidate a claim that the clip matches the *active* reference. ObserveAct therefore separates raw observations from derived requirement judgments. The active validity test checks the versions in a judgment's read set, including reference and contract bindings. A write invalidates affected judgments and schedules revalidation; regeneration is a subsequent decision, not an automatic propagation along every graph edge.

A frozen proposer creates fully specified candidates using the same tool inventory for every controller. The candidate bank and its parameters remain fixed across observations of one asset version; a production or contract change starts a new bank. The action model f_θ encodes the available evidence, requirement scope, action descriptor, and affected assets, and predicts target-quality change, protected-content damage, execution probability, and cost. Its utility estimate is $\hat{U}(a) = \hat{\Delta Q}(a) - \mu\hat{D}(a) - \lambda\hat{c}(a)$. Quality and damage targets include failed and rejected branches; the auxiliary execution probability is not multiplied into these unconditional targets a second time. Only evidence already purchased by the controller enters the visual encoder. Budget filters the concrete candidates rather than changing the predicted outcome of a fixed action. Appendix C gives the head architecture and training objectives.

4.2 INFORMATION VALUE WITH A REMAINING PRODUCTION BUDGET

For a query with cost c_q, let $\mathcal{A}_q = \mathcal{A}_{B-c_q} \subseteq \mathcal{A}_B$ denote the revisions still executable afterward, including STAY. At a local snapshot, fix candidate parameters and let budget affect feasibility. With coherent posterior expectations and an observation that leaves production assets unchanged, define

$$V(q; E, B) = \mathbb{E}_y\left[\max_{a\in\mathcal{A}_q} U_{E,y}(a)\right] - \max_{a\in\mathcal{A}_B} U_E(a) - \lambda c_q. \tag{2}$$

Adding and subtracting the best prior utility on $\mathcal{A}_q$ gives

$$V(q; E, B) = I_q - O_q - \lambda c_q, \qquad (3)$$
$$I_q = \mathbb{E}_y \max_{a \in \mathcal{A}_q} U_{E,y}(a) - \max_{a \in \mathcal{A}_q} U_E(a), \quad O_q = \max_{a \in \mathcal{A}_B} U_E(a) - \max_{a \in \mathcal{A}_q} U_E(a).$$

Both I_q and O_q are nonnegative. The first is classical gross information value on the reduced action set; the second is the option value lost by spending the repair budget. Thus an informative query can be a poor production decision even before accounting for its direct price. Appendix H derives the identity, its estimation bounds, and the extra conditions needed when cost uncertainty changes the feasible set.

The relevant information concerns *action contrasts*. Let $a_q = \arg\max_{a \in \mathcal{A}_q} U_E(a)$. The tower property yields

$$I_q = \mathbb{E}_y \left[\max_{a \in \mathcal{A}_q} \{U_{E,y}(a) - U_{E,y}(a_q)\} \right]. \qquad (4)$$

If a query leaves every action's utility difference relative to a_q unchanged, its gross decision value is zero, even if it reveals uncertain visual details. Equation (4) also motivates an incumbent-anchored representation with a single aggregation over alternatives, avoiding an additive interpretation of overlapping pairwise gains.

4.3 Action-contrast evidence routing and aligned supervision

The router constructs temporal inspections, target crops, and cross-asset comparisons around the requirements that distinguish feasible revisions. For action embedding h_a, query embedding h_q, and the predicted post-query incumbent $\hat{a}_q$, a shared encoder forms

$$r_{qa} = \phi\Big(h_q, h_a, h_{\hat{a}_q}, \hat{U}(a) - \hat{U}(\hat{a}_q), \mathrm{rel}(q, a, \hat{a}_q)\Big), \qquad a \in \mathcal{A}_q, \qquad (5)$$

where rel includes requirement overlap, asset versions, temporal or spatial scope, and prediction uncertainty. A query-conditioned attention pool and a coordinatewise maximum summarize the set:

$$\hat{V}_\varphi(q) = \psi\left(\sum_{a \in \mathcal{A}_q} \alpha_{qa} r_{qa}, \max_{a \in \mathcal{A}_q} r_{qa}, \hat{O}_q, B - c_q \right) - \lambda c_q, \qquad \alpha_{qa} = \mathrm{softmax}_a\, \ell(r_{qa}). \qquad (6)$$

Shared parameters and set aggregation make the score invariant to candidate ordering, subject to a fixed semantic tie rule. The opportunity-loss estimate is an input feature, while ψ predicts the entire change in selected production utility. The final query price is charged exactly once. This direct predictor amortizes decision evaluation without requiring a calibrated generative model of every possible observation.

Its target depends on the deployed action selector. We first fit f_θ on one group of task families and freeze it. On separate query-label families, the same selector chooses a^- from $E, \mathcal{A}_B$ and $a^+(y)$ from $E, y, \mathcal{A}_q$. Independent repeated branch outcomes $Z_r(a)$ then give the supervision

$$Y_q = \frac{1}{R} \sum_{r=1}^{R} \big[Z_r(a^+(y)) - Z_r(a^-) \big], \qquad \mathcal{L}_{\mathrm{query}} = \mathrm{Huber}\big(\psi(\cdot), Y_q\big). \qquad (7)$$

Z_r includes target improvement, preservation penalty, and production cost, while the observation price remains outside this target. Query repetitions sample observation variability; outcome repetitions estimate the consequence of the chosen actions. The records used to evaluate a selected action are not reused to choose it. Freezing the downstream selector preserves the meaning of the labels at deployment; Appendix C describes grouped cross-fitting alternatives and the relabeling required if that selector changes.

Selector changes have a measurable price. The target in Eq. (7) is a property of a selector and an evidence state. Let $G_\sigma(q)$ be the query gain under selector σ, and define $d_E(\sigma, \sigma') = |U_E(\sigma(E)) - U_E(\sigma'(E))|$, using the appropriate feasible set in each state. Changing the downstream selector gives

$$|G_\sigma(q) - G_{\sigma'}(q)| \le d_E(\sigma, \sigma') + \mathbb{E}_y d_{E,y}(\sigma, \sigma'). \qquad (8)$$

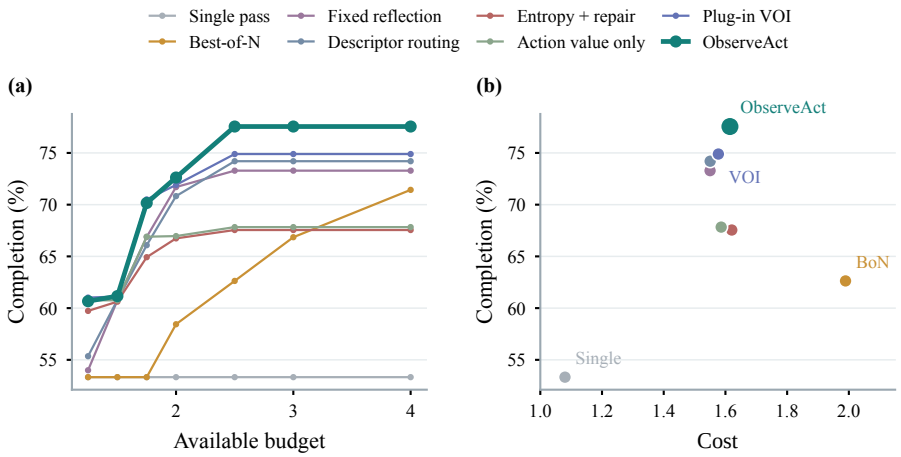


Figure 3: Budget and preservation tradeoffs. (a) Completion versus available total budget. (b) Completion and cost at budget 2.5; preservation profiles are reported in Appendix F. The controller permits bounded observation and one revision; additional sampling continues to benefit from larger budgets.

Table 1: Local-production results. Incremental utility at budget 2.5 initial-generation units. Completion, target coverage, preservation, and harmful revision (HR) are percentages. A dash denotes an undefined HR denominator for a policy that never revises.

Controller	Complete↑	Target↑	Keep↑	HR↓	Cost↓	Utility↑
Single pass	53.3	53.3	100.0	–	1.080	-0.000
Best-of-N	62.6	63.0	98.9	6.3	1.989	0.029
Fixed reflection	73.3	77.3	95.0	7.8	1.550	0.187
Descriptor routing	74.2	78.2	94.9	7.9	1.551	0.196
Entropy + repair	67.5	72.6	93.1	10.7	1.621	0.128
Action value only	67.8	72.9	93.1	10.5	1.586	0.133
Plug-in VOI	74.9	79.3	94.5	8.0	1.577	0.203
ObserveAct	77.6	81.7	94.9	7.7	1.615	0.226

The two terms describe changes to the decision before and after acquisition. Even a better action predictor can alter which observations are useful, so improving it does not preserve the query labels automatically. Freezing the selector makes the target stationary; a later selector upgrade triggers relabeling or an explicitly matched ensemble. The production bank remains fixed across the paired observations, while its evidence-conditioned scores are recomputed.

Prediction error and feasibility error. For an explicit observation model, uniform prior/posterior utility error ϵ, observation-distribution total variation δ, utility range length L, and query-price error ζ_q imply

$$|\widehat{V}(q) - V(q)| \leq 2\epsilon + L\delta + \lambda\zeta_q, \tag{9}$$

provided the feasible action sets agree. A small price error near a budget boundary can instead remove a high-value repair, creating a finite option loss. This distinction motivates explicit feasibility filtering and a separate cost ledger rather than folding every uncertainty into a scalar confidence score. For the direct router, the corresponding quantity is the error in selector-specific decision gain; Appendix H gives the action- and query-selection bounds and the cost-margin conditions.

4.4 Budgeted Execution, Query Bundles, and Stopping

At each node, ObserveAct filters stale or unavailable candidates, reserves costs, and compares query scores with a development-set threshold τ_q. A positive query is executed only when a meaningful continuation remains affordable. Otherwise the controller executes the highest-value revision above τ_a, or keeps the current asset. The mandatory post-action validation is part of the production macro. A rejected branch restores the old asset reference without refunding its cost, and success is assessed against the original requirement contract.

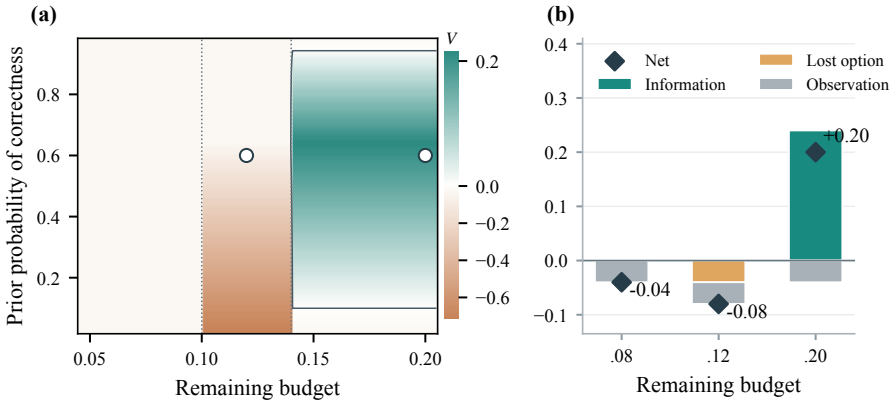


Figure 4: Exact value of inspection. (a) Net observation value across prior correctness and remaining budget in the transfer model. (b) Information, lost repair-option value, and query price at three budgets. The boundaries arise from feasibility, not a fitted temporal schedule.

Two observations can be valuable jointly when neither resolves an action contrast alone. We therefore allow small predeclared bundles that inspect complementary scopes of the same disputed requirement, such as before and after an object transfer. A bundle is scored once using its joint outcome and total cost. It is a bounded candidate extension, with atomic-only routing retained as an ablation. The first implementation limits a node to four concrete revisions plus STAY, four atomic queries, two additional observations, two revisions, and six local decisions; these settings are selected and frozen on development data.

Algorithm 1 ObserveAct at a production node

1: Acquire the common initial evidence once and charge its cost
2: Reuse the version-specific action bank; filter capability, version, and budget constraints
3: Estimate production utilities and net query values on post-query feasible sets
4: If the best query exceeds τ_q, execute it, append evidence, and debit cost
5: Otherwise, execute the best revision above τ_a through its validation macro
6: If neither is selected, return STAY with the current completion status
7: After an adopted write, revalidate affected judgments and rebuild the action bank
8: Repeat within the local decision limit and global budget

The same loop is invoked on a script, reference, shot, or timeline asset. Tool-specific descriptors determine executable operations, while the decision objective and learned heads are shared. This separates interface extensibility from the empirical question of whether outcome predictions transfer, which is tested with held-out stages and backends.

5 EVALUATION

Video study. The study separates local decision snapshots from complete short-film runs. The collection uses 320 snapshots grouped by 40 independent semantic families, with family-level training, development, and test partitions. The 72 task specifications cover 18 development families and are used for requirement review and interface checks; the reported video results come from the snapshot collection. Each snapshot includes the original request, immutable assets, common initial evidence, and repeated outcomes of concrete alternatives. Correct but hard-to-assess clips, genuine generation failures, controlled perturbations, and runtime failures retain separate labels. Complete production tests enter through a brief, references and a script, an existing storyboard, or a video-edit request. Appendices D and G specify independent full-video annotation, grouped uncertainty estimation, and success denominators.

All controllers receive the same tool capabilities, initial evidence, action proposals, and total budget. Baselines include single production, budget-matched Best-of-N, fixed reflection, descriptor routing, uncertainty-based inspection, action values without added observations, and an explicit observation-model VOI controller. Routing and reflection policies are controlled baselines inspired by prior mechanisms; native-system reproduction is reported separately (Li et al., 2026; Do et al., 2026; Song

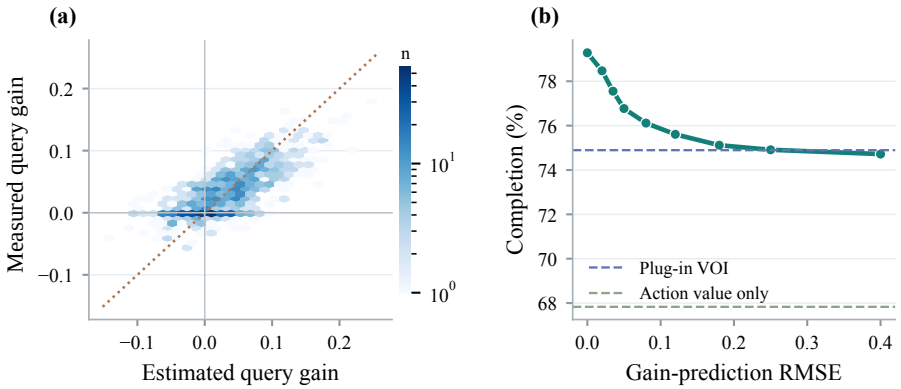


Figure 5: Value estimation. (a) Estimated query gain against the measured query gain. (b) Completion as gain-prediction error increases, with fixed baseline profiles.

Table 2: Mechanism results. The same setting and budget as Table 1; each row changes the stated decision component or error assumption.

Variant	Complete↑	HR↓	Obs. cost↓	Total cost↓
Full controller	77.6	7.7	0.119	1.615
No added observations	67.8	10.5	0.000	1.586
No action-conditioned queries	67.5	10.7	0.035	1.621
Explicit observation model	74.9	8.0	0.069	1.577
No preservation penalty	77.1	9.0	0.116	1.691
Stale version judgments	76.9	7.3	0.120	1.559
Ignore lost repair options	75.2	7.4	0.070	1.569
Larger gain-estimation error	75.4	9.4	0.100	1.636

et al., 2026). VBench and DirectorBench provide complementary diagnostic perspectives, while task success remains tied to the original hard requirements (Huang et al., 2023; Zheng et al., 2025; Chen et al., 2026). Appendix E defines the baseline controls and accounts for planning, observation, selection, retries, and failed calls.

Operating profile. Costs are reported in initial-generation units. The main comparison is at budget 2.5. Appendix F gives the budget, expenditure, and preservation breakdowns.

Failure persistence changes the resource tradeoff. When a missing attribute originates in the reference, repeated calls with that reference share the defect, while a reference-aware revision changes the condition. A cheap timeline edit serves a different regime, where regenerating visual content adds risk without targeting the error. Reporting these strata alongside the aggregate tests whether the controller uses the appropriate action mechanism and whether the overall gain depends on a favorable task mixture.

ObserveAct is above the explicit VOI comparator on completion, while spending more on observation. The comparison tests whether direct prediction of downstream decision gain justifies that added evidence cost. Best-of-N preserves many initially correct assets and improves as additional samples become affordable. Harmful revision is therefore accompanied by the newly introduced failure rate over *all* initially correct cases in Appendix F; a policy can post a low conditional risk by declining to revise.

Which observations are worth buying? The exact transfer example makes the budget coupling visible. With prior correctness 0.6, unobserved repair has terminal utility 0.64. A perfect observation costing 0.04 raises utility to 0.84 when the repair remains affordable, but its net value becomes negative if its cost removes that option. The same evidence and semantic uncertainty therefore support different choices at different budgets. Equation (3) matches direct enumeration to numerical precision over $1{,}500$ randomly instantiated finite decision problems.

The complete finite-case values and query-complementarity check appear in Appendix H. Figure 4 shows where that value changes sign.

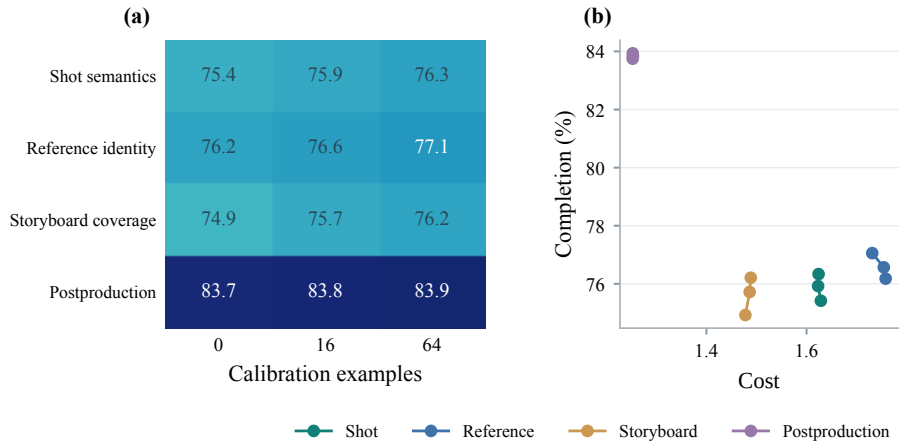


Figure 6: Cross-stage adaptation. Completion and cost for 0, 16, and 64 calibration examples.

Attributing the decision mechanism. Table 2 reports the effects of removing observations, action conditioning, preservation, or version validity, and of increasing gain-estimation error. Figure 5 compares the estimated query gain with the measured gain. It exposes two different risks: noisy acquisition can trigger a harmful change, while stale evidence can suppress a useful one. The version variant can therefore have a lower conditional harmful-revision rate and still complete fewer tasks. These comparisons match observation counts and visual-token budgets, retain a frozen proposer, and compare atomic queries with the same bounded query bundles.

Precision and intervention design. The video study uses paired binary outcomes. If $p_{\neq}$ is the discordance rate between two controllers and Δ_s their success difference, then $\mathrm{Var}(S_A - S_B) = p_{\neq} - \Delta_s^2$. The pilot estimates this variance and its family correlation before fixing the final held-out sample count, directing collection toward independent decision families.

Learned-component ablations distinguish an inference knockout from a retrained alternative. The former keeps the remaining predictors fixed; the latter uses the same training and development budget to recover an alternative policy. Whenever the downstream selector changes, its query labels are regenerated. Both modes retain the concrete production bank and report the resulting query, action, and acceptance distributions, making the source of a performance change inspectable.

Transfer and operating boundaries. The stage study holds the control interface fixed while changing defect prevalence, tool costs, and estimation error. Figure 6 reports completion and cost after 0, 16, and 64 calibration examples. Zero-shot transfer uses a frozen controller, and calibration examples come only from new-domain development families. A new tool name, a new capability description, and a genuinely changed execution model are distinct interventions.

The diagnostics also identify boundaries that a successful implementation must handle. Decision-irrelevant evidence can have high entropy reduction, and two complementary queries can have positive joint value despite negative individual values. Independent outcome evaluation removes the maximum-selection optimism illustrated in Appendix H. Query failures, correlated reviewer errors, hidden events, and unavailable repairs require different responses, detailed through the cases in Appendix I. The complete prompts, execution records, and measurement handoff are specified in Appendix J.

6 Conclusion

We introduce ObserveAct, which makes evidence acquisition part of the production decision itself. By combining budget-feasible action contrasts, outcome-based supervision, and versioned evidence, it gives a common control interface for inspection, revision, and local stopping. Its analysis and evaluation framework establish concrete tests for improving video-production decisions while preserving useful work and accounting for the resources spent.

AI USE STATEMENT

Generative AI tools were used for literature retrieval, source formatting, and checking algebraic derivations. They were not used to produce the reported measurements or to replace review of the claims. We reviewed the assisted material and take responsibility for the final content, including the text and figures. Finite-model calculations and Gaussian-noise diagnostics were executed on CPU.

REFERENCES

Frederick Callaway, Sayan Gul, Paul M. Krueger, Thomas L. Griffiths, and Falk Lieder. Learning to select computations. *arXiv preprint arXiv:1711.06892*, 2017. URL https://arxiv.org/abs/1711.06892.

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*, 2026. URL https://arxiv.org/abs/2605.30090.

Xuan Long Do, Xingchen Wan, Hootan Nakhost, Chen-Yu Lee, Tomas Pfister, and Sercan Ö. Arık. VISTA: A Test-Time Self-Improving Video Generation Agent. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6021–6032, 2026. URL https://arxiv.org/abs/2510.15831.

Priya L. Donti, Brandon Amos, and J. Zico Kolter. Task-based End-to-end Model Learning in Stochastic Optimization. *arXiv preprint arXiv:1703.04529*, 2017. URL https://arxiv.org/abs/1703.04529.

Adam N. Elmachtoub and Paul Grigas. Smart "Predict, then Optimize". *arXiv preprint arXiv:1710.08005*, 2017. URL https://arxiv.org/abs/1710.08005.

Jiayi Gao, Changcheng Hua, Jiaqi Tang, Yuxin Peng, and Yang Liu. Identity-Preserving Text-to-Video Generation via Agentic Enhancement and Semantic Repair. *arXiv preprint arXiv:2608.20749*, 2026. URL https://arxiv.org/abs/2608.20749.

Nicholas Hay, Stuart Russell, David Tolpin, and Solomon Eyal Shimony. Selecting Computations: Theory and Applications. In *Conference on Uncertainty in Artificial Intelligence*, 2012. URL https://aima.cs.berkeley.edu/~russell/papers/uai12-meta.pdf.

Liu He, Yizhi Song, Hejun Huang, Pinxin Liu, Yunlong Tang, Daniel Aliaga, and Xin Zhou. Kubrick: Multimodal Agent Collaborations for Synthetic Video Generation. *arXiv preprint arXiv:2408.10453*, 2024. URL https://arxiv.org/abs/2408.10453.

Panwen Hu, Jin Jiang, Jianqi Chen, Mingfei Han, Shengcai Liao, Xiaojun Chang, and Xiaodan Liang. StoryAgent: Customized Storytelling Video Generation via Multi-Agent Collaboration. *arXiv preprint arXiv:2411.04925*, 2024. URL https://arxiv.org/abs/2411.04925.

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*, 2024. URL https://arxiv.org/abs/2412.04440.

Yidong Huang, Zun Wang, Han Lin, Dong-Ki Kim, Shayegan Omidshafiei, Jaehong Yoon, Yue Zhang, and Mohit Bansal. Planning with Sketch-Guided Verification for Physics-Aware Video Generation. *arXiv preprint arXiv:2511.17450*, 2025. URL https://arxiv.org/abs/2511.17450.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. VBench: Comprehensive Benchmark Suite for Video Generative Models. *arXiv preprint arXiv:2311.17982*, 2023. URL https://arxiv.org/abs/2311.17982.

Sejoon Jun, Zheng Ding, Huangyuan Su, Weirui Ye, and Yilun Du. Temporal Backtracking Search for Test-time Generative Video Reasoning. *arXiv preprint arXiv:2606.13861*, 2026. URL https://arxiv.org/abs/2606.13861.

Daeun Lee, Jaehong Yoon, Jaemin Cho, and Mohit Bansal. Self-Correcting Text-to-Video Generation with Misalignment Detection and Localized Refinement. In *Findings of the Association for Computational Linguistics: ACL 2026*, 2026. URL https://arxiv.org/abs/2411.15115.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam R. Kosiorek, Seungjin Choi, and Yee Whye Teh. Set Transformer: A Framework for Attention-based Permutation-Invariant Neural Networks. In *International Conference on Machine Learning*, 2019. URL https://arxiv.org/abs/1810.00825.

Zhendong Li, Lei Sun, Letian Shi, Deheng Zhang, Ruibo Ming, Mengshun Hu, Dannong Xu, Jian Wang, Danda Paudel, Luc Van Gool, and Jinjin Gu. FRAMEWORKERS: A Dynamic Multi-Agent Framework for AI-Generated Video Production. *arXiv preprint arXiv:2608.29814*, 2026. URL https://arxiv.org/abs/2608.29814.

Junxiang Liu, Lin Wang, Haiyu Shi, Hongxu Ma, Xiaoyu Yang, Chunjie Chen, Xiaoxiao Xu, Kaiqiao Zhan, Boao Wang, Shuizhou Shi, Tianyun Zhu, Jie Li, and Jiangtong Li. CineForge: Self-Improving Agents for Long-Horizon Video Generation. *arXiv preprint arXiv:2608.29621*, 2026. URL https://arxiv.org/abs/2608.29621.

Chao Ma, Sebastian Tschiatschek, Konstantina Palla, José Miguel Hernández-Lobato, Sebastian Nowozin, and Cheng Zhang. EDDI: Efficient Dynamic Discovery of High-Value Information with Partial VAE. *arXiv preprint arXiv:1809.11142*, 2018. URL https://arxiv.org/abs/1809.11142.

Martin Q. Ma, Willis Guo, Aditya Agrawal, Ankit Gupta, Paul Pu Liang, Ruslan Salakhutdinov, and Louis-Philippe Morency. Video Active Perception: Effective Inference-Time Long-Form Video Understanding with Vision-Language Models. *arXiv preprint arXiv:2605.01662*, 2026. URL https://arxiv.org/abs/2605.01662.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-Refine: Iterative Refinement with Self-Feedback. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2303.17651.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language Agents with Verbal Reinforcement Learning. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2303.11366.

Yale Song, Yiwen Song, Nick Losier, Nathan Hodson, Ye Jin, Rhyard Zhu, Yan Xu, Daniel Vlasic, Carina Claassen, Jasmine Leon, Khanh G. LeViet, Zack Chomyn, Joe Timmons, Brett Slatkin, Scott Penberthy, and Tomas Pfister. Co-Director: Agentic Generative Video Storytelling. *arXiv preprint arXiv:2604.24842*, 2026. URL https://arxiv.org/abs/2604.24842.

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*, 2026. URL https://arxiv.org/abs/2606.08091.

Weijia Wu, Zeyu Zhu, and Mike Zheng Shou. Automated Movie Generation via Multi-Agent CoT Planning. *arXiv preprint arXiv:2503.07314*, 2025. URL https://arxiv.org/abs/2503.07314.

Zhenran Xu, Longyue Wang, Jifang Wang, Zhouyi Li, Senbao Shi, Xue Yang, Yiyu Wang, Baotian Hu, Jun Yu, and Min Zhang. FilmAgent: A Multi-Agent Framework for End-to-End Film Automation in Virtual 3D Spaces. *arXiv preprint arXiv:2501.12909*, 2025. URL https://arxiv.org/abs/2501.12909.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing Reasoning and Acting in Language Models. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2210.03629.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Ruslan Salakhutdinov, and Alexander Smola. Deep Sets. In *Advances in Neural Information Processing Systems*, 2017. URL https://arxiv.org/abs/1703.06114.

Lisai Zhang, Baohan Xu, Siqian Yang, Mingyu Yin, Jing Liu, Chao Xu, Siqi Wang, Yidi Wu, Yuxin Hong, Zihao Zhang, Yanzhang Liang, and Yudong Jiang. AniME: Adaptive Multi-Agent Planning for Long Animation Generation. *arXiv preprint arXiv:2508.18781*, 2025. URL https://arxiv.org/abs/2508.18781.

Dian Zheng, Ziqi Huang, Hongbo Liu, Kai Zou, Yinan He, Fan Zhang, Lulu Gu, Yuanhan Zhang, Jingwen He, Wei-Shi Zheng, Yu Qiao, and Ziwei Liu. VBench-2.0: Advancing Video Generation Benchmark Suite for Intrinsic Faithfulness. *arXiv preprint arXiv:2503.21755*, 2025. URL https://arxiv.org/abs/2503.21755.