VERIPHY: AGENTIC PHYSICAL REASONING FOR WORLD MODEL EVALUATION AND REFINEMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Video generative models are increasingly used to simulate physical interactions and support downstream decisions, yet their evaluators still tend to return scalar scores or free-text flaw lists. Such outputs make it difficult to tell which prompt requirement was satisfied, what evidence was resolved, and where physics laws were silently disobeyed. We present VeriPhy, a trace-first critic that evaluates generated video through executable, claim-bound evidence. VeriPhy decomposes the prompt into span-bound claims and a typed check plan. Then it composes a tool chain for each plan, and performs execution that records each check as *measured*, *abstained*, or *errored*. These records are finally assembled into verdicts that determine the physics alignment of each video. This trace supports diagnosis, learning, and refinement. Such a planning architecture also allows the introduction of in context learning capabilities, training free by prompt injection. VeriPhy demonstrated strong learning performance from human annotations, raising the flaws found on held-out clips significantly, and correlates more strongly with human semantic-adherence ratings than VideoScore2. Finally, VeriPhy can be directly wired back into the generation process for trace-guided prompt rewrite, and demonstrated better generation performance training free. Together, these results position executable evidence traces as an interface for auditing, improving, and using physical-video evaluators.

1 INTRODUCTION

Video generative models are now widely used as world models. They serve as world simulators (Brooks et al., 2024; NVIDIA et al., 2025) and as interactive simulators for agents and autonomous driving (Yang et al., 2024; Bruce et al., 2024; Hu et al., 2023a), as planners, policy backbones and data sources in robot learning (Du et al., 2023; Wen et al., 2024; Jang et al., 2025; Huang et al., 2025; Shang et al., 2025), and increasingly as general-purpose visual reasoners (Wiedemer et al., 2025). In each of these roles, the physics consistency of a generated clip becomes crucial as it affects the accuracy of multiple downstream tasks. Current generators often fall short of this, especially on physical dynamics (Kang et al., 2025; Bansal et al., 2026; Gu et al., 2026; Zhang et al., 2026), and at the scale of modern generation, the task of catching such failures falls to automatic evaluators, which select samples at test time, provide rewards for fine-tuning and decide which outputs to repair (Liu et al., 2025; Furuta et al., 2026; Lee et al., 2026).

Most evaluators return a scalar score or a free-text list of flaws (He et al., 2025; Lin et al., 2025). Such outputs make it hard to tell which requirement of the prompt was satisfied, what evidence decided it, and where a physical law was silently violated. When the evaluator is wrong, they also do not show at which step the error arose: in extracting requirements from the prompt, in checking one of them, in obtaining usable evidence, or in combining the evidence into a verdict. Existing evaluators expose parts of this process. TIFA and DSG keep the question-answer pairs they derive from the prompt (Hu et al., 2023b; Cho et al., 2024), VPEval keeps the outputs of its visual modules (Cho et al., 2023), NeuS-V keeps per-proposition scores under a temporal-logic specification (Sharan et al., 2025), and physics benchmarks add rule-level criteria, temporal localization and explicit measurements (Bansal et al., 2026; Zhang et al., 2026; Wang et al., 2026; Cui et al., 2026). We are not aware of an evaluator that records, for every verdict, the prompt span it tested, the check it planned, the evidence with its measurement status, and the rule that combined the evidence.

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.

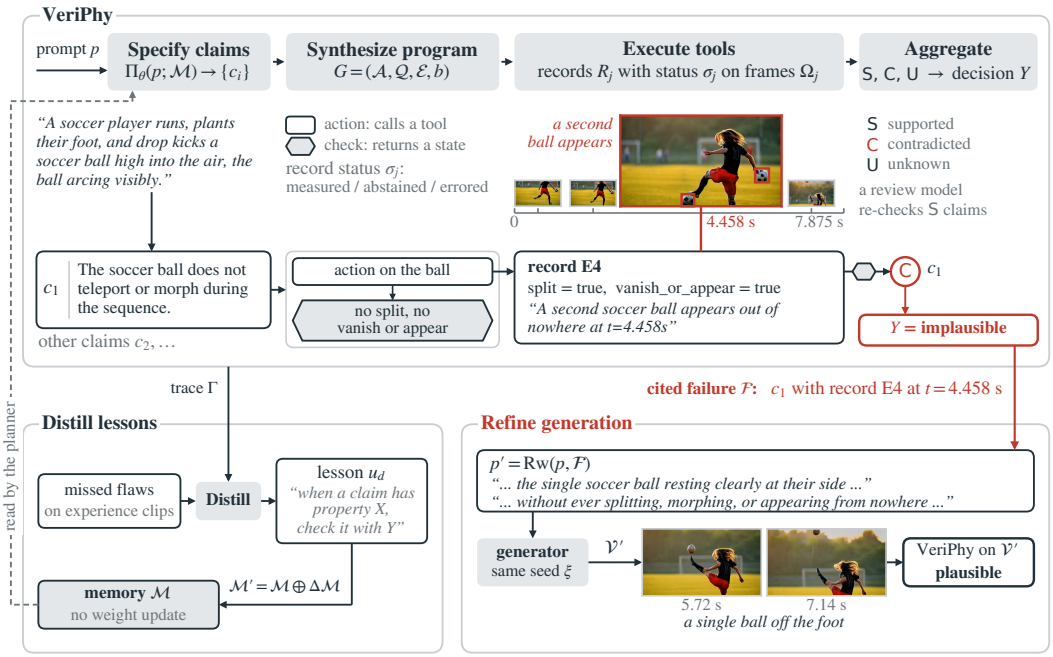


Figure 1: Overview of VeriPhy on one Cosmos3-Nano clip: claims, program, evidence record and decision (Sections 2.1 to 2.4), lesson distillation (Section 2.5), and trace-guided rewriting (Section 2.6).

We present VeriPhy, an agentic critic that evaluates a generated clip by planning and executing physical checks, and returns every verdict together with an executable evidence trace (Figure 1). A planner decomposes the prompt into claims bound to verbatim spans and compiles them into a typed check plan. VeriPhy then composes a tool chain for each plan, calling specialists for grounding, tracking, counting, depth, and on-screen text alongside a dense vision-language verifier, and the result of one call may narrow the scope of the next. Every executed check writes an evidence record marked *measured*, *abstained* or *errored*, so a genuine negative measurement is kept apart from a measurement that could not be made. Fixed three-valued rules assemble the records into claim states and a verdict on the clip. At the end, an advanced vision language model (VLM) double-checks every claim that passed for possible misidentified failures. We keep its changes separate so we can see how much they affect the results.

This planning architecture also makes the critic improvable and usable without training. We compare VeriPhy's reports with flaws marked by human annotators. Most of the flaws it failed to report were not overlooked, as VeriPhy checked the relevant requirement but wrongly judged the video to be correct, often because it checked a weaker version of what the prompt asked. Lessons distilled from these failures and added to VeriPhy's instructions, without retraining, raise the number of annotated flaws it reports on 306 clips not used to write the advice from 340 to 375 of 502. The same records let VeriPhy be wired back into generation and be used for prompt rewrite. Such a scheme demonstrated superior performance in prompt alignment, spatial correlation, and physics consistency compared with native prompt on open-source video generation models, training-free.

Our contributions are as follows.

An agentic critic with executable evidence traces (Section 2). VeriPhy binds every claim to a verbatim span of the prompt, composes and executes a tool chain per plan, records the status of every measurement, and assembles verdicts by fixed three-valued rules.

Diagnosis and in-context learning from the trace (Sections 4.3 and 4.5). The trace attributes each miss to a step of the evaluation, and planner lessons distilled from misses improve recall on held-out clips without weight updates; the gain coincides with finer prompt decomposition. **Evaluation against humans and baselines, and use in generation** (Sections 4.1 and 4.4).

VeriPhy matches the recall of a single call to its own backbone while exposing the evidence behind every verdict, and ranks 200 VideoPhy-2 clips more consistently with human semantic-adherence ratings than VideoScore2 (Spearman 0.465 against 0.327). Trace-guided prompt rewrites turn 49 of 65 of its own implausible verdicts as plausible, with visually perceivable change.

2 VERIPHY

VeriPhy decides whether a generated video satisfies each requirement that its prompt states, including the physical behavior the prompt describes. Its central property is that every verdict can be traced back along one chain: from the words of the prompt, to the claim that restates them, to the checks and tool records that test the claim, to the rule that combines the results. Three design choices build this chain. First, every requirement becomes a claim bound to the verbatim span of the prompt that it tests, so each verdict states which words of the prompt it concerns. Second, every tool call leaves a record whose status says whether the tool could measure, so a failed measurement is never read as a negative one. Third, fixed rules combine the records into verdicts, so every verdict can be recomputed from its records, and a final model review is logged apart from them.

Figure 1 follows one clip through the four stages that implement these choices. The prompt asks a player to drop kick a soccer ball high into the air. Among its claims, the planner states that the ball does not teleport or morph (Section 2.1). The compiler turns this claim into a program that calls a tool on the ball and checks its output (Section 2.2). Execution records that a second ball appears at 4.458 s (Section 2.3). The rules then contradict the claim and judge the clip implausible (Section 2.4). We call the full set of records the evidence trace. The same trace lets VeriPhy improve and act without training: flaws it misses become lessons for its planner (Section 2.5), and failures it cites become a revised prompt for the generator (Section 2.6). Throughout, a video $\mathcal{V}$ is a sequence of timestamped frames with indices $\mathcal{T}$. Each claim takes one of three states, supported (S), contradicted (C) or unknown (U), and each video receives a decision $Y \in \{\textsf{plausible}, \textsf{implausible}, \textsf{abstain}\}$.

2.1 CLAIM-LEVEL SPECIFICATION OF THE PROMPT

A claim fixes one requirement of the prompt and the words it comes from. The planner Π_θ, a language model, reads the prompt p and an experience memory $\mathcal{M}$ of lessons, which stays empty until VeriPhy learns (Section 2.5). It returns claims together with one plan line for each claim:

$$\Pi_\theta(p; \mathcal{M}) = \{(c_i, \pi_i)\}_{i=1}^{N_p}, \qquad c_i = (s_i, \phi_i), \quad s_i \text{ a verbatim span of } p. \tag{1}$$

Each claim pairs a span s_i, copied word for word from p, with an assertion ϕ_i that the video can confirm or refute. Because of the span, a verdict on c_i always names the words of the prompt that it tests. The planner works in two passes so that claims about the same object stay consistent. The first pass gives a fixed name to every object and event occurrence in the prompt, and the second writes the claims in groups of at most four that share these names.

The plan line π_i names the evidence that settles c_i. It uses one of nine operations or a conjunction of them (Table 4). The operations cover general assertions, existence, counts, order and containment in time, spatial relations, motion, on-screen text and audio. A claim about two salmon, for instance, receives both a semantic check and a count check (Figure 5), so the number is counted rather than judged.

2.2 VERIFICATION PROGRAM SYNTHESIS

A deterministic compiler turns the plan lines into a verification program, so that how each claim will be checked is as explicit as the claim itself. The program has two kinds of steps. An action $a_j = (o_j, \alpha_j, g_j)$ calls tool o_j on the video with typed arguments α_j and a scope rule g_j that picks the frames it reads (Section 2.3). A check reads the outputs of earlier steps and returns a state. The compiler expands each operation into the actions and checks it needs. For instance, `before(A, B)` becomes two event-timing actions, two checks that the events occur, and one check that compares their times. The relation *behind* becomes mask and depth actions followed by one comparison. The program is a typed directed graph $G = (\mathcal{A}, \mathcal{Q}, \mathcal{E}, b)$ with actions $\mathcal{A}$, checks $\mathcal{Q}$, and edges $\mathcal{E}$ from each step to the steps that consume its output. The map b binds each check to one claim, and $K_i = \{q \in \mathcal{Q} : b(q) = i\}$ collects the checks of claim c_i.

The compiler validates the program before any frame is decoded. A valid program parses, refers only to defined names, is well typed, has no cycle, and gives every claim at least one check (Equation (7)). If a plan line still fails after a fixed number of re-asks, it falls back to one semantic check of its claim, and the trace records the substitution.

2.3 Tool-Grounded Execution with Evidence Provenance

Execution runs each program on the video and records what every tool returned and whether it could measure at all. Measurement tools settle what can be measured and return numbers. A promptable segmentation model grounds, segments, tracks and counts objects. Motion tools measure the tracked subject after dense optical flow removes camera motion (Section C). A depth model orders objects in depth. A recognition-head OCR engine transcribes rendered text for exact string comparison. A vision-language model settles the rest: it reads densely decoded, timestamped frames to answer semantic questions and to localize events in time.

The executor runs the actions in the topological order of G, so the output of one tool can narrow the input of the next. The scope rule g_j of action a_j reads the records of the actions it depends on and chooses the frames Ω_j that its tool reads:

$$\Omega_j = g_j\big(\{R_k : (a_k, a_j) \in \mathcal{E}\},\, \mathcal{T}\big), \qquad R_j = F_{o_j}\big(\mathcal{V}|_{\Omega_j},\, \alpha_j\big). \tag{2}$$

Here F_{o_j} is the frozen model or program behind tool o_j, and $\mathcal{V}|_{\Omega_j}$ is the video restricted to the frames Ω_j. A timing action, for example, can restrict a later action to the window in which it localized an event.

The record R_j names the video, the prompt, the spans of the claims it serves, the tool and the frames Ω_j it read. Its status σ_j and payload μ_j hold the result, and the status takes one of three values. A `measured` record carries a valid result, including a negative one, such as a complete track that shows no requested motion. An `abstained` record carries the reason why the tool could not measure, such as blur, occlusion or a subject too small to ground. An `errored` record reports an infrastructure failure. The status is what separates "the ball did not move" from "the ball could not be tracked", and the two lead to different verdicts.

2.4 Three-Valued Evidence Aggregation

Fixed rules turn records into states in two steps, so that a verdict follows from its records alone. First, each check q applies a rule ψ_q, its resolver, to the payloads of the actions it reads, $\mathrm{Pa}(q)$, but only if all of them measured:

$$\beta_q = \begin{cases} \psi_q\big(\{\mu_j : a_j \in \mathrm{Pa}(q)\}\big), & \sigma_j = \texttt{measured} \text{ for every } a_j \in \mathrm{Pa}(q), \\ \mathsf{U}, & \text{otherwise.} \end{cases} \tag{3}$$

A tool that could not measure therefore yields an unknown, never a contradiction. Resolvers are deterministic for counts, tracks, transcriptions and time intervals. Semantic checks, event occurrence checks and the depth reading of *behind* and *in front of* use a learned resolver. A learned resolver returns C only when the model reports a visible contradiction, so an inconclusive answer also yields U. Negation and event gating then turn the base state β_q into the check state z_q (Section C.6). A negated check swaps S and C and keeps U. An order check such as `before(A, B)` is C if either event is contradicted and U if either is unknown. Only when both events are supported does it compare their time windows, with a tolerance of a quarter of the shorter window (Equations (8) and (10)). Timing can therefore order two events but cannot establish that they occurred.

Second, the check states roll up into a state y_i for each claim and a decision Y for the video:

$$y_i = \begin{cases} \mathsf{C}, & \exists\, q \in K_i : z_q = \mathsf{C}, \\ \mathsf{S}, & \forall\, q \in K_i : z_q = \mathsf{S}, \\ \mathsf{U}, & \text{otherwise,} \end{cases} \qquad Y = \begin{cases} \textsf{implausible}, & \exists\, i : y_i = \mathsf{C}, \\ \textsf{plausible}, & N_p \geq 1 \text{ and } \forall\, i : y_i = \mathsf{S}, \\ \textsf{abstain}, & \text{otherwise.} \end{cases} \tag{4}$$

A review stage follows. A separate review model re-examines each supported claim and may change it to C, and Equation (4) then recomputes Y. The review can only add contradictions, and the trace keeps each claim's state before and after it, so any run can be re-scored without the review.

These rules have the following properties.

Proposition 1. *Fix a valid program and the answers of the review model. Suppose that some base states β_q change from* U *to* S *or* C*, and that no other base state changes. Then every check state, claim state and decision that was already decided (*S *or* C*;* plausible *or* implausible*) keeps its value, both before and after the review stage. Moreover, before the review stage, $Y =$ plausible if and only if $N_p \geq 1$ and $z_q =$ S for every check q of G.*

Section C.9 proves the proposition by checking each rule in turn. In words, resolving an unknown never overturns a verdict already reached, and a video passes only if every check passes.

2.5 Training-Free Adaptation via Experience Distillation

Because each claim is bound to the prompt, the trace shows where a missed flaw was lost. Often the planner checked the right part of the prompt with a claim weaker than the requirement, for instance one that drops a quantity (Section 4.5). VeriPhy therefore learns from human annotations by editing the planner's instructions, not the weights of any model. The experience memory $\mathcal{M}$ of Equation (1) is a list of lessons, short instructions that the planner reads together with the prompt. The lessons change the claims and programs that the planner writes, and through them the verdicts.

Experience distillation writes $\mathcal{M}$ from experience clips, which are annotated clips set aside for learning. For each annotated flaw d that VeriPhy does not report on such a clip, under the matching of Section 3, a language model reads the flaw record and the evidence trace Γ. The trace shows which claims the planner wrote for that part of the prompt and what each check returned. From this record, the model writes one lesson u_d of the form "when a claim has property X, check it with Y". The update $\mathcal{M}' = \mathcal{M} \oplus \Delta\mathcal{M}$ adds the new lessons $\Delta\mathcal{M}$, groups lessons that give the same advice, and keeps one lesson from each group. Every lesson is readable text.

2.6 Verifier-Guided Generation Refinement

The evidence trace also tells a video generator what to fix. Each failure that VeriPhy cites names a claim, its prompt span, its evidence records and the time at which the evidence lies. When a clip has a nonempty set $\mathcal{F}$ of cited failures, a language model rewrites the prompt from them, $p' = \mathrm{Rw}(p, \mathcal{F})$. The generator then samples a new clip $\mathcal{V}' \sim \mathrm{Gen}(p', \xi)$ with the seed ξ of the original clip, so the prompt is the only input that changes.

3 Experimental Setup

Data. The development core has 149 generated clips with 304 flaws, each a place where annotators found the clip departing from its prompt. The held-out split has 306 clips with 502 flaws, and the lessons come from 117 experience clips. We also use 200 VideoPhy-2 test clips with human semantic-adherence ratings (Bansal et al., 2026), the first 601 clips of the Physion-Eval stream with human glitch reports (Zhang et al., 2026), and 600 Cosmos3-Nano clips generated from VideoPhy-2 prompts.

Metrics. A language model labels each pair of an annotated flaw and a finding as a match or not, and a maximum bipartite matching counts the flaws found. Recall is the share of flaws found.

Implementation. Qwen3-VL-30B-A3B-Instruct (Bai et al., 2025) runs the planner and the semantic checks, SAM 3 (Carion et al., 2026) and further tools run the measurements, and a closed-source model runs the review stage.

Baselines. We compare with a single call to the same Qwen3-VL model, TIFA-style (Hu et al., 2023b) and DSG-style (Cho et al., 2024) question decomposition, VideoScore2 (He et al., 2025), and the critics that Physion-Eval evaluates. Section A gives the full setup.

4 Results

4.1 Agreement with Human Judgments

VeriPhy agrees with human annotators on individual flaws and with human raters on whole clips. On the development core, VeriPhy finds 228 of the 304 annotated flaw entries (75.0%), 191 of them as

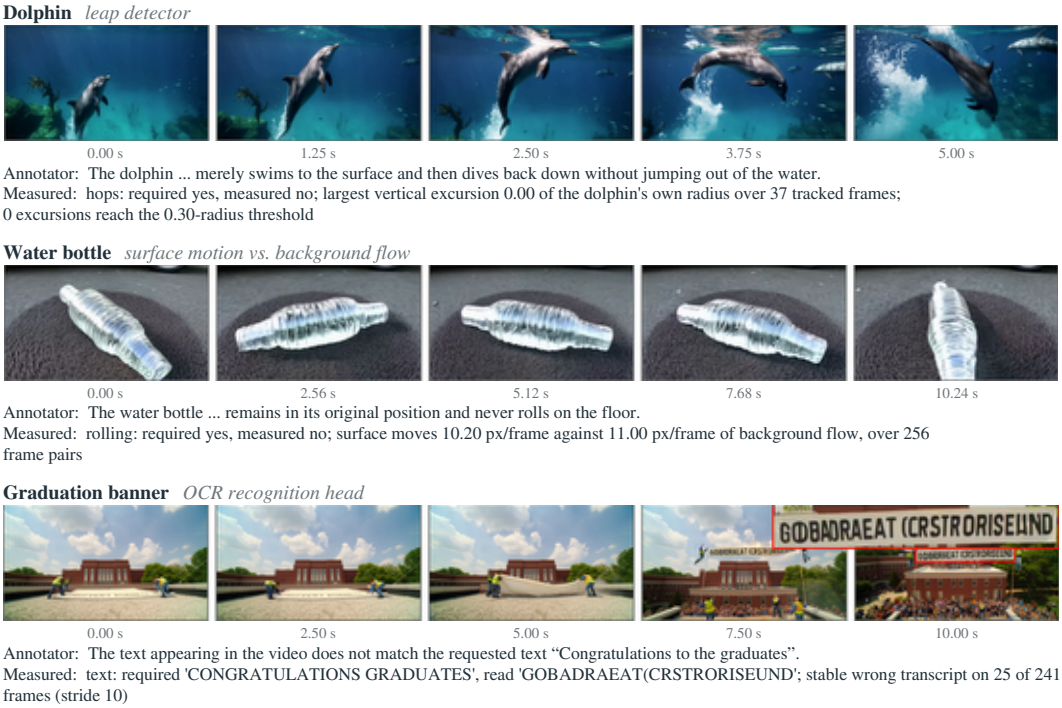


Figure 2: Flaws that VeriPhy finds through measurement tools (three of the six cases in Section H): five frames, the annotator’s flaw sentence and the recorded measurement, which leaves the claim contradicted.

whole matches (Table 1). The single-pass baseline finds 222 (73.0%) with one call. The TIFA-style and DSG-style baselines, which check the prompt one yes/no question at a time, find 164 (53.9%) and 161 (53.0%). VeriPhy finds every annotated flaw on 91 of the 149 clips, and it finds more flaws than the TIFA-style baseline on 70 clips and fewer on 16.

Each finding rests on evidence that a reader can check. Figure 2 shows three flaws that VeriPhy found through a measurement tool in an earlier run on the development core. A hop detector finds no qualifying jump for a dolphin that the prompt asks to leap. The surface of a bottle moves at 10.20 pixels per frame against 11.00 for the background flow, so its apparent rolling is camera motion. The OCR head reads a garbled banner that fails an exact match with the requested text.

VeriPhy also ranks the 200 clips of the VideoPhy-2 sample more consistently with human semantic-adherence ratings than VideoScore2 (Table 2). The Spearman correlation rises by $+0.138$ on all 200 clips and by $+0.309$ on the 55 clips of the hard subset, and both 95% intervals exclude zero.

Table 1: Flaw recall on the development core (149 clips, 304 flaw entries; higher is better). Match rate is the share of findings matched to a flaw. Of the 228 flaws that VeriPhy finds, 83 match findings from the review stage. “–”: not recovered.

Evaluator	Calls/clip	Findings	Match rate	Whole	Found (recall)
Single pass, same backbone	1.0	–	–	–	222 (73.0%)
TIFA-style	21.9	557	29%	125	164 (53.9%)
DSG-style	52.1	–	–	–	161 (53.0%)
VeriPhy	14.1	871	26%	191	**228 (75.0%)**

Table 2: Agreement with human semantic-adherence ratings (1–5) on the VideoPhy-2 sample (higher is better; better value in bold). The last rows give VeriPhy minus VideoScore2 with paired bootstrap 95% intervals.

	All clips ($n = 200$)				Hard subset ($n = 55$)	
Evaluator	Exact	Within 1	PLCC	SRCC	PLCC	SRCC
VideoScore2	**33.0%**	**83.0%**	0.353	0.327	0.239	0.151
VeriPhy	31.0%	76.0%	**0.464**	**0.465**	**0.480**	**0.460**
Difference	–	–	+0.110	+0.138	+0.242	+0.309
95% CI			[−0.018, +0.242]	[+0.005, +0.277]	[+0.018, +0.511]	[+0.050, +0.583]

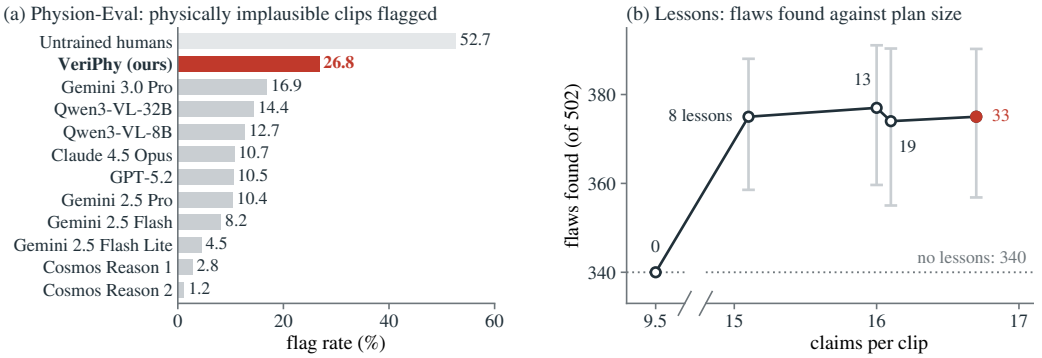


Figure 3: (a) Flag rate for physical implausibility on Physion-Eval, in which 91% of clips carry a human-reported glitch. (b) Flaws found on the held-out split (502 flaws) against claims per clip, one point per lesson set. Whiskers show the 95% interval of the change from no lessons.

4.2 Physical Implausibility Detection on Physion-Eval

VeriPhy flags physically implausible videos more often than every critic that Physion-Eval evaluates (Figure 3a). From its check records, VeriPhy flags 26.8% of the 601 clips as physically unrealistic. The strongest critic in the benchmark, Gemini 3.0 Pro, flags 16.9%. Untrained humans flag 52.7%. A finding of VeriPhy matches the human glitch report on 18.4% of the 543 clips with a report, and 15.5% of all clips receive a finding that explicitly alleges a physical anomaly.

4.3 Training-Free Adaptation from Human Annotations

Lessons distilled from human annotations raise the number of flaws that VeriPhy finds on the held-out split (Figure 3b). With 33 lessons, VeriPhy finds 375 of the 502 flaws (74.7%), against 340 (67.7%) without lessons. With the lessons, VeriPhy newly finds 62 flaws and misses 27 that it found before, a change of +35 flaws (+7.0 points, 95% CI $[+3.4, +10.0]$; sign test $p = 2.7 \times 10^{-4}$).

Most of the gain comes from the first lessons. With 8 lessons from 25 experience clips, VeriPhy already finds 375 flaws. With 13, 19 and 33 lessons, it finds 377, 374 and 375 (pairwise $p \geq 0.8$). By flaw type, the largest gains are for flaws about actions and events (+17 of 211) and about on-screen text (+8 of 73), the two types whose intervals exclude zero (Section D).

The lessons act through the plan, as Section 2.5 describes. With 33 lessons, the planner writes 16.7 claims per clip instead of 9.5, and the number of checks per claim moves only from 1.13 to 1.18. The lessons therefore make the planner split each prompt into finer claims. For on-screen text, text checks rise from 81 without lessons to 114 with 8 lessons and 136 with 33 lessons.

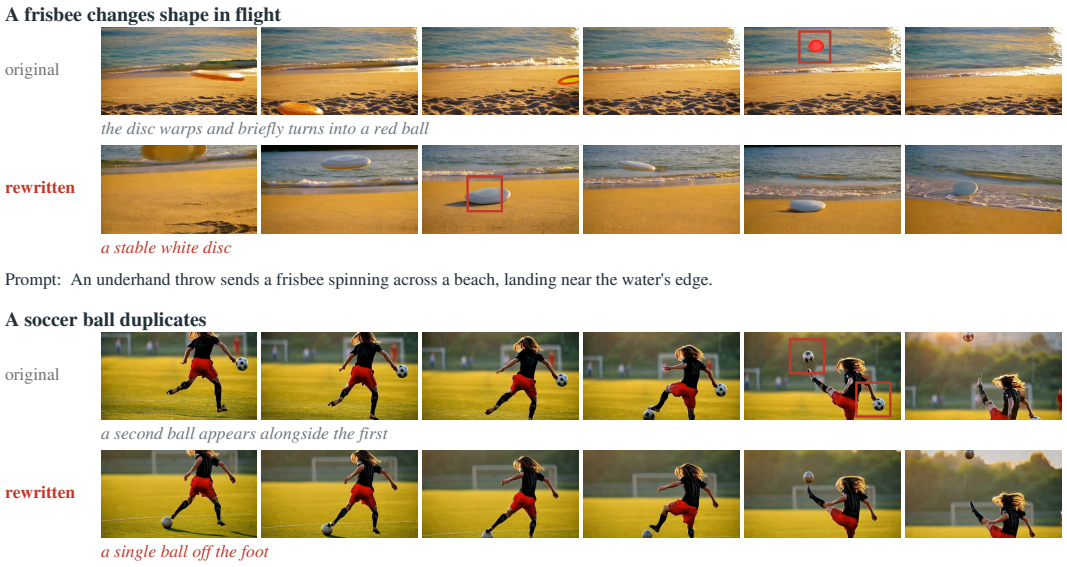


Figure 4: Trace-guided rewriting on Cosmos3-Nano clips: the original clip and the clip regenerated from the rewritten prompt with the same seed. Red boxes mark the cited failure and its repair.

4.4 Verifier-Guided Generation Refinement

One round of trace-guided rewriting (Section 2.6) turns 49 of the 65 clips that VeriPhy first judged implausible into plausible clips (Figure 4). We run the round on 600 Cosmos3-Nano clips.

Figure 4 shows how the cited evidence shapes the rewrite. In the frisbee clip, VeriPhy contradicts three claims about the disc, among them that it keeps a rigid disc shape and follows a continuous trajectory. Its records flag that the disc vanishes and reappears, and the track of the disc breaks between 3.375 s and 5.583 s. The rewritten prompt turns these failures into explicit requirements: the disc keeps "its rigid, solid disc shape without any morphing, splitting, or color shifts" and "never vanishes, teleports, or reappears at a different location". In the soccer clip, VeriPhy cites one claim, that the ball does not teleport or morph, with a record that a second ball appears at 4.458 s. The rewritten prompt places "the single soccer ball" at the player's side and requires the kick to send it into the air "without ever splitting, morphing, or appearing from nowhere". Neither regenerated clip shows the cited failure, and VeriPhy judges both plausible. Section J reports additional results.

4.5 Failure Attribution from Evidence Traces

The evidence trace also shows where VeriPhy misses a flaw. In a run on the development core without the review stage, VeriPhy finds 197 of 286 distinct flaws, 18 more than a variant that verifies one claim at a time (95% CI $[+8, +28]$). Of the 89 flaws it misses, the planner wrote no claim for 2, and the scorer did not credit a contradicted claim for 8. For the other 79, VeriPhy checked the claims without contradicting them, 76 times by a semantic yes/no judgment. In roughly half of these 76 cases, the checked claim is weaker than the annotated requirement, because the plan drops a quantity, a spatial qualifier or part of a compound sentence. These misses arise in the plan, which is where lessons act (Section 2.5).

5 Related Work

Physical-video and world-model evaluation. Video evaluation ranges from per-dimension benchmark suites (Huang et al., 2024; Liu et al., 2024; Sun et al., 2025) and learned or VQA-based scorers (He et al., 2024; 2025; Lin et al., 2025) to question-decomposition methods (Hu et al., 2023b; Cho et al., 2024). Recent work increasingly targets physical consistency, using physical-commonsense

ratings, rule-level criteria, localized errors, or trajectory measurements to assess generated video (Bansal et al., 2025; 2026; Meng et al., 2025; Gu et al., 2026; Motamed et al., 2026; Zhang et al., 2026; Chinchure et al., 2026; Wang et al., 2026). Video generators are also being used explicitly as world models for downstream robot learning (Jang et al., 2025), making evaluator failures consequential beyond benchmark ranking. VeriPhy complements these efforts with a representation rather than a new taxonomy: each judgement is bound to a prompt span, executable check, evidence payload, and explicit measurement status.

Tool use and program-based verification. Plan-first agents and visual programs compile requests into tool calls before observing outputs (Xu et al., 2023; Kim et al., 2024; Surís et al., 2023; Gupta & Kembhavi, 2023), and video question-answering methods answer with programs over perception tools (Choudhury et al., 2024; Min et al., 2024). For evaluation, VPEval exposes modular visual-program outputs (Cho et al., 2023), VideoGen-Eval combines language-model agents with evaluation tools (Yang et al., 2025), and NeuS-V verifies generated video against a temporal-logic specification of the prompt (Sharan et al., 2025). VeriPhy synthesizes a verification program for every claim of the prompt, and each verdict traces back to the records of the tool calls in that program.

Learning from feedback. Agents can convert prior experience into verbal reflections, strategies, optimized context, or structured long-term memory without parameter updates (Shinn et al., 2023; Ouyang et al., 2026; Huang et al., 2026; Latimer et al., 2026). Evaluator feedback also improves video generation through fine-tuning, test-time search, prompt refinement, and localized repair (Furuta et al., 2026; Liu et al., 2025; Xue et al., 2025; Choi et al., 2026; Lee et al., 2026). VeriPhy uses one evidence trace for both: missed flaws become lessons in its planner's memory, and cited failures become a rewritten prompt for the generator.

6 Conclusion

VeriPhy reframes physical-video evaluation as the construction of an executable evidence trace. It turns the prompt into claims, synthesizes a verification program for each claim, records what each tool measured or could not measure, and aggregates the records into claim states. This trace provides a common interface for three tasks that are usually separated: diagnosing evaluator failures, learning from those failures without weight updates, and turning the resulting findings into targeted refinement. Empirically, traces localize the dominant source of observed misses, trace-derived lessons raise held-out flaw recall, and VeriPhy shows stronger rank agreement with human VideoPhy-2 semantic-adherence ratings than VideoScore2. On Physion-Eval, VeriPhy flags 26.8% of clips as physically implausible, against 16.9% for the strongest critic that the benchmark evaluates. The refinement experiment shows that the same records can drive corrective prompts. Evidence traces are thus a practical substrate for building evaluators that can be inspected, improved and used inside world-model development loops. Future work will extend the evaluation to several annotators per clip and to human ratings of the regenerated videos.

AI use statement

Required-disclosure tasks. We used generative AI tools for four of the twelve tasks that require disclosure. (1) *Generating synthetic datasets*: Cosmos3-Nano generated the clips used in the prompt-rewrite experiment; the main corpus reached us already generated. (2) *Implementing methods*: AI coding assistants wrote or modified parts of the VeriPhy and analysis code. (3) *Cleaning or reformatting a dataset* and (4) *qualitative or thematic data analysis*: a language model assigns modality and type tags to each flaw record by majority vote over three zero-temperature predictions (Section E), a text matching judge labels finding–flaw pairs, and a text model distils candidate lessons from traces of missed flaws.

Recommended-disclosure tasks. We also used the same assistant to condense an author-written technical description of the system into this manuscript. The assistant proposed manuscript organization and title candidates, drafted and edited prose, and created or edited figure and table code, including the matplotlib code for Figure 1 and the plotting script for Figure 6.

Models that are part of the method. Generative models are also components of the systems evaluated in this paper, independently of their use during authoring. Qwen3-VL-30B-A3B-Instruct serves as the planner of VeriPhy, as the vision-language model of its semantic checks, and as the

backbone of the three baselines. Tables 5 and 18 identify the models used for the matching judge, type-tag voters, lesson distiller, grounding guard, DSG-style question generator, prompt rewriter, and third-party raters.

Verification and responsibility. We reviewed all AI-assisted work before inclusion in the manuscript. Authors checked the reported numbers in text, tables, and figures against the run records listed in Table 3. The authors reviewed all AI-written code and validated it through deterministic gates. We take responsibility for the final content of this work, including its text, claims, code, and other artifacts produced with generative-AI assistance.

ETHICS STATEMENT

Data provenance and rights. The corpus of 1,500 generated clips and 2,582 flaw records is internal data of the authors' organization. The source generator of each corpus clip is not retained. Figures 2, 6 and 9 show frames of corpus clips.

People depicted and personal data. Some prompts describe people, and the clips may show realistic human figures. All clips in this paper are generated by video models and show no real person. We use the clips only for evaluator research, and the supplement contains no corpus video. The flaw records contain no annotator identity.

Human annotation. A single annotator at a commercial data vendor recorded the flaw annotations. No additional participants were recruited for the experiments in this paper.

Third-party models and data. We use the VideoPhy-2 test split and AutoRater (Bansal et al., 2026) under the MIT licence, VideoScore2 (He et al., 2025) under Apache-2.0 (weights) and MIT (code), Physion-Eval (Zhang et al., 2026) under its research-use-only licence, and Cosmos3-Nano under OpenMDW-1.1. Qwen3-VL-30B-A3B-Instruct (Bai et al., 2025) is released under Apache-2.0 and SAM 3 (Carion et al., 2026) under the SAM License, and the tracking, text-recognition, and sound-event tools in Table 5 are released under Apache-2.0 or MIT. The depth model is released under CC-BY-NC-4.0 and is used here only for non-commercial research. The review model, GPT-5.6 Sol, is accessed under its provider's terms of service.

Dual use. An evaluator that identifies where generated video departs from a prompt can also help make generated video more convincing, including in deceptive applications. VeriPhy evaluates prompt consistency rather than media authenticity and therefore should not be treated as a synthetic-media detector or misuse safeguard. Our prompt-rewrite experiment changes prompts only; the independent rater's aggregate scores move by at most one point (Section J). We release no generator, and no experiment updates generator weights.

Conflicts of interest. The authors declare no conflicts of interest.

REPRODUCIBILITY STATEMENT

Section 2 and Section C specify the claim representation, plan vocabulary and validity predicate, scoped execution, evidence records, resolvers, roll-up, and review stage; Algorithm 1 collects these components into one procedure. Table 5 identifies the model or tool behind each tool role, and Table 18 maps each model prompt to the model that received it. Qwen3-VL-30B-A3B-Instruct, served with vLLM, runs the planner, the semantic checks, and the three baselines, and GPT-5.6 Sol runs the review stage. Section E defines the flaw-record fields, the development core, the machine-assigned type tags, and the matching protocol, and Section B records the population, review configuration, scoring protocol, and cost of every run we report. Section D specifies lesson distillation and the dose design, Section F the baselines, and Section J the prompt-rewrite experiment. Table 1 and Section F report model and tool calls per clip.

The supplement contains 22 Cosmos3-Nano clips of the prompt-rewrite experiment before and after rewriting, with the records of VeriPhy for them, the prompts of the planner, the semantic checks, and the matching judge of run R5, and the lesson file with all 33 lessons. VideoPhy-2 (Bansal et al., 2026) and VideoScore2 (He et al., 2025) are public, and their prompts, ratings, and models are available from their releases; Physion-Eval (Zhang et al., 2026) releases its clips and human glitch reports for non-commercial research.

REFERENCES

James F. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843, November 1983. ISSN 1557-7317. doi: 10.1145/182.358434. URL http://dx.doi.org/10.1145/182.358434.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report, 2025. URL https://arxiv.org/abs/2511.21631.

Hritik Bansal, Zongyu Lin, Tianyi Xie, Zeshun Zong, Michal Yarom, Yonatan Bitton, Chenfanfu Jiang, Yizhou Sun, Kai-Wei Chang, and Aditya Grover. Videophy: Evaluating physical commonsense for video generation. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 102075–102121, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/fce2d8a485746f76aac7b5650db2679d-Paper-Conference.pdf.

Hritik Bansal, Clark Peng, Yonatan Bitton, Roman Goldenberg, Aditya Grover, and Kai-Wei Chang. Videophy-2: A challenging action-centric physical commonsense evaluation in video generation. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 118456–118470, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/c02f6a1d5c55e16db50d339dad905b4d-Paper-Conference.pdf.

Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators. 2024. URL https://openai.com/research/video-generation-models-as-world-simulators.

Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, Yusuf Aytar, Sarah Maria Elisabeth Bechtle, Feryal Behbahani, Stephanie C.Y. Chan, Nicolas Heess, Lucy Gonzalez, Simon Osindero, Sherjil Ozair, Scott Reed, Jingwei Zhang, Konrad Zolna, Jeff Clune, Nando De Freitas, Satinder Singh, and Tim Rocktäschel. Genie: Generative interactive environments. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 4603–4623. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/bruce24a.html.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris Coll-Vinent, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, Jie Lei, Tengyu Ma, Baishan Guo, Arpit Kalla, Markus Marks, Joseph Greer, Meng Wang, Peize Sun, Roman Rädle, Triantafyllos Afouras, Effrosyni Mavroudi, Katherine Xu, Tsung-Han Wu, Yu Zhou, Liliane Momeni, RISHI HAZRA, Shuangrui Ding, Sagar Vaze, Francois Porcher, Feng Li, Siyuan Li, Aishwarya Kamath, Ho Kei Cheng, Piotr Dollar, Nikhila Ravi, Kate Saenko, Pengchuan Zhang, and Christoph Feichtenhofer. Sam 3: Segment anything with concepts. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 138846–138923, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/e0982cbc81401df3430ee1ff780dc7a2-Paper-Conference.pdf.

Aditya Chinchure, Sahithya Ravi, Pushkar Shukla, Vered Shwartz, and Leonid Sigal. *Spotlight: Identifying and Localizing Video Generation Errors Using VLMs*, pp. 1–17. Springer Nature Switzerland, 2026. ISBN 9783032373212. doi: 10.1007/978-3-032-37321-2_1. URL http://dx.doi.org/10.1007/978-3-032-37321-2_1.

Jaemin Cho, Abhay Zala, and Mohit Bansal. Visual programming for step-by-step text-to-image generation and evaluation. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 6048–6069. Curran Associates, Inc., 2023. doi: 10.52202/075280-0265. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/13250eb13871b3c2c0a0667b54bad165-Paper-Conference.pdf.

Jaemin Cho, Yushi Hu, Jason Baldridge, Roopal Garg, Peter Anderson, Ranjay Krishna, Mohit Bansal, Jordi Pont-Tuset, and Su Wang. Davidsonian scene graph: Improving reliability in fine-grained evaluation for text-to-image generation. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 15625–15645, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/446905abfec9db8856f7f7465bbd5552-Paper-Conference.pdf.

Minkyu Choi, S P Sharan, Harsh Goel, Sahil Shah, and Sandeep Chinchali. We'll fix it in post: Improving text-to-video generation with neuro-symbolic feedback, 2026. URL https://arxiv.org/abs/2504.17180.

Rohan Choudhury, Koichiro Niinuma, Kris M. Kitani, and László A. Jeni. *Video Question Answering with Procedural Programs*, pp. 315–332. Springer Nature Switzerland, October 2024. ISBN 9783031729201. doi: 10.1007/978-3-031-72920-1_18. URL http://dx.doi.org/10.1007/978-3-031-72920-1_18.

Zijun Cui, Xiulong Liu, Hao Fang, Mingwei Xu, Jiageng Liu, Zexin Xu, Weiguo Pian, Shijian Deng, Feiyu Du, Chenming Ge, and Yapeng Tian. Do joint audio-video generation models understand physics?, 2026. URL https://arxiv.org/abs/2605.07061.

Yilun Du, Sherry Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Josh Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 9156–9172. Curran Associates, Inc., 2023. doi: 10.52202/075280-0403. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/1d5b9233ad716a43be5c0d3023cb82d0-Paper-Conference.pdf.

Hiroki Furuta, Heiga Zen, Dale Schuurmans, Aleksandra Faust, Yutaka Matsuo, Percy Liang, and Sherry Yang. Improving dynamic object interactions in text-to-video generation with ai feedback, 2026. URL https://arxiv.org/abs/2412.02617.

Jing Gu, Xian Liu, Yu Zeng, Ashwin Nagarajan, Fangrui Zhu, Daniel Hong, Yue Fan, Qianqi Yan, Kaiwen Zhou, Ming-Yu Liu, and Xin Wang. Phyworldbench: A comprehensive evaluation of physical realism in text-to-video models. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 75130–75164, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/79d86433c2acd12b6fa98553435d226e-Paper-Conference.pdf.

Tanmay Gupta and Aniruddha Kembhavi. Visual programming: Compositional visual reasoning without training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14953–14962, June 2023.

Jiarui Hai, Helin Wang, Weizhe Guo, and Mounya Elhilali. Flexsed: Towards open-vocabulary sound event detection. In *2025 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pp. 1–5. IEEE, October 2025. doi: 10.1109/waspaa66052.2025.11230975. URL http://dx.doi.org/10.1109/WASPAA66052.2025.11230975.

Xuan He, Dongfu Jiang, Ge Zhang, Max Ku, Achint Soni, Sherman Siu, Haonan Chen, Abhranil Chandra, Ziyan Jiang, Aaran Arulraj, Kai Wang, Quy Duc Do, Yuansheng Ni, Bohan Lyu, Yaswanth Narsupalli, Rongqi Fan, Zhiheng Lyu, Bill Yuchen Lin, and Wenhu Chen. VideoScore: Building automatic metrics to simulate fine-grained human feedback for video generation. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 2105–2123, Miami, Florida, USA,

November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.127. URL https://aclanthology.org/2024.emnlp-main.127/.

Xuan He, Dongfu Jiang, Ping Nie, Minghao Liu, Zhengxuan Jiang, Mingyi Su, Wentao Ma, Junru Lin, Chun Ye, Yi Lu, Keming Wu, Benjamin Schneider, Quy Duc Do, Zhuofeng Li, Yiming Jia, Yuxuan Zhang, Guo Cheng, Haozhe Wang, Wangchunshu Zhou, Qunshu Lin, Yuanxing Zhang, Ge Zhang, Wenhao Huang, and Wenhu Chen. Videoscore2: Think before you score in generative video evaluation, 2025. URL https://arxiv.org/abs/2509.22799.

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. CLIPScore: A reference-free evaluation metric for image captioning. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 7514–7528, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.595. URL https://aclanthology.org/2021.emnlp-main.595/.

Anthony Hu, Lloyd Russell, Hudson Yeo, Zak Murez, George Fedoseev, Alex Kendall, Jamie Shotton, and Gianluca Corrado. Gaia-1: A generative world model for autonomous driving, 2023a. URL https://arxiv.org/abs/2309.17080.

Yushi Hu, Benlin Liu, Jungo Kasai, Yizhong Wang, Mari Ostendorf, Ranjay Krishna, and Noah A. Smith. Tifa: Accurate and interpretable text-to-image faithfulness evaluation with question answering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 20406–20417, October 2023b.

Siyuan Huang, Liliang Chen, Pengfei Zhou, Shengcong Chen, Yue Liao, Zhengkai Jiang, Yue Hu, Peng Gao, Hongsheng Li, Maoqing Yao, and Guanghui Ren. Enerverse: Envisioning embodied future space for robotics manipulation. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 37693–37720. Curran Associates, Inc., 2025. doi: 10.52202/085713-1263. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/360052c2c6d0c8ec24c476d43236ab25-Paper-Conference.pdf.

Zeyu Huang, Adhiguna Kuncoro, Qixuan Feng, Jiajun Shen, Lucio Dery, Arthur Szlam, and Marc'Aurelio Ranzato. Context training with active information seeking, 2026. URL https://arxiv.org/abs/2605.13050.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. Vbench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21807–21818, June 2024.

Joel Jang, Seonghyeon Ye, Zongyu Lin, Jiannan Xiang, Johan Bjorck, Yu Fang, Fengyuan Hu, Spencer Huang, Kaushil Kundalia, Yen-Chen Lin, Loïc Magne, Ajay Mandlekar, Avnish Narayan, You Liang Tan, Guanzhi Wang, Jing Wang, Qi Wang, Yinzhen Xu, Xiaohui Zeng, Kaiyuan Zheng, Ruijie Zheng, Ming-Yu Liu, Luke Zettlemoyer, Dieter Fox, Jan Kautz, Scott Reed, Yuke Zhu, and Linxi Fan. Dreamgen: Unlocking generalization in robot learning through video world models. In Joseph Lim, Shuran Song, and Hae-Won Park (eds.), *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 5170–5194. PMLR, 27–30 Sep 2025. URL https://proceedings.mlr.press/v305/jang25a.html.

Sebastian Jung, Artem Zholus, Martin Sundermeyer, Carl Doersch, Ross Goroshin, David Joseph Tan, Sarath Chandar, Rudolph Triebel, and Federico Tombari. Tapnext++: What's next for tracking any point (tap)? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Findings*, pp. 8429–8438, June 2026.

Bingyi Kang, Yang Yue, Rui Lu, Zhijie Lin, Yang Zhao, Kaixin Wang, Gao Huang, and Jiashi Feng. How far is video generation from world model: A physical law perspective. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine*

Learning, volume 267 of *Proceedings of Machine Learning Research*, pp. 28991–29017. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/kang25g.html.

Sehoon Kim, Suhong Moon, Ryan Tabrizi, Nicholas Lee, Michael W. Mahoney, Kurt Keutzer, and Amir Gholami. An LLM compiler for parallel function calling. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 24370–24391. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/kim24y.html.

Christopher Latimer, Nicolò Boschi, Andrew Neeser, Chris Bartholomew, Gaurav Srivastava, Xuan Wang, and Naren Ramakrishnan. Hindsight: Structured agent memory that retains, recalls, and reflects. In Greg Durrett and Ping Jian (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pp. 275–285, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-392-0. doi: 10.18653/v1/2026.acl-demo.27. URL https://aclanthology.org/2026.acl-demo.27/.

Daeun Lee, Jaehong Yoon, Jaemin Cho, and Mohit Bansal. Self-correcting text-to-video generation with misalignment detection and localized refinement. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 36464–36489, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.1817. URL https://aclanthology.org/2026.findings-acl.1817/.

Zhiqiu Lin, Deepak Pathak, Baiqi Li, Jiayao Li, Xide Xia, Graham Neubig, Pengchuan Zhang, and Deva Ramanan. Evaluating text-to-visual generation with image-to-text generation. In Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol (eds.), *Computer Vision – ECCV 2024*, pp. 366–384, Cham, 2025. Springer Nature Switzerland. ISBN 978-3-031-72673-6.

Fangfu Liu, Hanyang Wang, Yimo Cai, Kaiyan Zhang, Xiaohang Zhan, and Yueqi Duan. Video-t1: Test-time scaling for video generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 18671–18681, October 2025.

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 (CVPR)*, pp. 22139–22149, June 2024.

Fanqing Meng, Jiaqi Liao, Xinyu Tan, Quanfeng Lu, Wenqi Shao, Kaipeng Zhang, Yu Cheng, Dianqi Li, and Ping Luo. Towards world simulator: Crafting physical commonsense-based benchmark for video generation. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 43781–43806. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/meng25c.html.

Juhong Min, Shyamal Buch, Arsha Nagrani, Minsu Cho, and Cordelia Schmid. Morevqa: Exploring modular reasoning models for video question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13235–13245, June 2024.

Saman Motamed, Laura Culp, Kevin Swersky, Priyank Jaini, and Robert Geirhos. Do generative video models understand physical principles? In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 948–958, March 2026.

NVIDIA, :, Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, Daniel Dworakowski, Jiaojiao Fan, Michele Fenzi, Francesco Ferroni, Sanja Fidler, Dieter Fox, Songwei Ge, Yunhao Ge, Jinwei Gu, Siddharth Gururani, Ethan He, Jiahui Huang, Jacob Huffman, Pooya Jannaty, Jingyi Jin, Seung Wook Kim, Gergely Klár, Grace Lam, Shiyi Lan, Laura Leal-Taixe, Anqi Li, Zhaoshuo Li, Chen-Hsuan Lin, Tsung-Yi Lin, Huan Ling, Ming-Yu Liu, Xian Liu, Alice Luo, Qianli Ma,

Hanzi Mao, Kaichun Mo, Arsalan Mousavian, Seungjun Nah, Sriharsha Niverty, David Page, Despoina Paschalidou, Zeeshan Patel, Lindsey Pavao, Morteza Ramezanali, Fitsum Reda, Xiaowei Ren, Vasanth Rao Naik Sabavat, Ed Schmerling, Stella Shi, Bartosz Stefaniak, Shitao Tang, Lyne Tchapmi, Przemek Tredak, Wei-Cheng Tseng, Jibin Varghese, Hao Wang, Haoxiang Wang, Heng Wang, Ting-Chun Wang, Fangyin Wei, Xinyue Wei, Jay Zhangjie Wu, Jiashu Xu, Wei Yang, Lin Yen-Chen, Xiaohui Zeng, Yu Zeng, Jing Zhang, Qinsheng Zhang, Yuxuan Zhang, Qingqing Zhao, and Artur Zolkowski. Cosmos world foundation model platform for physical ai, 2025. URL https://arxiv.org/abs/2501.03575.

Siru Ouyang, Jun Yan, I-Hung Hsu, Yanfei Chen, Ke Jiang, Zifeng Wang, Rujun Han, Long Le, Samira Daruki, Xiangru Tang, Vishy Tirumalashetty, George Lee, Mahsan Rofouei, Hangfei Lin, Jiawei Han, Chen-Yu Lee, and Tomas Pfister. Reasoningbank: Scaling agent self-evolving with reasoning memory. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 94327–94354, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/980ea04d23d1f6908964eba2a74afe45-Paper-Conference.pdf.

Prasan Roy, S. Seshadri, S. Sudarshan, and Siddhesh Bhobe. Efficient and extensible algorithms for multi query optimization. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, SIGMOD/PODS00, pp. 249–260. ACM, May 2000. doi: 10.1145/342009.335419. URL http://dx.doi.org/10.1145/342009.335419.

Timos K. Sellis. Multiple-query optimization. *ACM Transactions on Database Systems*, 13(1): 23–52, March 1988. ISSN 1557-4644. doi: 10.1145/42201.42203. URL http://dx.doi.org/10.1145/42201.42203.

Yu Shang, Xin Zhang, Yinzhou Tang, Lei Jin, Chen Gao, Wei Wu, and Yong Li. Roboscape: Physics-informed embodied world model. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 63674–63698. Curran Associates, Inc., 2025. doi: 10.52202/085713-2138. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/5c1ff00b27ba052039bb41531236baac-Paper-Conference.pdf.

S P Sharan, Minkyu Choi, Sahil Shah, Harsh Goel, Mohammad Omama, and Sandeep Chinchali. Neuro-symbolic evaluation of text-to-video models using formal verification. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pp. 8395–8405, June 2025.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: language agents with verbal reinforcement learning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 8634–8652. Curran Associates, Inc., 2023. doi: 10.52202/075280-0377. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/1b44b878bb782e6954cd888628510e90-Paper-Conference.pdf.

Kaiyue Sun, Kaiyi Huang, Xian Liu, Yue Wu, Zihan Xu, Zhenguo Li, and Xihui Liu. T2v-compbench: A comprehensive benchmark for compositional text-to-video generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pp. 8406–8416, June 2025.

Dídac Surís, Sachit Menon, and Carl Vondrick. Vipergpt: Visual inference via python execution for reasoning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11888–11898, October 2023.

Shuai Wang, Yaxin Feng, Xuekun Jiang, Shihan Tian, Ningyu Yan, Xing Shen, Chaoyang Lyu, Hui Wang, Yunsong Zhou, Hanqing Wang, Jiangmiao Pang, Yang Xiang, Xing Gao, Chunhua Shen, and Weinan Zhang. Gauge: A measurement-grounded benchmark for physical fidelity in simulation engines and video world models, 2026. URL https://arxiv.org/abs/2608.05948.

Youpeng Wen, Junfan Lin, Yi Zhu, Jianhua Han, Hang Xu, Shen Zhao, and Xiaodan Liang. Vidman: Exploiting implicit dynamics from video diffusion model for effective robot manipulation. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 41051–41075. Curran Associates, Inc., 2024. doi: 10.52202/079017-1298. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/481c70828a4ff20d31a646cc6cc95f3d-Paper-Conference.pdf.

Thaddäus Wiedemer, Yuxuan Li, Paul Vicol, Shixiang Shane Gu, Nick Matarese, Kevin Swersky, Been Kim, Priyank Jaini, and Robert Geirhos. Video models are zero-shot learners and reasoners, 2025. URL https://arxiv.org/abs/2509.20328.

Binfeng Xu, Zhiyuan Peng, Bowen Lei, Subhabrata Mukherjee, Yuchen Liu, and Dongkuan Xu. Rewoo: Decoupling reasoning from observations for efficient augmented language models, 2023. URL https://arxiv.org/abs/2305.18323.

Qiyao Xue, Xiangyu Yin, Boyuan Yang, and Wei Gao. Phyt2v: Llm-guided iterative self-refinement for physics-grounded text-to-video generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pp. 18826–18836, June 2025.

Sherry Yang, Yilun Du, Seyed Ghasemipour, Jonathan Tompson, Leslie Kaelbling, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 45210–45234, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/c4d66eae503694424123b93ac0fbaf17-Paper-Conference.pdf.

Yuhang Yang, Ke Fan, Shangkun Sun, Hongxiang Li, Ailing Zeng, FeiLin Han, Wei Zhai, Wei Liu, Yang Cao, and Zheng-Jun Zha. Videogen-eval: Agent-based system for video generation evaluation, 2025. URL https://arxiv.org/abs/2503.23452.

Qin Zhang, Peiyu Jing, Hong-Xing Yu, Fangqiang Ding, Fan Nie, Weimin Wang, Yilun Du, James Zou, Jiajun Wu, and Bing Shuai. Physion-eval: Evaluating physical realism in generated video via human reasoning, 2026. URL https://arxiv.org/abs/2603.19607.

Artem Zholus, Carl Doersch, Yi Yang, Skanda Koppula, Viorica Patraucean, Xu Owen He, Ignacio Rocco, Mehdi S. M. Sajjadi, Sarath Chandar, and Ross Goroshin. Tapnext: Tracking any point (tap) as next token prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9693–9703, October 2025.

A Experimental Details

Annotated corpus. Our annotated corpus contains 1,500 generated clips with 2,582 flaw records. For each flaw, one annotator recorded the violated prompt fragment, a rationale, a severity and a confidence score, and optionally a time span and boxes on the objects involved. A language model also assigns each flaw a type tag, such as actions and events or on-screen text. Every clip has at least one flaw, so the corpus measures recall and cannot measure precision. The development core is a stratified sample of 150 clips. One clip ended in an infrastructure failure, which leaves 149 clips with 304 flaw entries, 286 of them distinct. We distill nested sets of 8, 13, 19 and 33 lessons from the first 25, 50, 75 and 117 experience clips. The held-out split shares no clip identifier and no normalized prompt with the experience clips. Section E describes the corpus and the sampling in full.

External data. The VideoPhy-2 sample holds 200 test clips of VideoPhy-2 (Bansal et al., 2026), drawn with a fixed seed, and 55 of them belong to its hard subset. The public release of Physion-Eval (Zhang et al., 2026) holds 9,569 clips from five video generators, each with a human report of physical glitches, and 8,745 clips report at least one glitch. We used no Physion-Eval clip to develop VeriPhy. We run VeriPhy on the first 601 clips of a fixed randomized stream, 543 of them with a reported glitch. The lesson memory stays active during this run, VeriPhy updates it as the stream progresses, and no component sees a label before it scores the clip. The rewrite set holds 600 clips that Cosmos3-Nano generated from VideoPhy-2 test prompts.

Flaw matching. The findings of VeriPhy are its contradicted claims, and the findings of a baseline are the failures it reports. A text-only language model at temperature zero labels each pair of a flaw d and a finding f from the same clip as a whole match, a partial match or no match. A partial match captures one part of a compound flaw. Let B be the bipartite graph between the flaws $\mathcal{D}$ and the findings F of a clip. Its edges are the whole and partial pairs, and $\nu(B)$ is the size of its maximum matching. Summed over clips,

$$\text{Found} = \nu(B), \qquad \text{Recall} = \frac{\nu(B)}{|\mathcal{D}|}, \qquad \text{MatchRate} = \frac{\nu(B)}{|F|}. \tag{5}$$

The matching uses each flaw and each finding at most once. The run of Section 4.5 uses a simpler scorer, which credits each flaw with the first finding labeled a match and gives no partial credit.

Clip-level metrics. On the VideoPhy-2 sample, a fixed rule maps the claim states and the decision of VeriPhy to the 1–5 scale. We report exact agreement with the human rating, agreement within one point, and the Pearson (PLCC) and Spearman (SRCC) correlations, and we score both evaluators with the released evaluation script of VideoScore2 (He et al., 2025). On Physion-Eval, the evidence-conditioned flag rate is the share of clips that the backbone calls physically unrealistic when it reads only the check records of VeriPhy. These records hold each question, verdict, reason and measured value, and they contain no frame and no glitch report. The matched-report rate is the share of clips with a reported glitch for which a finding of VeriPhy matches the report under Equation (5). The anomaly-phrased rate is the share of clips with a finding that explicitly alleges a physical anomaly. The statistic J of Physion-Eval needs real reference videos that the public release withholds, so we report flag rates. The rates of the other critics and of untrained humans are those that Physion-Eval reports on its full evaluation set. On the rewrite set, the VideoPhy-2 AutoRater (Bansal et al., 2026) rates semantic adherence (SA) and physical commonsense (PC) from 1 to 5.

Baselines. The single-pass baseline gives the backbone of VeriPhy the prompt and all decoded frames in one call and asks for a list of flaws. The TIFA-style baseline (Hu et al., 2023b) asks the same backbone one yes/no question for each claim of VeriPhy over densely sampled frames. The DSG-style baseline (Cho et al., 2024) answers generated questions with dependency gating. We reimplement both decomposition baselines. The per-claim variant of Section 4.5 verifies one claim at a time.

Implementation. One Qwen3-VL-30B-A3B-Instruct checkpoint (Bai et al., 2025), served in BF16 on two GPUs, runs the planner, the semantic checks and event timing. SAM 3 (Carion et al., 2026) provides grounding, masks, tracks and counts, and Table 5 lists the other tools. We report cost as model and tool calls per clip, excluding the model that labels matches. All lessons load on every clip. Selecting lessons for each claim would add one model call per lesson and claim, which is 132

Table 3: Run ledger: every run whose numbers appear in the paper or appendix. Calls are mean model and tool calls per clip, excluding matching calls. The R2 cells list the single pass, the TIFA-style baseline, and the DSG-style variant in that order. Every run of VeriPhy uses the model revisions of Table 5. –: not recorded.

Run	Use (Sec.)	Population	Review	Scored by	Calls	Result used
R1	VeriPhy recall (4.1, F)	development core: 149 clips, 304 flaw entries	GPT-5.6 Sol	whole/partial bipartite judge	14.1	228 found (191 whole, 37 partial); 83 of the 228 from the review stage
R2	baselines (4.1, F)	as R1	none	as R1	1.0; 21.9; 52.1	222; 164 (125 whole); 161 found
R3	measurement cases (4.1, H)	development core; earlier routing pass	–	case selection rule	n/a	6 cases shown; 10 candidates excluded
R4	planner validity (C)	22 annotated clips, 233 claims; no video or tools	n/a	compiler validity	1 to 2	233 of 233 claims valid; 27 carry a specialist action
R5	miss attribution (4.5, G)	development core: 149 clips, 286 distinct flaws	none	binary first-match	13.0; 12.1	197 of 286 found (per-claim variant 179); 89 misses attributed
R7	clip-level agreement (4.1, I)	200 VideoPhy-2 test clips, 55 in its hard subset	–	VideoPhy-2 semantic-adherence ratings	–	SRCC 0.465 for VeriPhy against 0.327 for VideoScore2
R8	Physion-Eval flag rates (4.2)	first 601 clips of a fixed randomized Physion-Eval stream, 543 with a glitch report	–	flag rate from check records; matching judge	–	26.8% flagged; 18.4% of 543 matched; 15.5% anomaly-phrased
R9	lessons (4.3, D)	held-out split: 306 clips, 502 flaws	–	whole/partial judge	–	340 found with no lessons; 375 with 33 lessons
R10	prompt-rewrite loop (4.4, J)	600 Cosmos3-Nano clips from VideoPhy-2 test prompts; 589 paired	–	VeriPhy verdicts; AutoRater	–	49 of 65 implausible verdicts flip; joint agreement 22.8% (600 clips) to 23.1% (589)

calls for a prompt with four claims, while loading all 33 lessons adds 9 kB to a planner prompt of 17.8 kB.

Statistics. Each configuration runs once. When we score two configurations on the same flaws, n_+ counts the flaws that only the second finds. Likewise, n_- counts the flaws that only the first finds. We report an exact sign test and the change in recall,

$$p = \min\Big\{1,\ 2\Pr\big[\mathrm{Bin}(n_+ + n_-, \tfrac{1}{2}) \geq \max(n_+, n_-)\big]\Big\}, \qquad \hat{\Delta} = \frac{n_+ - n_-}{|\mathcal{D}|}. \tag{6}$$

The 95% interval for $\hat{\Delta}$ comes from a Wilson interval for $n_+/(n_+ + n_-)$. These tests treat flaws as independent. Differences in correlation, and the difference between VeriPhy and the per-claim variant, use a paired bootstrap.

B Run ledger and populations

Table 3 lists every run that contributes a number to the paper or appendix. For each run it records the population, the review stage, the scoring protocol, the cost, and the result we use. These fields show which results compare directly.

Comparisons across runs. R1 and the three R2 baselines share the 149 clips, the 304 flaw entries, and the whole/partial matching judge, so their counts compare directly. R5 covers the same 149

Table 4: The nine plan operations. A plan line uses one operation or a conjunction of them. The last column names the evidence that settles each question (Section 2.3).

Operation	Question	Evidence
`judge(a)`	Does assertion a hold in the video?	semantic check by a vision-language model over dense frames
`exists(x)`	Is object x visible at some time?	check by the vision-language model over dense frames
`eq(count(x), N)`	Are there exactly N of x?	identities tracked by SAM 3
`before(A, B)`, `during(A, B)`	Does event A come before event B, or within it?	event time windows and an interval rule
`rel(x, y, r)`	Does spatial relation r hold between x and y?	object masks, plus depth for *behind* and *in front of*
`traj(x, r, y)`	Does x move relative to y as motion r states?	object tracks with camera motion removed
`text(s)`	Does the video render the string s?	text recognition and exact string match
`sound(e)`	Does the audio contain event e?	sound-event detection

clips without the review stage and with the first-match scorer, and it counts 286 distinct flaws. R9 runs on the held-out split, a different population from the development core.

Held-out split. We build the held-out split of R9 from 363 clips with 626 flaws. We remove every clip that shares a clip identifier or an exact normalized prompt with one of the 117 experience clips. It holds 306 clips with 502 flaws. The experience clips and the held-out split both come from the annotated corpus of Section E.

C VeriPhy specification

This appendix completes the definitions of Section 2. Table 4 lists the plan vocabulary, Figure 5 shows a compiled plan, Table 5 lists the model behind each backend, and Algorithm 1 collects the steps into one procedure.

C.1 Plan vocabulary and validity

The planner maps a prompt p to claims $c_i = (s_i, \phi_i)$, $i = 1, \ldots, N_p$, where each span s_i occurs verbatim in p, and to one plan line π_i per claim (Equation (1)). It works in two passes. A naming pass over the whole prompt fixes a stable identifier for every distinct entity and every separately asserted event occurrence. The planner then plans claims in groups of at most four, each against the full prompt and the shared names, and renames each group's local identifiers into the shared names before it concatenates the groups. Renaming on assembly prevents two groups from referring to one entity by different names, or to two occurrences by one name.

For claim c_i the planner writes one line `check(ci, <expr>)`, whose expression is one operation of Table 4 or a conjunction of operations. The planner has no video at this stage, so the line states what evidence would settle the claim and says nothing about whether the claim holds. The terms `count` and `window` are structural. The surface name `judge` denotes a semantic check and is unrelated to the matching judge of Section E.

Graph and validity. The compiler expands the surface plan into the typed graph $G = (\mathcal{A}, \mathcal{Q}, \mathcal{E}, b)$ of Section 2.2. The action set $\mathcal{A}$ holds semantic passes of the vision-language model and specialist calls, and each action $a_j = (o_j, \alpha_j, g_j, r_j)$ records its operation o_j, typed arguments α_j, scope rule g_j, and result type r_j. The check set $\mathcal{Q}$ holds typed checks, each edge in $\mathcal{E}$ points from a prerequisite to its consumer, and $b(q) = i$ binds check q to claim c_i. A program is valid when

$$\mathrm{Valid}(G) = \mathrm{Parse}(G) \wedge \mathrm{Refs}(G) \wedge \mathrm{Typed}(G) \wedge \mathrm{Acyclic}(G) \wedge \big[\forall i\ \exists q \in \mathcal{Q} : b(q) = i\big]. \quad (7)$$

Here Parse checks that the surface program parses; Refs checks identifiers and dependency closure; Typed checks operator arity, the fixed relation vocabularies, and that every check receives the result types it requires; and $\mathrm{Acyclic}$ checks the dependency graph for cycles. For example, a directed trajectory must carry a target, and a depth-ordering check must consume both masks and depth. The

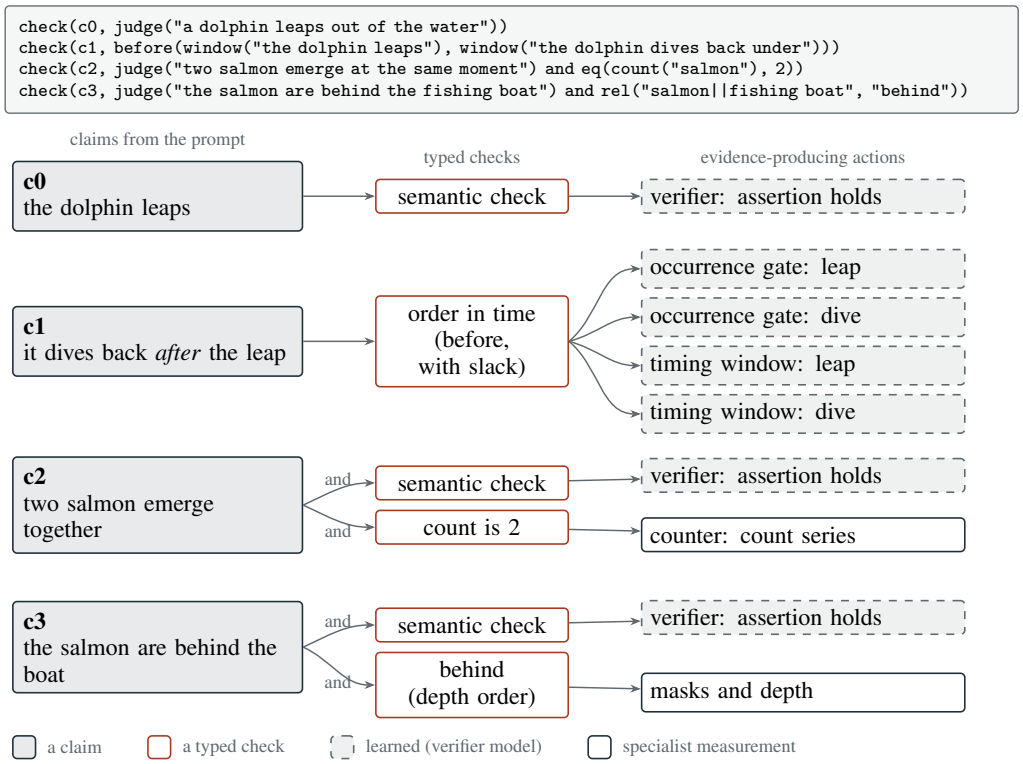


Figure 5: An illustrative plan for a four-claim prompt, compiled into claims, typed checks, and the evidence-producing actions each check consumes. A numeral adds a count check beside a semantic check (c2). An order check consumes two occurrence gates and two timing windows, and its predicate reads the windows only after both gates return supported (c1). A depth relation consumes masks and depth (c3). Dashed boxes are learned outputs of the vision-language model; solid white boxes are specialist measurements.

vocabulary has no *on top of* relation and rejects negated measurements, so such claims go to plan fallback.

Plan fallback. The compiler validates the assembled plan as a whole, because its binding constraints cross claims, checks, and actions. For a failing group, VeriPhy re-asks the planner with that group's parser or type complaints and a restatement of the format, but with no suggested answer. After a fixed number of attempts, plan fallback applies: an invalid line becomes one semantic verification of its claim, and an invalid term becomes a semantic sub-check. The trace records each substitution and charges it as a model call. A failed naming pass sends every claim to plan fallback. Plan fallback is distinct from the review stage defined below.

Planner validity diagnostic (R4). Before any end-to-end run, and with no video or tool access, the planner produced a valid plan for all 233 claims of 22 annotated clips at one or two calls per clip, and 27 of the 233 claims (11.6%) carried a specialist action that survived validation. On diagnostic sets that require a quantitative measurement, the planner routed 19 of 30 counting, spatial, and ordering claims and 10 of 10 synthetic counting prompts to a specialist.

A compiled plan. Figure 5 compiles a four-claim prompt. The numeral in c2 adds a count check beside its semantic check, the order claim c1 reads two timing windows only after both occurrence gates pass, and the depth relation c3 consumes masks and depth, so each claim draws on the evidence its wording calls for.

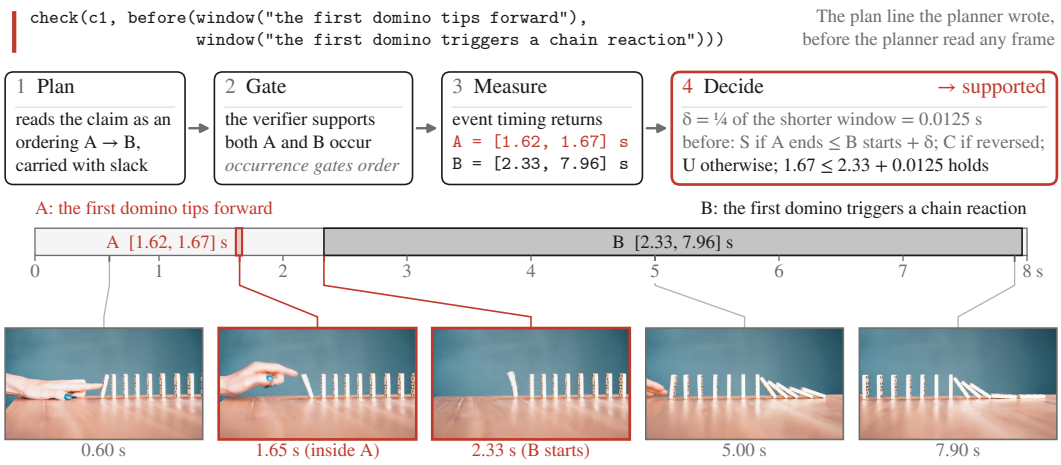


Figure 6: Recorded trace for an ordering claim from a domino-chain prompt. Occurrence checks support both prerequisite events, and the vision-language model localizes them to $W_A = [1.62, 1.67]$ s and $W_B = [2.33, 7.96]$ s. A deterministic temporal predicate with slack 0.0125 s then resolves *before* as supported.

C.2 Scoped Execution

A clip is $\mathcal{V} = \{(I_t, \tau_t)\}_{t \in \mathcal{T}}$ with frame indices $\mathcal{T} = \{0, \dots, T-1\}$ and ordered timestamps $\tau_0 < \cdots < \tau_{T-1}$; separating index from time allows nonuniform decode timestamps. The executor visits the actions of G in topological order. The compiled scope rule g_j of action a_j maps its prerequisite records to the full index set or to a declared subset, and the frozen backend F_{o_j} runs on the scoped clip (Equation (2)). The scope rule is part of the validated plan. The executor pads a usable measured window, enforces a minimum duration, and converts the window to the frame indices whose timestamps fall inside it. A missing, invalid, or too broad window falls back to the whole clip, so a failure to localize does not count as evidence that the event or relation is absent. Two actions share one execution only when their realized signatures match: clip identity, operation, typed arguments, references, and realized scope. The executor runs each such equivalence class once and charges it once, the exact-match special case of common-subexpression sharing in multiple-query optimization (Sellis, 1988; Roy et al., 2000). Because the realized scope belongs to the signature, a whole-clip request and a localized request do not merge on similar query text alone.

C.3 Temporal Predicates

A timing relation $R \in \{\text{before}, \text{during}\}$ between events A and B compares their measured windows $W_A = [t_A^-, t_A^+]$ and $W_B = [t_B^-, t_B^+]$ through an order predicate $\rho_R(W_A, W_B)$. The occurrence gate of Equation (10) applies the predicate only when both occurrence checks are S; the relation is C if either occurrence is C, and U if either is U or a window is unavailable. Timing can therefore compose an order but cannot establish that an event occurred. The order predicate is

$$\rho_{\text{before}}(W_A, W_B) = \begin{cases} \mathsf{S}, & t_A^+ \le t_B^- + \delta, \\ \mathsf{C}, & t_B^+ \le t_A^- + \delta, \\ \mathsf{U}, & \text{otherwise}, \end{cases} \qquad \delta = \tfrac{1}{4} \min\big(t_A^+ - t_A^-,\ t_B^+ - t_B^-\big). \tag{8}$$

For positive durations the quarter-window slack makes the first two cases mutually exclusive. Containment uses the same slack δ. These predicates relax Allen's *before* and *during* relations (Allen, 1983) by a tolerance that scales with the window lengths. In the trace of Figure 6, $W_A = [1.62, 1.67]$ s and $W_B = [2.33, 7.96]$ s give $\delta = 0.0125$ s, and $t_A^+ = 1.67$ s lies below $t_B^- + \delta$, so *before* resolves to S.

C.4 Evidence records and status

Each executed action writes a record bound to each claim it serves:

$$R_{j,i} = \big(\mathrm{id}(\mathcal{V}),\, p,\, s_i,\, o_j,\, \Omega_j,\, \sigma_j,\, \mu_j,\, h_j,\, \kappa_j\big). \tag{9}$$

The status σ_j is `measured`, `abstained`, or `errored`. The payload μ_j holds either a typed measurement or an explicitly tagged learned base state; h_j holds native-output and artifact hashes; and κ_j holds calls and accelerator time, with optional wall time, price, and token counts. Shared execution can bind one record to several claims, and the executor charges its cost once per execution identifier.

The three statuses are mutually exclusive. A `measured` record carries a payload and no abstention reason or error, an `abstained` record carries a reason and no payload, and an `errored` record carries an infrastructure message. A valid negative measurement, such as a complete usable track that lacks the requested motion, stays `measured` and can contradict a claim. A failure to track through blur, occlusion, or low quality, or a backend refusal, is `abstained`. VeriPhy retries a terminal `errored` record; if the error persists, the clip becomes $\perp_{\mathrm{infra}}$ and we exclude it from evaluation, which happened to one development-core clip (Section E).

Counting reduces a per-frame count series to its mode. When grounding fails on most frames, the mode can be a confident zero that looks like a true absence, so the record keeps auxiliary fields: a stability flag, the range of the series, and the frames read.

Specialist schemas reject judgement-shaped fields, nonfinite numbers, duplicate keys, and non-canonical payloads. Every record carries the full prompt and the verbatim span s_i, and the store addresses heavy artifacts by path and byte hash. These constraints keep a specialist from emitting a prompt-satisfaction label.

C.5 Specialists and operations

Table 5 lists the model behind each capability and the object it returns. One Qwen3-VL checkpoint covers planning, semantic checks, and event timing, while the measurement backends return masks, counts, tracks, depth order, transcripts, or sound intervals, which resolvers turn into check states.

Track-based motion operations. The motion operations share one grounding step: SAM 3 segments the named subject, and the operations measure over tracks of that subject. The implementation provides eleven operations: displacement, contact, shape change, path shape, permanence, hop counting, damping, onset ordering, clip duration, playback rate, and viewpoint motion. Displacement uses multiples of the subject's own radius as its unit, so it needs no scene calibration; its direction comes from the furthest excursion, since a rise followed by a fall cancels in the net travel; and dense whole-frame optical flow removes camera motion. Contact measures the closest approach between two tracked masks. Shape change measures surface motion relative to the background flow, which registers a scroll rolling in place with no net displacement. Path shape separates, for example, a circle from a figure of eight. Every operation returns a number with its evidence (frames read, tracker, and threshold) and leaves the decision to the resolver. It abstains when the subject covers less than 1% of the frame or the track is too short to measure. A grounding guard, a learned check that shows one frame with the candidate boxes and asks whether they hold the named subject, discards boxes that do not, and the check then resolves unknown. In R1, the most common reason a physical specialist abstains is that no candidate box passes the guard.

On-screen text. Text claims use a recognition-head OCR engine, PaddleOCR PP-OCRv5 with EasyOCR as fallback. A recognition head returns a malformed string as rendered, and a deterministic comparison with the prompt string rejects it, whereas a generative model asked what a sign says can return the well-formed string the prompt requested. Each record states the frames read. In the earlier routing pass of Section H, the engine read every tenth frame.

Sound. The sound operation calls FlexSED on the audio track and returns sound intervals. Flaw selection removed the 64 audio-only flaw records (Section E), so no flaw in the development core concerns sound alone.

Table 5: Backends of VeriPhy in run R1: the model behind each capability and the object it returns. Deterministic resolvers turn count, timing, track, and transcription outputs into check states. Semantic checks, event timing, and the depth-ordered spatial interpretation are learned, and their records carry a learned tag. The review stage acts after the roll-up. Released components use Apache-2.0, MIT, or the SAM licence, except the depth model, which is CC-BY-NC-4.0 and which we use for research only.

Capability	Model and use	Returned object
Planner	Qwen3-VL-30B-A3B-Instruct (Bai et al., 2025), BF16, 31 B parameters (3 B active, mixture of experts); prompt only	claims and a typed plan, before any frame
Semantic checks	same checkpoint, separate call on densely decoded, timestamped frames	tagged learned state with rationale
Event timing	same checkpoint, over timestamped frames	window, absence, or refusal
Review stage	GPT-5.6 Sol	change of a supported claim to contradicted; both states logged
Grounding, masks, counting	SAM 3 (Carion et al., 2026), about 0.85 B parameters	per-frame masks, tracked identities, count series
Tracks for motion operations	TAPNext++ (Jung et al., 2026), which extends TAPNext (Zholus et al., 2025), over the SAM 3 subject masks	tracks for the motion operations
Camera cancellation	dense whole-frame optical flow	subject-relative displacement
Depth ordering	depth model on localized evidence, CC-BY-NC-4.0	behind or in front
Rendered text	PaddleOCR PP-OCRv5, EasyOCR as fallback	literal glyph transcription
Sound events	FlexSED (Hai et al., 2025)	sound intervals

C.6 Resolvers and composition

Each record R_j carries a status σ_j and a payload μ_j (Equation (9)). A typed resolver ψ_q maps the payloads of the prerequisite actions $\mathrm{Pa}(q)$ of check q to its base state β_q (Equation (3)), which is U when a prerequisite abstained or carries no usable evidence. A prerequisite that ends `errored` after retry aborts the clip before any resolver runs (Algorithm 1, line 9). Resolvers are deterministic for count, track, and transcription checks, for the order predicate over learned timing windows, and for the other explicit predicates. They are learned for three semantic sub-checks: subtype identity, action or event occurrence, and depth-ordered spatial interpretation on localized evidence. A learned resolver uses an asymmetric gate: it returns C only when the model reports a visible contradicting feature, and unresolved evidence, unparseable output, or a missing modality yields U, so an instrument failure does not become a finding against the clip.

The check state z_q applies negation and the occurrence gate to the base state; a check with neither has $z_q = \beta_q$. Negation acts on the positive check state, and no backend measures a negative quantity: a negated check has $z_q = \neg_3 \beta_q$, where $\neg_3 \mathsf{S} = \mathsf{C}$, $\neg_3 \mathsf{C} = \mathsf{S}$, and $\neg_3 \mathsf{U} = \mathsf{U}$. An order check q between events A and B has the order predicate of Equation (8) as its base state and also reads the states z_A and z_B of the two occurrence checks:

$$z_q = \begin{cases} \mathsf{C}, & z_A = \mathsf{C} \text{ or } z_B = \mathsf{C}, \\ \mathsf{U}, & \text{otherwise, if } z_A = \mathsf{U} \text{ or } z_B = \mathsf{U}, \\ \beta_q, & \text{otherwise.} \end{cases} \tag{10}$$

We write $\tilde{y}_i$ and $\tilde{Y}$ for the claim states and the decision that the rule of Equation (4) gives before the review stage. A trace that ends in a terminal infrastructure error yields $\perp_{\mathrm{infra}}$ for the clip in place of a decision.

Algorithm 1 One-clip procedure of VeriPhy. Learned steps: planning, semantic checks and event timing, learned resolvers, and review. All other steps are deterministic. The review stage sends each claim in state S to GPT-5.6 Sol.

Require: clip $\mathcal{V} = \{(I_t, \tau_t)\}_{t \in \mathcal{T}}$, prompt p
Ensure: claim states $\{y_i\}$, clip decision Y or $\perp_{\text{infra}}$, trace of records

1: $\{(c_i, \pi_i)\}_{i=1}^{N_p} \leftarrow \textsc{Plan}(p)$ ▷ text only; learned
2: $G = (\mathcal{A}, \mathcal{Q}, \mathcal{E}, b) \leftarrow \textsc{Compile}(\{\pi_i\})$, re-asking failing groups ▷ before any frame
3: **if** $\neg\text{Valid}(G)$ after the last attempt **then** ▷ Equation (7)
4: $G \leftarrow \textsc{PlanFallback}(G)$; record and charge each substitution
5: **end if**
6: **for** $a_j \in \mathcal{A}$ in topological order of G **do** ▷ frames first read here
7: $\Omega_j \leftarrow g_j(\{R_k : (a_k, a_j) \in \mathcal{E}\}, \mathcal{T})$; $R_j \leftarrow F_{o_j}(\mathcal{V}|_{\Omega_j}, \alpha_j)$ ▷ Equation (2)
8: **if** R_j is `errored` after retry **then**
9: **return** $\perp_{\text{infra}}$ ▷ clip excluded
10: **end if**
11: **end for**
12: **for** $q \in \mathcal{Q}$ **do**
13: $z_q \leftarrow$ base state β_q, negated or gated where required ▷ Equations (3) and (10)
14: **end for**
15: **for** $i \leftarrow 1$ to N_p **do**
16: $y_i \leftarrow \tilde{y}_i$, the claim roll-up over $K_i = \{q : b(q) = i\}$ ▷ Equation (4)
17: **end for**
18: **for all** i with $y_i = \mathsf{S}$ **do** ▷ review stage; learned
19: $v_i \leftarrow \textsc{Review}(c_i)$; log $\tilde{y}_i$ and v_i
20: **if** $v_i = \mathsf{C}$ **then**
21: $y_i \leftarrow \mathsf{C}$
22: **end if**
23: **end for**
24: $Y \leftarrow$ clip roll-up over $\{y_i\}$ ▷ Equation (4)
25: **return** $\{y_i\}$, Y, and the trace

C.7 Review stage

After the roll-up, the review stage passes each claim resolved as S to GPT-5.6 Sol, which returns its own verdict on the claim. Write $v_i \in \{\mathsf{S}, \mathsf{C}\}$ for this verdict, with $v_i = \mathsf{S}$ for a claim that the review does not re-examine. The final claim state is $y_i = \mathsf{C}$ if $\tilde{y}_i = \mathsf{S}$ and $v_i = \mathsf{C}$, and $y_i = \tilde{y}_i$ otherwise, and the final decision Y applies the clip rule of Equation (4) to the final states y_i. The review stage does not re-examine specialist measurements, and the only change it can make is from S to C. The trace stores the pre-review and post-review states side by side, so a run can be re-scored on its pre-review states. On the development core, 83 of the 228 flaws that VeriPhy finds match findings from the review stage (Table 1).

C.8 Procedure

Algorithm 1 gives the one-clip procedure. Lines 1–5 plan and compile the checks before any frame is read, lines 6–11 execute the actions and can end the clip at line 9, lines 12–17 resolve the checks and roll them up into claim states, lines 18–23 apply the review stage, and line 24 forms the clip decision.

C.9 Proofs for the aggregation rules

We order states by information: $\mathsf{U} \sqsubseteq \mathsf{S}$ and $\mathsf{U} \sqsubseteq \mathsf{C}$, while S and C are incomparable. A state is decided if it is S or C, and a decided state is maximal in this order. A rule is monotone if refining its inputs in $\sqsubseteq$ refines its output in $\sqsubseteq$.

Order predicate. Suppose both decided cases of Equation (8) hold, so $t_A^+ \le t_B^- + \delta$ and $t_B^+ \le t_A^- + \delta$. Adding the two inequalities gives $(t_A^+ - t_A^-) + (t_B^+ - t_B^-) \le 2\delta$. Let $m = \min(t_A^+ - t_A^-,\, t_B^+ - t_B^-)$. The left side is at least $2m$ and the right side equals $m/2$, so $2m \le m/2$ and hence $m \le 0$. The two cases therefore never hold together when both windows have positive length. The same argument

holds for any tolerance factor below one, so the factor $\frac{1}{4}$ sets the size of the tolerance and is not needed for exclusivity.

Formal statement. In the notation of Section C.6, Proposition 1 reads as follows. Fix a valid program G and the review verdicts v_i. Suppose some base states β_q change from U to S or C, and no other base state changes. Then (i) every check state z_q and claim state $\tilde{y}_i$ or y_i that was S or C keeps its value; (ii) every decision $\tilde{Y}$ or Y that was plausible or implausible keeps its value; and (iii) in any case, $\tilde{Y} =$ plausible if and only if $N_p \geq 1$ and $z_q =$ S for every check q of G.

Proof of Proposition 1. We first show that every rule of Section C.6 and the review rule are monotone. Negation maps U to U and swaps S and C, so it is monotone. For the gate of Equation (10), let the inputs (z_A, z_B, β_q) refine to (z'_A, z'_B, β'_q). If $z_q =$ C, then some z_A or z_B equals C, the refined input keeps C, and $z'_q =$ C. If $z_q = \beta_q$, then $z_A = z_B =$ S, so $z'_A = z'_B =$ S and $z'_q = \beta'_q \sqsupseteq \beta_q$. If $z_q =$ U, there is nothing to show. For the claim rule of Equation (4), if $\tilde{y}_i =$ C, some check of K_i is C and stays C, so $\tilde{y}'_i =$ C. If $\tilde{y}_i =$ S, every check of K_i is S and stays S, so $\tilde{y}'_i =$ S. The same argument applies to the decision rule, since the program fixes N_p. For the review rule with fixed verdicts v_i, a decided y_i is either C from $\tilde{y}_i =$ S and $v_i =$ C, or equal to a decided $\tilde{y}_i$, and both cases persist when $\tilde{y}_i$ is refined.

Now let the base states refine from β to β'. The program is acyclic, so we can visit the checks in topological order. By induction along this order, every check state refines, because each check state applies monotone rules to its base state and to the states of earlier checks. Claim states and decisions then refine by the monotonicity of the claim, decision and review rules. A decided state is maximal, so a refinement leaves it unchanged, which proves (i) and (ii). For (iii), $\tilde{Y} =$ plausible holds exactly when $N_p \geq 1$ and every claim is S. A claim is S exactly when all of its checks are S, every claim has at least one check by Equation (7), and every check belongs to one claim. Hence $\tilde{Y} =$ plausible exactly when $N_p \geq 1$ and every check of G is S. □

D LESSONS

Distillation and curation. The lesson set is distilled from the flaws that VeriPhy misses on 117 experience clips. For every annotated flaw missed on an experience clip, a language model reads the corresponding trace together with the flaw record and proposes a lesson that links a recurring claim property to a verification strategy; Table 18 identifies the prompt, whose full text is provided in the supplement. Curation then groups these candidates into 33 families of equivalent advice and keeps one lesson per family. The resulting lessons are appended to the planner prompt; no model weights are updated. Distillation is performed once offline, and each experimental arm uses a fixed lesson file.

The 33 lessons. Each lesson names a claim property and the check it calls for. For example, one lesson turns a singular-entity or exact-quantity requirement into a separate cardinality check verified across the beginning, middle, and end of the clip. The 8 lessons from the first 25 experience clips all concern actions. The supplement contains the full lesson file.

Dose design. The five arms are nested: 0, 8, 13, 19, and 33 lessons, distilled from the first 0, 25, 50, 75, and 117 experience clips, with each lesson set a prefix of the next. The untaught arm loads an empty lesson file through the same path. We compare all four taught arms with one untaught run, and every arm scores the same 502 flaws, so comparisons are paired by flaw.

Gate diagnostic. We dropped an acceptance gate that admits a candidate lesson only if it improves a validation score. On 80 earlier candidates over a 10-clip validation set, the gate admitted none: 39 failed its minimum-evidence criterion and 41 fired on no validation clip. The intervention we evaluate is therefore the curated set without admission control.

Statistics. Every test and interval in this appendix is flaw-level and treats the 502 flaws as independent. For 33 lessons against untaught, an exact sign test on the 62 gained and 27 lost flaws gives $p = 3 \times 10^{-4}$, and a Wilson interval on the switched flaws gives $+3$ to $+10$ points. Each taught arm lies 34 to 37 flaws above untaught (sign tests, $p < 10^{-3}$), and no two taught arms differ by more than 3 (pairwise $p \geq 0.8$).

Table 6: Lesson effect by machine-assigned type tag on the 502 flaws of the held-out split (R9; more found is better). Middle columns count flaws found with ℓ lessons. Change is the untaught-to-33-lesson difference in flaws, with percentage points of that tag's flaws in brackets. Intervals are flaw-level 95% intervals in points, computed from the flaws whose found status switched; bold rows exclude zero. Families counts lessons whose family name matches the tag by keyword, and names without a match default to *action*, so the action count includes every unmatched name. The 5 flaws tagged *other* appear only in the All row.

		Found with ℓ lessons							
Type tag	Flaws	$\ell = 0$	8	13	19	33	Change	95% interval	Families
action or event occurs	**211**	**146**	**169**	**169**	**166**	**163**	**+17 (+8)**	**[+2, +12]**	**20**
on-screen text	**73**	**53**	**63**	**56**	**62**	**61**	**+8 (+11)**	**[+2, +15]**	**1**
attribute	46	29	30	32	31	33	+4 (+9)	[−2, +12]	2
order and timing	22	11	13	13	13	14	+3 (+14)	[−9, +27]	3
how many	19	9	9	11	11	12	+3 (+16)	[−10, +31]	2
object identity	52	39	41	39	42	41	+2 (+4)	[−3, +7]	4
camera and style	14	11	11	12	11	12	+1 (+7)	[−4, +7]	0
spatial relation	45	28	29	30	28	28	0 (0)	[−10, +10]	1
physical motion	15	9	8	12	9	9	0 (0)	[−11, +11]	0
All flaws	**502**	**340**	**375**	**377**	**374**	**375**	**+35 (+7)**	**[+3, +10]**	**33**

By type tag and severity. Two tags have flaw-level intervals that exclude zero (Table 6): *action or event occurs* gains 17 of 211 flaws and *on-screen text* gains 8 of 73. By annotator severity, found flaws rise by 19 of the 288 severity-3 flaws (flaw-level sign test, $p = 0.01$) and by 13 of the 163 severity-4 flaws ($p = 0.03$); the remaining 51 flaws carry grades 1, 2, and 5.

Timing of the text gain. The *on-screen text* tag finds 53 of 73 flaws untaught, 63 with 8 lessons, 56 with 13, 62 with 19, and 61 with 33. The 8-lesson gain of 10 flaws is the largest percentage-point gain of any tag at that dose, and none of the first 8 lessons names text. The added text checks go to clips the planner already checked: text checks rise from 81 untaught to 114 with 8 lessons and 136 with 33, yet among the 56 clips that carry a text flaw, 41 receive any text check untaught and 40 with 33 lessons. This pattern fits a general decomposition instruction better than content aimed at text.

Plan effects. Claims per clip rise across all five arms: 9.5, 15.1, 16.0, 16.1, and 16.7. Over the 306 clips, claims rise from 2,905 untaught to 5,106 with 33 lessons, while checks per claim move from 1.13 to 1.18.

E CORPUS, DEVELOPMENT CORE, AND MATCHING PROTOCOL

Corpus. The corpus contains 1,500 generated clips and 2,582 annotated prompt departures, which we call flaws. A single annotator evaluated each clip against its prompt. The source generator of each clip is not recorded. Every clip carries at least one annotated flaw, so the corpus measures recall and not precision. The ethics statement describes the provenance, rights, and annotation of the corpus.

Derivation. Selection narrows the corpus in three recorded steps (Figure 7). Clip selection sets aside 894 clips: 713 whose prompts describe no motion and whose flagged moments are not actions, and 181 whose video was not retrievable. It leaves 606 clips with 1,171 flaws. Flaw selection then removes the 64 flaw records whose complaint concerns sound only and drops no clip, which leaves 606 clips with 1,107 flaws. A deterministic seeded draw, stratified by severity, flaw count, and duration, selects the frozen core of 150 clips and 306 flaws, and a continuity stratum of 29 clips (85 flaws) is exempt from replacement. A further 711 annotated clips (1,130 flaws), disjoint from the 606 selected clips, form a learning pool.

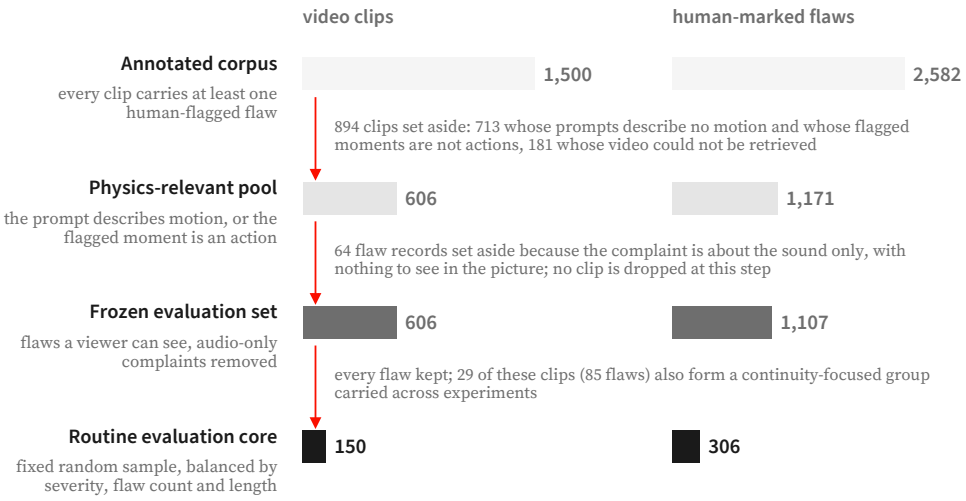


Figure 7: Derivation of the development core from the corpus (clips left, flaws right; bar lengths to scale within each column). Only the first step removes clips; the second removes audio-only flaw records. In the graphic, "physics-relevant pool" is the set left by clip selection, "frozen evaluation set" is the set left by flaw selection, and "routine evaluation core" is the development core.

Table 7: Coverage of the six flaw-record fields on the frozen core (150 clips, 306 flaws). The first four fields appear in every record; the annotator supplied a time span for 237 flaws and a box track for 208.

Field	Note	Coverage
violated fragment	median 8 words	306 of 306
rationale	median 25 words	306 of 306
severity	1 to 5, centred at 3 to 4	306 of 306
confidence	1 to 5; 239 marked 4 or 5	306 of 306
time span	optional	237 of 306
box track	optional	208 of 306

Flaw records. Each flaw record has four fields present in every record (the violated prompt fragment, a free-text rationale, a severity from 1 to 5, and the annotator's confidence from 1 to 5) and two optional fields (a time span and a box track). Table 7 gives their coverage on the frozen core.

Example records. Table 8 reproduces six flaw records verbatim. We first drew ten clips with a fixed seed from the corpus clips whose video is available: one clip for each clip-level severity from 1 to 5, and five more for record features (object relations, a record without boxes, five or more records, and two frame rates). From these ten clips we chose six records that differ in flaw type and in the optional fields, so the table illustrates the format and is not a random sample of flaws. The violated fragment is always a verbatim substring of the prompt, just as the span of a claim is (Section 2.1), and the rationale states what the video shows instead. The records also show three properties of the corpus described above: the text record carries neither a time span nor boxes, the audio record is of the kind that flaw selection removes, and one of the ten sampled clips lists the same fragment and rationale twice under two identifiers, the kind of duplicate that reduces the 304 core entries to 286 distinct flaws.

Core statistics. Core clips run a median of 8.0 s (range 5.0 to 10.3 s), 118 of the 150 at 24 frames per second, under prompts of median length 42 words (longest 188). Sixty-six clips carry flaws of more than one type. Clips carry a median of 2 flaws (mean 2.04): 63 clips carry one flaw, 47 two, 23 three, 11 four, 4 five, and 2 clips more (Figure 8). Of the 306 flaws, 237 carry a time span; 57

Table 8: Six flaw records from the corpus, reproduced verbatim with unedited rationales. Sev. and Conf.: severity and annotator confidence, both from 1 to 5. Span: annotated time span in seconds, from the frame indices and the clip's frame rate. Box: whether the record carries a box track. –: not given. The last record concerns sound only, and flaw selection removes such records.

Violated fragment	Annotator rationale	Sev.	Conf.	Span (s)	Box
"A dog walks up and takes a bite from the bowl, then walks away"	The dog in the video bites the food from the bowl multiple times, which does not match the prompt's requirement of taking a single bite.	1	4	1.67–3.92	yes
"a fox's ears appear among tall grass"	Two foxes appear in the video, which contradicts the prompt's request for only one fox.	3	4	1.17–2.50	yes
"with the text 'Morning Dose Taken.'"	The instruction requires the screen to display the text 'Morning Dose Taken.'. In the actual video, the text does not appear or is illegible, which does not match the instruction's requirements.	2	4	–	–
"He walks into glass door"	The man in the frame never bumps into the glass door; instead, it is the man outside the door who bumps into it.	5	5	0.00–2.83	yes
"A fountain's water flows sideways and up"	The fountain by the road sprays outwards, but the water still flows downwards.	3	4	2.80–4.97	yes
"only birdsong"	There is no audio in the video, and no birdsong can be heard, which does not match the prompt's requirements.	3	4	0.00–4.97	–

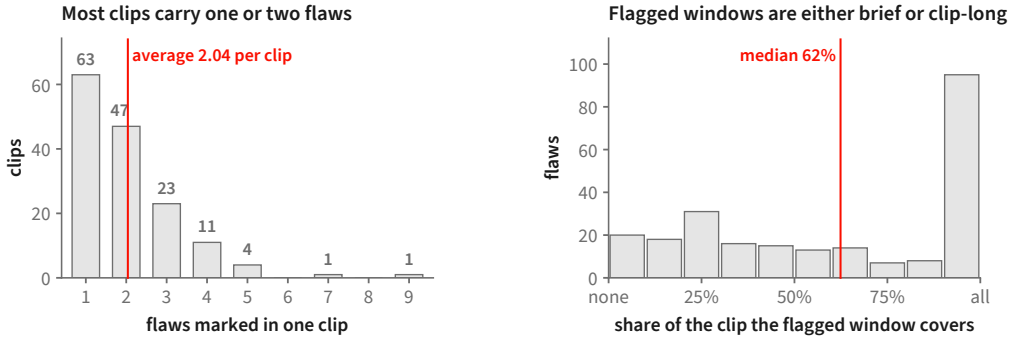


Figure 8: Frozen-core statistics (150 clips, 306 flaws). Left: flaws per clip; most clips carry one or two. Right: share of its clip covered by each of the 237 time-spanned flaws; windows are either brief or nearly clip-long. The graphic's "routine evaluation core" is the development core.

of these span at most a quarter of their clip and 95 at least nine tenths, and the median window lasts 5.00 s and covers 62% of its clip. The core therefore calls for both localized and whole-clip checks.

Machine-assigned type tags. A majority of three zero-temperature votes from a language model assigns each flaw a modality tag (seen, heard, or both) and one of ten type tags. On the 306 core flaws the type tags split as *action or event occurs* 139, *how many* 36, *object identity* 30, *on-screen text* 25, *spatial relation* 19, *order and timing* 17, *attribute* 16, *camera and style* 10, *physical motion* 7, and *other* 7. The *how many*, *on-screen text*, and *spatial relation* tags together hold 80 flaws, and *physical motion* holds 7, so the core tests prompt adherence more than physics. The per-tag recall table of R1 (Table 10) uses the tags stored in the R1 run record, which list 37 *how many* and 18 *spatial relation* flaws among the 304 entries.

Evaluated clips and duplicate entries. One core clip with 2 flaws ended in a terminal infrastructure error (a model endpoint dropped connections). We exclude it and report development-core recall over 149 clips and 304 flaw entries, without counting its flaws as missed; the excluded clip belongs to the continuity stratum. Among the 304 entries, 18 repeat a span already listed for the same clip,

Table 9: Development-core flaw recall (149 clips, 304 flaw entries). The VeriPhy row is run R1 and is the only row with the review stage; 83 of its 228 found flaws match findings from the review stage. Match rate is the share of findings matched to a flaw. "–": not recovered.

Evaluator	Review	Findings	Match rate	Whole	Found (recall)	Calls/clip
VeriPhy	yes	871	26%	191	228 (75.0%)	14.1
Single pass, same backbone	no	–	–	–	222 (73.0%)	1.0
TIFA-style decomposition (reimpl.)	no	557	29%	125	164 (53.9%)	21.9
DSG-style variant (reimpl.)	no	–	–	–	161 (53.0%)	52.1

since 12 clips list one span two or three times under different identifiers, which leaves 286 distinct flaws. Recall counts entries: bipartite matching lets a finding match only one entry, so a duplicated span counts as found twice only when two findings match it.

Matching protocol. For each clip, the scoring unit is a sentence pair: the rationale of one annotated flaw and one evaluator finding. At temperature zero, a text judge labels the pair as a whole match (the same flaw), a partial match (the finding captures at least one component of a compound flaw), or a non-match. Whole and partial labels define the candidate edges of a maximum bipartite matching in which each finding and each annotated flaw can be used at most once. The resulting flaw label is whole, partial, or missed, and both whole and partial matches count toward recall. The judge retries unusable outputs, and the scorer emits no aggregate result if more than 2% of flaws lack a usable outcome. Table 18 identifies the judge prompt, and the supplement gives its full text. R1 and R2 use this protocol. R5 instead credits each flaw with the first finding labeled a match and gives no partial credit. Partial matches contribute 37 of the 228 flaws that VeriPhy finds and 39 of the 164 that the TIFA-style baseline finds.

Finding formats. The judge reads text only, and the evaluators emit findings in different formats. VeriPhy reports contradicted claims tied to a prompt span, the TIFA-style baseline reports claims derived from yes/no answers, and the single pass returns a free-text list.

F Development-core breakdowns, baselines, and cost

Unless labelled otherwise, all numbers here come from runs R1 and R2 on the development core (149 clips, 304 single-annotator flaw entries). They support Section 4.1.

Recall and cost. Table 9 gives findings, match rate, whole matches, recall, and calls per clip for each evaluator. VeriPhy has the highest recall, 75.0% at 14.1 calls per clip. The two decomposition baselines use more calls (21.9 and 52.1) and find fewer flaws (164 and 161). The TIFA-style baseline matches 29% of its 557 findings, against 26% of the 871 findings of VeriPhy.

Same-backbone comparison. VeriPhy finds 228 of 304 flaw entries, and a one-call baseline with the same open backbone finds 222. Of the 228 flaws that VeriPhy finds, 83 match findings from the review stage, so VeriPhy finds at least 145 flaws before the review stage.

Recall by type tag. By machine-assigned type tag (Table 10), *camera and style* (9 of 10) and *other* (6 of 7) rank above *on-screen text* (21 of 25), and the *how many* tag ranks lowest (18 of 37). VeriPhy finds 4 of the 7 *physical motion* flaws. The counting path takes the mode of a per-frame series and can return a confident zero (Section C).

Recall by clip property. Grouped by annotator severity, clip duration, flaws per clip, and the continuity stratum (Table 11), subgroup recall lies between 63% and 79% except the 3-flaw severity-1 cell at 100%. The lowest cells are flaws in clips of 6 to 8 s (29 of 46) and severity-5 flaws (18 of 28).

Baseline specifications. All three baselines use Qwen3-VL-30B-A3B-Instruct, the checkpoint that runs the planner and the semantic checks of VeriPhy, on the same 149 clips and without the review stage. The matching protocol of Section E scores their outputs. The single-pass baseline receives the prompt and all decoded frames in one call and returns a free-text flaw list, for 1.0 call per clip. Adapting Hu et al. (2023b) to video, the TIFA-style reimplementation turns each claim of VeriPhy

Table 10: Flaw recall of VeriPhy (run R1) by machine-assigned type tag on the development core (149 clips, 304 flaw entries; higher is better). Whole and partial are match outcomes (Section E), and Found is their sum. A language model assigned the tags.

Type tag	Flaw entries	Whole	Partial	Found	Recall
action or event occurs	137	100	14	114	83%
how many	37	10	8	18	49%
object identity	30	14	6	20	67%
on-screen text	25	21	0	21	84%
spatial relation	18	10	1	11	61%
order and timing	17	10	3	13	76%
attribute	16	10	2	12	75%
camera and style	10	7	2	9	90%
other	7	5	1	6	86%
physical motion	7	4	0	4	57%
All	304	191	37	228	75.0%

Table 11: Flaw recall of VeriPhy (run R1) by clip property on the development core (149 clips, 304 flaw entries; higher is better). Severity is the annotator's 1 to 5 grade of the flaw. Flaws per clip counts the annotated flaws of the clip that holds the flaw. The continuity stratum is the 29-clip stratum of the core draw, from which we excluded one clip (Section E).

Group	Flaw entries	Found	Recall
Annotator severity			
severity 1	3	3	100%
severity 2	28	22	79%
severity 3	146	107	73%
severity 4	99	78	79%
severity 5	28	18	64%
Clip duration			
6 s or less	76	60	79%
6 to 8 s	46	29	63%
over 8 s	182	139	76%
Annotated flaws per clip			
1	63	49	78%
2	92	65	71%
3 or 4	113	87	77%
5 or more	36	27	75%
Continuity stratum			
in stratum	83	60	72%
outside stratum	221	168	76%

into a yes/no question, asks the backbone over densely sampled frames, and reports claims answered in the negative. It averages 21.9 calls per clip. Its claim list matches that of VeriPhy on 148 of 149 clips. Following Cho et al. (2024), the DSG-style variant adds dependency gating and averages 52.1 calls per clip. Table 18 identifies all baseline prompts, whose complete text appears in the supplement.

Clip-by-clip comparison with the TIFA-style baseline. VeriPhy finds every flaw on 91 of the 149 clips. Clip by clip, it finds more flaws than the TIFA-style baseline on 70 clips, the same number on 63, and fewer on 16, and the baseline returns no finding on 12 clips.

CLIPScore diagnostic. CLIPScore (Hessel et al., 2021) returns one scalar per clip and no findings to match, so we use it as a clip-level diagnostic. Across the 149 clips it correlates with severity at $r = +0.035$ ($p = 0.67$) and with flaw count at $r = -0.024$ ($p = 0.77$). Of the 149 clips, 86 carry

Table 12: Where run R5 lost the flaws it missed (development core, 149 clips, 286 distinct single-annotator flaws, of which R5 found 197 and missed 89). Most misses sit on claims that VeriPhy checked and cleared, some of them weaker than the flaw, and 76 of those 79 clearances rest on a bare yes/no judgement.

Stage that lost the flaw	Flaws	Share of 89
Checked and cleared: a claim covering the flaw, or a weaker claim than the flaw, was cleared by VeriPhy	79	88.8%
cleared by a bare yes/no judgement with no measurement attached	76	
cleared with a measurement involved	3	
Flagged but uncredited: VeriPhy reported a finding that the scorer left unmatched	8	9.0%
Never decomposed: no claim covered the flaw	2	2.2%
Missed in total	89	

more than one flaw (241 of the 304 entries), and one clip-level scalar cannot indicate which of a clip's flaws it reflects.

Cost. In run R1, VeriPhy averages 14.1 model and tool calls per clip. The count covers planning, execution, and review calls and excludes matching calls. The single pass uses 1.0 call per clip, the TIFA-style baseline 21.9, and the DSG-style variant 52.1. A call to the planner, a SAM 3 video pass, and a review call each count as one call. The served Qwen3-VL model runs in BF16 with tensor parallelism across two GPUs. Over the 149 clips of run R5 (13.0 calls per clip; Section G), the median clip took 102 s, the mean 373.0 s, the 90th percentile 1086 s, and the slowest 6119 s. A minority of clips that invoke dense per-frame specialists pulls the mean above the median.

G Traces of a separate development run (R5)

This appendix provides the trace-level evidence underlying the failure attribution in Section 4.5. The analysis uses R5, a separate run of VeriPhy over the 149 evaluated clips of the development core. R5 has no review stage, and its scorer credits each flaw with the first finding labeled a match and gives no partial credit. Section B lists the configuration of R5.

The 149 clips carry 304 flaw entries. Eighteen of these entries repeat a span already listed on the same clip, because 12 clips list one span two or three times under different identifiers. R5's scoring credits a finding to at most one flaw and, by its construction, credits no repeated entry, so we count R5 against the 286 distinct flaws and give the 304-entry count beside it. Under this count R5 found 197 of the 286 distinct flaws and missed 89.

Where the misses arise. The trace makes each miss attributable to a specific stage (Table 12). A flaw is *checked and cleared* when a covering claim exists but VeriPhy accepts it, *flagged but uncredited* when VeriPhy emits a corresponding finding that the scorer leaves unmatched, and *never decomposed* when no claim covers the flaw. This decomposition follows directly from the claim-bound trace: a free-text flaw list provides no comparable per-claim execution history.

Learned yes/no judgements decided the cleared misses. The trace analysis points to learned semantic checks as the dominant source of cleared misses. Of the 79 checked-and-cleared flaws, 76 were decided without an attached measurement, while measurements account for 14.1% of all executed checks in R5. We identified no miss caused by a specialist returning an incorrect measurement. In roughly half of these 76 cases, the checked claim is weaker than the annotated requirement, because the plan drops a quantity, a spatial qualifier or part of a compound sentence. For the prompt requiring two people to lose balance, "arms flailing, and tumble overboard", VeriPhy checked only "Two people lose balance", which the semantic check supported.

Misses by type tag. Table 13 breaks R5 down by machine-assigned type tag. Its left block counts found flaws over all 304 entries, and its right block counts misses over the 286 distinct flaws. Repeated entries fall mostly under *how many* (10 of 18) and only 3 under *action or event occurs*, so collapsing duplicates changes the *how many* cell most.

Table 13: Run R5 by machine-assigned type tag (development core, 149 clips; higher found and lower missed are better). Left: flaws found by VeriPhy in R5 and by the per-claim variant, over all 304 entries including repeated spans. Right: distinct flaws, repeated entries removed, and flaws that VeriPhy missed.

	Over 304 entries			Over 286 distinct flaws		
Type tag	Entries	VeriPhy	Per-claim	Distinct	Repeated	Missed
action or event occurs	137	98	89	134	3	36
how many	36	18	16	26	10	8
object identity	30	18	18	30	0	12
on-screen text	25	19	15	23	2	4
spatial relation	19	13	11	18	1	5
order and timing	17	11	11	16	1	5
attribute	16	6	5	15	1	9
camera and style	10	5	5	10	0	5
other	7	5	4	7	0	2
physical motion	7	4	5	7	0	3
Total	304	197	179	286	18	89

Table 14: VeriPhy against per-claim verification within run R5 (development core, 149 clips, one run each; higher found is better). VeriPhy finds 18 more of 286 distinct flaws, with a 95% interval of $+8$ to $+28$ from a paired bootstrap over clips with 10,000 resamples, at 0.9 more calls per clip.

	Found			
Condition	of 286 distinct	of 304 entries	Calls/clip	Match rate
VeriPhy (plan-based, all fixed specialists)	197 (68.9%)	197 (64.8%)	13.0	40.1%
Per-claim verification	179 (62.6%)	179 (58.9%)	12.1	40.2%
Difference	$+18$ [$+8$, $+28$]	$+18$	$+0.9$	

Match rate is the share of findings that R5's scorer credited to a flaw.

Comparison with per-claim verification. R5 also ran a per-claim verification variant that checks one claim at a time on the same clips. VeriPhy found 18 more distinct flaws than this variant, at 0.9 more calls per clip (Table 14). Section F reports the wall-clock time per clip of R5.

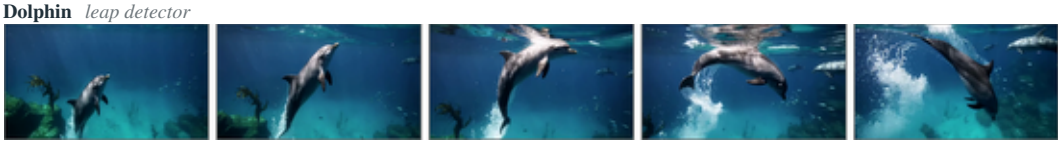
H MEASUREMENT CASES FROM AN EARLIER ROUTING PASS (R3)

Figure 9 shows five frames of each case with the annotator's flaw sentence and the recorded measurement. In every case the measured value misses the value the claim requires, such as 5 babies counted against 7 asserted and 4 jets against 3, so the claim resolves to contradicted. Table 15 gives six concrete examples in which an explicit specialist measurement identifies an annotated flaw. They come from R3, an earlier routing pass of VeriPhy on the development core. We inspected its per-clip records after 79 clips had completed. Five examples are resolved from numeric measurements, while the text example uses a recognition-head transcript followed by deterministic string comparison.

Selection. We selected cases by one rule: the run linked every annotated flaw on the clip to a finding, and the raw per-check record attributed that finding to a specialist measurement and stored its values. The rule excluded 10 candidates for mismatched linkage between finding and flaw, for a finding that the semantic checks produced, or, once, for an invalid rationale that read a "vertical 9:16" aspect ratio as a 9.0 versus 16.0 second duration. We then checked each of the six against every decoded frame of its clip.

What each measurement separates. Each measurement separates the prompt's requirement from a nearby condition that a whole-clip impression would accept. The dolphin swims up and down, so a test for any motion would pass, while its largest excursion by the hop measure stays at 0.00 radius. The bottle's surface moves at the rate of the background flow, so its apparent rolling is camera motion. The sun check scores peak departure from the starting height, so a rise followed by

Dolphin *leap detector*



Annotator: The dolphin ... merely swims to the surface and then dives back down without jumping out of the water.
Measured: hops: required yes, measured no; largest vertical excursion 0.00 of the dolphin's own radius over 37 tracked frames; 0 excursions reach the 0.30-radius threshold

Water bottle *surface motion vs. background flow*



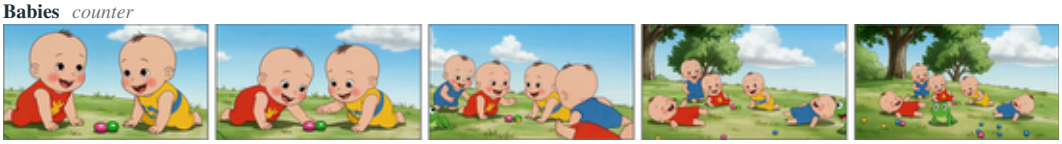
Annotator: The water bottle ... remains in its original position and never rolls on the floor.
Measured: rolling: required yes, measured no; surface moves 10.20 px/frame against 11.00 px/frame of background flow, over 256 frame pairs

Sun *vertical travel, camera cancelled*



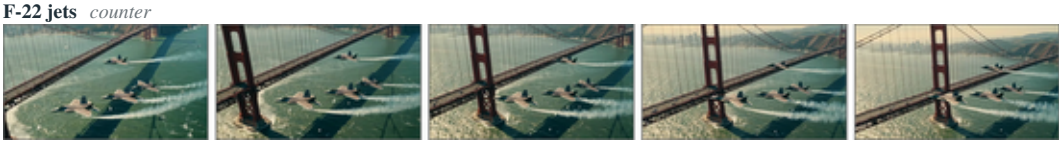
Annotator: The sun remains completely stationary in its original position without slowly rising.
Measured: vertical travel: required up; largest departure from its starting height 0.71 radius, under the 0.75 that counts as travel; net -2 px over 192 tracked frames

Babies *counter*



Annotator: There are only five babies in the video, which does not match the quantity requested in the text prompt.
Measured: count: prompt asserts 7, measured 5 (exact mode)

F-22 jets *counter*



Annotator: A total of four fighter jets appear in the video, which does not match the requested quantity of three.
Measured: count: prompt asserts 3, measured 4 (exact mode)

Graduation banner *OCR recognition head*



Annotator: The text appearing in the video does not match the requested text “Congratulations to the graduates”.
Measured: text: required 'CONGRATULATIONS GRADUATES', read 'GOBADRAEAT(CRSTRORISEUND'; stable wrong transcript on 25 of 241 frames (stride 10)

Figure 9: All six flaws that VeriPhy finds through measurement tools in the routing pass, in the layout of Figure 2.

a fall cannot cancel to zero net travel, and the peak stays below threshold. The two counts compare asserted and measured values in exact mode. The OCR head returns the rendered glyphs, and the transcript fails string equality with the required text.

Independent measurements on one clip. Two further specialists contradicted a second requirement on the dolphin clip, each on its own evidence. The prompt required the dolphin to disappear, and

Table 15: Six flaws found through a specialist measurement in the earlier routing pass R3 (development core; selected by the rule in the text). Each row gives the claim the planner wrote, the annotator's flaw sentence verbatim, the specialist's required and measured values as the run recorded them, and the resulting claim state. Displacements are in units of the subject's own radius, so no scene calibration enters.

Claim checked	Annotator's flaw sentence	Specialist: required vs. measured	State
A dolphin leaps out of the water	"The dolphin … merely swims to the surface and then dives back down without jumping out of the water."	Hop detection: required a hop; the operator counts excursions that rise above and return below a 0.30-radius threshold and reports the largest of them; 0 excursions qualified over 37 tracked frames, so it reports 0.00 radius	contradicted
The water bottle is rolling on a car floor.	"The water bottle … remains in its original position and never rolls on the floor."	Rolling: required rolling; surface moves 10.20 px/frame against 11.00 px/frame of background flow over 256 frame pairs	contradicted
The sun is rising.	"The sun remains completely stationary in its original position without slowly rising."	Vertical travel, camera motion cancelled: required upward travel; peak departure 0.71 radius, below the 0.75 threshold; net -2 px over 192 tracked frames	contradicted
There are seven babies	"There are only five babies in the video, which does not match the quantity requested in the text prompt."	Count, exact mode: asserted 7, measured 5	contradicted
Three F-22 Raptor fighter jets are in tight formation.	"A total of four fighter jets appear in the video, which does not match the requested quantity of three."	Count, exact mode: asserted 3, measured 4	contradicted
A massive 'CONGRATULATIONS GRADUATES' banner is being unfurled across a university building.	"The text appearing in the video does not match the requested text 'Congratulations to the graduates'."	Recognition-head OCR: required `CONGRATULATIONS GRADUATES`; read `GOBADRAEAT(CRSTRORISEUND`, a stable transcript on 25 of 241 frames read at stride 10	contradicted

the annotator recorded that it "does not disappear into the depths of the ocean". The vertical-travel check required downward motion and, on its own partial silhouette track of the same clip, measured 2.07 radii of upward travel (-317 px over 37 frames, where negative denotes up). The persistence check required the dolphin to vanish and found it on 121 of 121 frames, with no gap of 3 or more frames.

Coverage. Each measurement record also stores its temporal coverage. In the banner case the recognition head read 25 of the clip's 241 frames at stride 10. A transcript stable across those 25 frames indicates a persistent banner. Section C gives the OCR configuration.

Table 16: Agreement with human semantic-adherence ratings on 200 VideoPhy-2 test clips, 55 of them in the VideoPhy-2 hard subset (1 to 5 scale; higher is better in every column). VeriPhy trails VideoScore2 on exact and within-one agreement and exceeds it on both correlations. The 95% intervals of the differences are in the text.

	All clips ($n = 200$)				Hard subset ($n = 55$)	
Evaluator	Exact	Within 1	PLCC	SRCC	PLCC	SRCC
VideoScore2 (He et al., 2025)	33.0%	83.0%	0.353	0.327	0.239	0.151
VeriPhy	31.0%	76.0%	0.464	0.465	0.480	0.460

I VideoPhy-2 clip-level agreement (R7)

This appendix evaluates a complementary property to flaw recall: whether the clip-level semantic-adherence scores of VeriPhy preserve the ordering of human clip ratings. Flaw recall evaluates localized findings, whereas the VideoPhy-2 labels summarize each clip as a whole.

Setup. We drew 200 clips from the VideoPhy-2 test split (Bansal et al., 2026) with a fixed seed, and 55 of them belong to the VideoPhy-2 hard subset. We compare VeriPhy with VideoScore2 (He et al., 2025) against the human semantic-adherence rating on the 1 to 5 scale, and we score both with VideoScore2's released evaluation script. VeriPhy outputs claim states and a clip decision, and a fixed mapping converts them to a 1 to 5 score. We report exact agreement, agreement within one point, the Pearson linear correlation coefficient (PLCC), and the Spearman rank correlation coefficient (SRCC), and we drop any unparsed prediction from the metrics it affects. We exclude the physical-commonsense rating because VeriPhy emits no score on that scale.

Results. Table 16 reports the four metrics for both evaluators. VeriPhy trails VideoScore2 by 2 points on exact agreement (31.0% against 33.0%) and by 7 points on agreement within one (76.0% against 83.0%). It has higher PLCC and SRCC on all clips and on the hard subset, where the SRCC of VideoScore2 falls to 0.151 and that of VeriPhy stays at 0.460.

Differences. A paired bootstrap over clips gives the differences of VeriPhy minus VideoScore2. On all 200 clips, SRCC rises by $+0.138$ (95% interval $+0.005$ to $+0.277$) and PLCC by $+0.110$ (-0.018 to $+0.242$). On the 55 hard-subset clips, PLCC rises by $+0.242$ ($+0.018$ to $+0.511$) and SRCC by $+0.309$ ($+0.050$ to $+0.583$).

Summary. VeriPhy produces a stronger rank correlation with human semantic-adherence ratings than VideoScore2: the SRCC difference is positive with a bootstrap lower bound of $+0.005$. VideoScore2, however, achieves higher exact and within-one rating agreement.

J Prompt-rewrite loop details (R10)

Protocol. We ran one round of evaluate, rewrite, and regenerate on Cosmos3-Nano over 600 clips generated from VideoPhy-2 test prompts (Bansal et al., 2026). VeriPhy first scores each baseline clip against its prompt p. When VeriPhy acts on a clip, a language model rewrites the prompt from the failures that VeriPhy cites. The generator then resamples from the new prompt with the seed of the baseline clip:

$$p' = \begin{cases} \mathrm{Rw}(p, \mathcal{F}) & \text{if } \mathcal{F} \neq \emptyset, \\ p & \text{otherwise,} \end{cases} \qquad \mathcal{V}' \begin{cases} \sim \mathrm{Gen}(p', \xi) & \text{if } \mathcal{F} \neq \emptyset, \\ = \mathcal{V} & \text{otherwise,} \end{cases}$$

where Rw is the rewrite of Section 2.6, $\mathcal{F}$ holds the failures that VeriPhy cites with their records, and ξ is the seed. The loop changes only the prompt: Cosmos3-Nano receives no control input, and its weights stay fixed. The loop acts on 87 of 589 clips (14.8%), those for which VeriPhy cites at least one failure. Of these 87, 65 carried an implausible clip decision and 22 a plausible or abstain decision. For the 502 clips the loop leaves unchanged, the prompt stays p, and VeriPhy scores each clip a second time. Of the 600 clips, 589 completed with a paired score, and the other 11 lack a completed regeneration or re-score.

We scored every clip in two ways. The first is the verdict of VeriPhy. The second is VideoPhy-2's AutoRater (Bansal et al., 2026), which returns a semantic-adherence (SA) score and a physical-

Table 17: VideoPhy-2's AutoRater on Cosmos3-Nano clips before and after one round of prompt rewrites triggered by VeriPhy (higher is better in every row). Joint agreement moves from 22.8% to 23.1% over all clips, and mean PC from 3.59 to 3.58. Baseline columns rest on 600 clips and rewrite columns on the 589 that completed the loop.

	All clips		VideoPhy-2 hard subset	
AutoRater metric	Baseline ($n = 600$)	Rewrite ($n = 589$)	Baseline	Rewrite
Mean semantic adherence (SA)	3.12	3.13	2.75	2.76
Mean physical commonsense (PC)	3.59	3.58	3.36	3.32
SA ≥ 4	27.8%	28.0%	8.9%	7.9%
PC ≥ 4	55.0%	54.5%	37.2%	36.2%
Joint (SA ≥ 4 and PC ≥ 4)	22.8%	23.1%	3.9%	3.4%

commonsense (PC) score on a 1 to 5 scale; joint agreement is the share of clips with both scores at least 4. The AutoRater plays no part in the loop and shares no component with VeriPhy.

VeriPhy's reading. By the verdicts of VeriPhy, 49 of the 65 acted clips it first judged implausible became plausible after the rewrite. Of the 502 clips the loop left unchanged, 8 (1.6%) changed from plausible to implausible.

The independent rater's reading. Table 17 gives every AutoRater metric. Joint agreement moves from 22.8% to 23.1% over all clips and from 3.9% to 3.4% on the VideoPhy-2 hard subset. Mean PC moves from 3.59 to 3.58 over all clips and from 3.36 to 3.32 on the hard subset.

Curated cases. We chose the before-and-after pairs of Figure 4 by a frame-by-frame review that kept a pair only when both clips showed the same scene and the rewrite introduced no new artifact. Figure 10 gives each pair with its rewritten prompt.

K PROMPTS

The supplement contains the exact prompt text of the planner, the semantic checks, and the binary matching judge that scores run R5, and the lesson file with all 33 lessons. VideoPhy-2's AutoRater and VideoScore2 use their released prompts and scripts. Table 18 lists the eleven model prompts, the model that receives each, and the section that describes its use. Six go to the Qwen3-VL-30B-A3B-Instruct checkpoint of VeriPhy, including the three baseline prompts, and the review prompt goes to GPT-5.6 Sol.

Table 18: Model prompts, the model that received each prompt, and where the paper describes its use.

Prompt	Role	Model	Described in
Planner	Writes claims and the check plan from the generation prompt before VeriPhy decodes any frame; lessons append to this prompt	Qwen3-VL-30B-A3B-Instruct	Sec. 2.1
Semantic checks	Learned semantic sub-checks and the plan fallback	Qwen3-VL-30B-A3B-Instruct	Sec. 2.3
Grounding guard	Shows one frame with candidate boxes and asks whether they hold the named subject	Qwen3-VL-30B-A3B-Instruct	Sec. 2.3; App. C
Review stage	Re-examines supported claims after the roll-up	GPT-5.6 Sol	Sec. 2.4
Matching judge	Labels each finding and flaw pair as whole, partial, or non-matching	Text-only language model	Sec. 3; App. E
Type-tag voters	Three zero-temperature votes per flaw record	Language model	App. E
Single pass	One call over the prompt and frames that returns a free-text flaw list	Qwen3-VL-30B-A3B-Instruct	Sec. 4.1; App. F
TIFA-style decomposition	One yes/no question per claim of the VeriPhy claim list	Qwen3-VL-30B-A3B-Instruct	Sec. 4.1; App. F
DSG-style variant	Generated questions with dependency gating	Qwen3-VL-30B-A3B-Instruct	Sec. 4.1; App. F
Lesson distiller	Reads a trace and the missed flaw record and writes a candidate lesson	Language model	Sec. 2.5; App. D
Prompt rewriter	Rewrites a generation prompt from the contradicted claims of VeriPhy	Language model	App. J

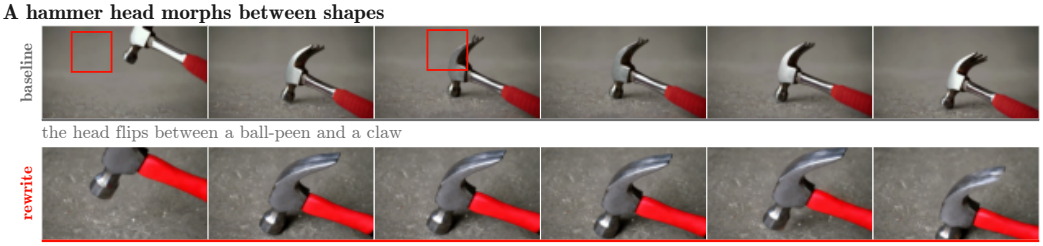
A ball appears from nothing



Original prompt: A baseball player throws a fastball; the ball leaves their hand and travels towards the catcher's mitt.

Rewritten prompt: In a sun-drenched baseball field, a dynamic moment unfolds as a player in a sleek black jersey and white pants stands poised on the mound, their right arm drawn back with a baseball firmly gripped and clearly visible in their palm. The camera captures the scene from a low angle, emphasizing the athlete's intense focus and the tension of the impending pitch. As the arm snaps forward, the ball is released with a continuous, fluid motion, maintaining its spherical shape and white leather texture as it travels in a realistic parabolic arc through the air, illuminated by the golden-hour glow. The background blurs into a vibrant tapestry of green grass and distant spectators, enhancing the sense of speed and precision without distorting the environment. The catcher, clad in protective gear, crouches expectantly, mitt poised to receive the fastball, while the umpire stands resolutely behind the plate, embodying the spirit of the game. This cinematic tableau, rich with detail and emotion, encapsulates the essence of baseball's thrilling artistry, ensuring the ball's trajectory and the player's biomechanics adhere strictly to the laws of physics.

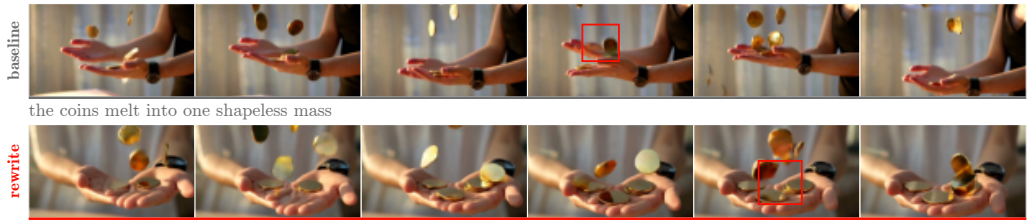
A hammer head morphs between shapes



Original prompt: A hammer is thrown and bounces once before stopping; the bounce and final resting position are clearly seen.

Rewritten prompt: A close-up, static shot captures a solid steel hammer with a vibrant red grip poised in mid-air, its head glinting under soft, diffused lighting that accentuates its metallic sheen. The camera remains fixed as the hammer is released, its trajectory arcing naturally under gravity until it strikes a textured surface with a sharp, audible impact. Upon contact, the hammer undergoes a realistic, brief elastic compression at the head before rebounding upward with a loss of kinetic energy, clearly showing a single, distinct bounce rather than a continuous glide. The hammer maintains its rigid, unchanging form throughout the motion, with no deformation, splitting, or morphing of the claw or handle. As it settles after the bounce, it comes to a complete rest on the ground, its head tilted slightly, while the camera subtly shifts to keep the object in frame, ensuring the entire sequence of release, impact, bounce, and final stop is captured with continuous, momentum-conserving motion.

Coins fuse into a blob



Original prompt: A person skillfully tosses several coins in the air, catching them one by one.

Rewritten prompt: In a serene, sunlit room, a skilled individual stands poised, their hands gracefully positioned in front of them. The camera captures a close-up of their hands, adorned with a sleek black watch, as they deftly toss a series of rigid, gleaming gold coins into the air. Each coin maintains its solid circular disk shape throughout its trajectory, following a smooth, continuous parabolic arc under gravity without distorting, splitting, or teleporting. The coins collide with the palm and fingers with clear impact, resting stably on the skin surface supported by normal force, never interpenetrating the hand or losing their metallic integrity. The background is softly blurred, enhancing the focus on the precise choreography of the tosses. The scene is bathed in a warm, golden-hour glow, casting gentle shadows that accentuate the coins' unchanging metallic sheen. As the coins are caught, the individual's hands remain steady, embodying a sense of calm control, while the camera maintains a static position, allowing the viewer to fully appreciate the artistry of this physically consistent display.

A frisbee changes shape in flight



Original prompt: An underhand throw sends a frisbee spinning across a beach, landing near the water's edge.

Rewritten prompt: In a sun-drenched beach scene, a vibrant frisbee is thrown underhand, maintaining its rigid, solid disc shape without any morphing, splitting, or color shifts as it travels through the air. The camera captures a continuous, unbroken trajectory from the thrower's off-screen hand to the water's edge, ensuring the disc never vanishes, teleports, or reappears at a