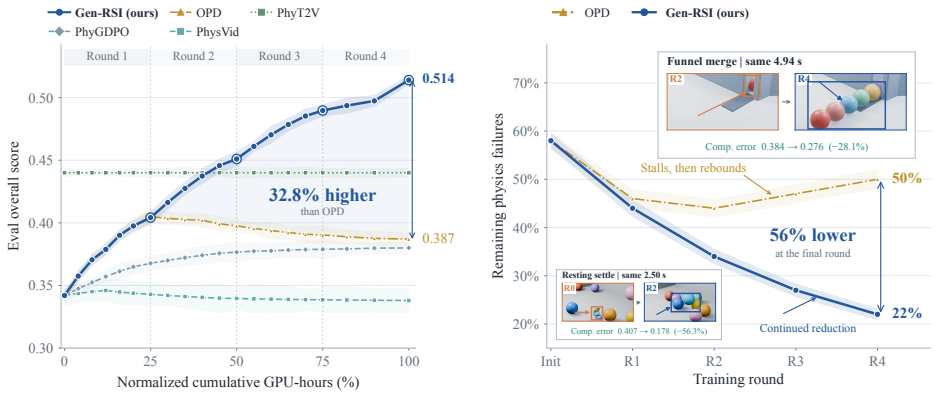
GEN-RSI: RECURSIVE SELF-IMPROVEMENT OF PHYSICAL VIDEO GENERATION THROUGH WORLD PROXIES

Anonymous authors
Paper under double-blind review



(a) Quality gain **(b) Physics failure decay**

Figure 1: **Gen-RSI rebuilds supervision every round to sustain improvement.** **(a)** PhyGenBench average on Wan2.1-1.3B: 0.514 for Gen-RSI versus 0.387 for OPD, which peaks after the first round and then declines. **(b)** Physics failures fall to 22% with Gen-RSI but rebound to 50% with OPD. Insets show proxy revisions.

ABSTRACT

Despite producing visually realistic videos, video generation models often violate physical laws in the motion and interactions they depict. Existing post-training methods usually rely on offline physics datasets to encode physical priors which may not address the model's current failures, or learned reward models that assess physical plausibility without providing explicit correction targets. We introduce **Gen-RSI**, a recursive self-improvement framework that constructs physical corrections from the model's own generations through executable world proxies. Specifically, a coding agent reconstructs the generated scene and motion as a Blender program and uses physics simulation to diagnose and repair physical violations. Physically verified repairs are rendered to condition a frozen video teacher, whose predictions provide on-policy distillation targets for the student generator at its own denoising states. Across rounds, Gen-RSI rebuilds correction targets from the updated generator and refines its agentic diagnosis and repair procedures using feedback from prior attempts. After four-round training on Wan2.1-14B, Gen-RSI improves the PhyGenBench average score from 0.369 to 0.558 and the mean score across the mechanics, thermotics, and material dimensions of VBench-2.0 from 73.09% to 80.00%. View the supplementary material video at https://anonymous-submission-21194.github.io/.

1 INTRODUCTION

Video generation models can produce realistic scenes and smooth motion (Team Wan, 2025; Team Seedance, 2026; HaCohen et al., 2026), yet their outputs may still violate physical laws (Meng et al., 2024; Zheng et al., 2025; Tragoudaras et al., 2025). The difficulty lies in the generated dynamics where object trajectories and interactions must respect gravity, contact, and support, even when

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.

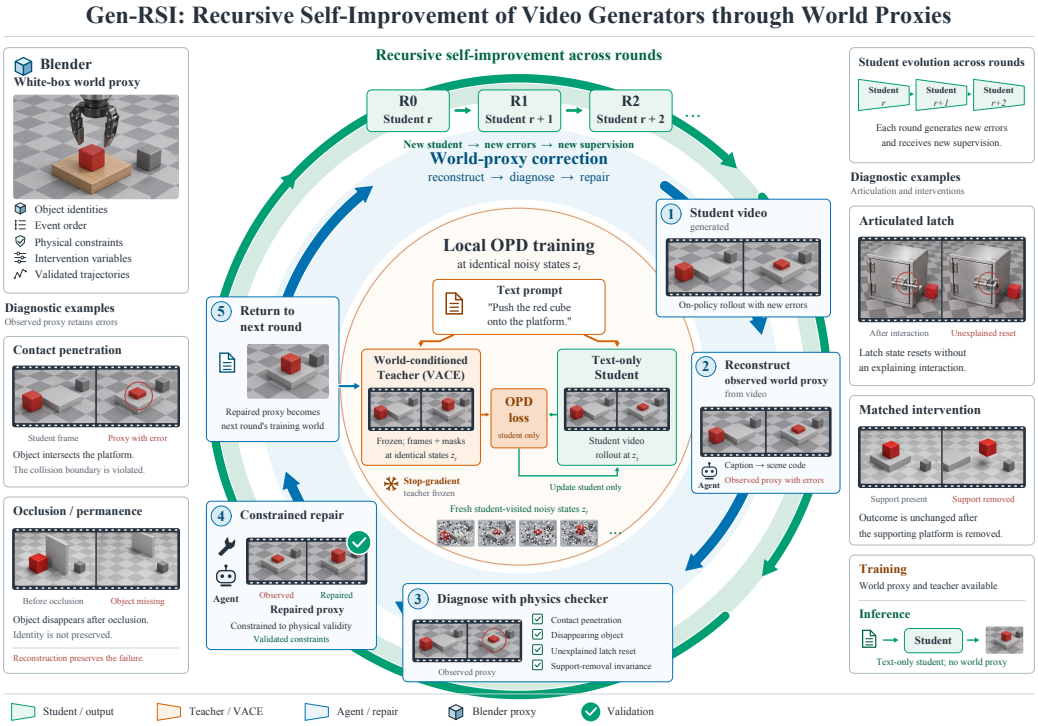


Figure 2: **Gen-RSI: one on-policy principle at three time scales.** Across rounds (green), an agent rebuilds the current student's videos as executable Blender proxies. World-proxy correction (blue) diagnoses and repairs them, and its operators are revised from verified outcomes. Local OPD (orange) queries a frozen VACE teacher at fresh student states and updates only the text-conditioned student, the only model kept at inference.

appearance alone provides little evidence of a violation (Motamed et al., 2025). For these models to serve as reliable world simulators (Wu et al., 2026; Zhang et al., 2026), they should improve how objects move and interact. This raises a concrete challenge for post-training: how to identify physical errors in the model's own generations and obtain reliable correction targets.

Existing post-training methods typically obtain physical supervision from offline datasets or learned reward models (Li et al., 2025; Zhang et al., 2025b; Cai et al., 2025; Le et al., 2025). Offline physics datasets encode physical priors through examples of valid motion and interactions (Li et al., 2025; Zhang et al., 2025b). However, these examples are collected in advance and are independent of online model behavior. Learned reward models assess the physical plausibility of the model's generations and guide optimization toward higher-scoring outputs (Cai et al., 2025; Le et al., 2025). Such scores indicate which outputs are preferable, but do not directly specify how an incorrect motion or interaction should be corrected. Consistent with this, PhyGDPO, which optimizes physics-aware preferences (Cai et al., 2025), levels off near 0.38 in Figure 1a. These limitations motivate a supervision scheme that targets the model's existing errors and provides explicit correction targets.

Our key idea is to use a coding agent coupled with Blender as an executable world proxy that constructs correction targets from the generator's own outputs. The supervision available to the generator therefore depends on which errors the proxy can reconstruct, diagnose, and repair. As training changes the generator's outputs, fresh samples expose its current errors, but fixed correction procedures can repeatedly fail to turn some of them into usable targets. Figure 1 shows what happens when supervision stops following the generator: without RSI, the PhyGenBench score peaks after the first round and then declines (a), and the rate of remaining physics failures stalls and rebounds to 50% (b). Gen-RSI therefore uses feedback from correction attempts to refine these procedures, retaining revisions only after independent validation. The insets of Figure 1b show two such revisions, each of which reconstructs the same moment of a generated video more faithfully than its

predecessor. This forms a recursive self-improvement loop (Good, 1965; Schmidhuber, 2003): corrections improve the generator, while experience from constructing those corrections improves the procedures used to produce supervision in subsequent rounds.

Building on this idea, we propose **Gen-RSI**, a recursive self-improvement framework for video generation (Figure 2). In each round, a coding agent reconstructs the current generator's outputs as executable Blender scenes and revises their dynamics using execution feedback and VLM assessments. Corrections accepted by a separate evaluator are rendered to condition a frozen video teacher (Jiang et al., 2025). Through on-policy distillation (OPD) (Zhao et al., 2026c; Jiang et al., 2026), the generator learns to match the teacher's predictions at its own denoising states. Across rounds, the agent uses successful and failed correction attempts to revise its reconstruction, diagnosis, and repair procedures, retaining independently validated revisions in a reusable tool library which are then applied to fresh outputs of the updated generator to construct the next round's supervision. Because both the supervision and the tools used to construct it are refreshed based on the current generator, improvement continues across rounds rather than stalling. As shown in Figure 1, for Wan2.1-1.3B, the PhyGenBench score rises to 0.514 after four rounds, a 32.8% improvement over training without RSI, while the proportion of remaining physics failures drops to 22%.

Our main contributions are:

- We introduce executable world proxies that turn physical errors in generated videos into simulator-verified correction targets, providing explicit supervision tailored to the generator's own failures.
- We propose **Gen-RSI**, a recursive self-improvement framework that adapts both correction targets and their construction procedures to the evolving generator. Validated tool revisions enable subsequent rounds to extend the coverage of physical correction supervision.
- Experiments on Wan2.1-1.3B and Wan2.1-14B demonstrate improvements in physical consistency on PhyGenBench and VBench-2.0. After four rounds on Wan2.1-14B, Gen-RSI raises the PhyGenBench average from 0.369 to 0.558, and the average across three evaluated VBench-2.0 physics dimensions from 73.09% to 80.00%.

2 Related Work

Physics supervision for video generators. Post-training methods fall into two groups. The first uses offline data: PISA trains on simulated videos of falling objects (Li et al., 2025), and VideoREPA aligns generator features with the representations of a video foundation model on real videos (Zhang et al., 2025b). Such data cover only scenes chosen in advance, so most of the model's own errors have no reference. The second group scores the model's own generations: PhyGDPO applies preference optimization to scores from a vision-language model (Cai et al., 2025), and NewtonRewards builds a verifiable reward from optical-flow acceleration (Le et al., 2025). The former judges only visual plausibility, the latter covers a few laws, and neither supplies the correct motion. Gen-RSI avoids the limitations of both groups: each round, it rebuilds the current student's generations as world proxies, diagnoses violations, and computes the correct motion in a simulator (Table 4).

Executable worlds and agentic verification. Existing work on executable worlds either reconstructs and evaluates, or intervenes at inference. Reconstruction and evaluation never feed back into the video generator: VIGA rebuilds an image as a Blender program through a code-render-inspect loop (Yin et al., 2026), Code as Worlds translates text or video into simulation programs to train vision-language models (Wang et al., 2026a), and VeriPhy evaluates generated videos with expert-tool measurements (Xu et al., 2026). Inference-time methods leave the generator unchanged and rerun their full pipeline for every generation: PhyT2V iteratively rewrites the prompt with a language model (Xue et al., 2025), and VideoCoCo has an agent write a Blender program whose rendering is then edited into a video (Li et al., 2026a). Gen-RSI instead uses executable worlds to train the generator: it adopts the VIGA loop to rebuild the student's videos as world proxies, in which the agent reproduces the failures and the simulator computes the correct motion. The repairs are distilled into the student, which therefore requires no additional input at inference (Table 4, last column).

Recursive self-improvement. Recursive self-improvement (RSI) describes a system that improves both its capabilities and the mechanism by which it improves (Good, 1965; Schmidhuber, 2003; Tang et al., 2026). Existing RSI work concentrates on language-model agents (Chen et al., 2026a; Zhang et al., 2025a; Zheng et al., 2026; Fan & Huang, 2026; NeoHorse Team et al., 2026); to our knowledge, none post-trains video generators. Video self-improvement loops serve embodied planning (Soni et al., 2024; Luo et al., 2026), and their feedback or correction mechanisms stay fixed. Gen-RSI brings RSI to video generation: it verifies physics in a simulator and, at each round boundary, retains the practices that independent verification confirms, so its ability to improve the student grows with the rounds.

3 Our Method

3.1 Framework Overview

Gen-RSI uses an executable world proxy to construct correction references for training a text-to-video generator. The world proxy couples a coding agent with a Blender environment, allowing generated scenes and motion to be reconstructed, edited, and rendered as programs. Given a generated video and its prompt, the agent reconstructs the depicted event from sampled frames. It then iteratively revises the scene program using motion measurements and VLM feedback, executing and rendering each revision in Blender. A separate VLM evaluator assesses the rendered result for event fidelity and physical plausibility, and only candidates passing the acceptance checks are retained as correction references.

The accepted references condition a frozen video teacher, whose predictions at the generator's own denoising states provide targets for on-policy distillation. This transfers guidance from the corrected scenes into the generator's parameters. Across training rounds, fresh videos from the updated generator are processed through the world proxy to construct new references for its current outputs. The framework therefore links program-level correction with generator-level learning: the agent refines executable scenes through feedback, and the generator learns from the accepted results. At inference, only the trained text-to-video generator is required.

3.2 Executable World Proxies for Physical Correction

We construct correction references by representing generated events as executable Blender programs. Given a prompt c and a generated video x, the coding agent receives six frames sampled from x and arranged chronologically in a single image, together with a condensed text prompt, to reconstruct the scene and its motion. It reconstructs the event's objects, motion, and interactions as a scene program P^{obs}. Following the code-render-compare loop of VIGA (Yin et al., 2026), the agent renders candidate programs and revises them using paired comparisons with the input frames. The comparison evaluates event identity, object roles, contact and causal order, trajectories, and the final state, while disregarding differences in rendering style and proxy geometry.

Program refinement uses feedback from execution and the rendered motion. At refinement step k, the current scene program P_k is executed in Blender to produce a video $\mathcal{R}(P_k)$. We extract visual motion descriptors $\mathbf{m}(\mathcal{R}(P_k))$, including frame differences, optical-flow magnitudes, motion-centroid trajectories, and event peaks. These measurements summarize the rendered motion, while a VLM evaluator assesses event fidelity and physical plausibility. Together with execution diagnostics, this feedback guides the coding agent's revision:

$$P_{k+1} = \mathcal{A}_{\text{edit}}\left(P_k, c, \mathbf{m}(\mathcal{R}(P_k)), e_k\right), \tag{1}$$

where c is the text prompt and e_k contains the execution diagnostics and evaluator feedback.

The agent can revise motion parameters, collision geometry, interaction properties such as friction and restitution, keyframe timing, and simulation settings. Edits are guided by the identified errors, with instructions to preserve the camera, appearance, prompt-specified content, initial state, and unrelated events. Each revision returns a complete Blender program, which is executed and rendered over the full sequence to evaluate the effects of the changes and provide feedback for further refinement. The permitted edits and execution settings are detailed in Appendix B.

A separate VLM auditor reviews each rendered candidate on five dimensions: visible, process, causality, continuity, and physical plausibility. Each dimension receives an ordinal score $a_j(P)$, and

we write $A(P) = \sum_{j=1}^{5} a_j(P)$ for the total. A candidate must execute and render successfully, pass the media and quality checks, clear the acceptance threshold, and regress on no dimension relative to the preceding version:

$$A(P_{k+1}) \geq \tau_A, \qquad a_j(P_{k+1}) \geq a_j(P_k) \quad \forall j, \tag{2}$$

where $a_j(P)$ is the auditor's score for dimension j based on the program's rendered output. We set τ_A close to the maximum attainable total, so an accepted candidate falls short on at most one dimension by at most one grade; a high score on reconstruction fidelity therefore cannot compensate for a low one on physical plausibility. If a candidate fails, the auditor's assessment of each dimension is returned to the coding agent as feedback for the next revision. Refinement continues within a fixed retry budget; unsuccessful candidates are discarded. The rendering of an accepted program P^+ provides the visual reference for the frozen video teacher. The scale, the value of τ_A, and the per-dimension rubrics are given in Appendix E.

3.3 On-Policy Distillation from Verified Corrections

We transfer the verified corrections to the text-to-video generator through a frozen video-conditioned teacher. Each accepted program P^+ is rendered into frames and masks, forming a visual condition u for a VACE conditioning branch (Jiang et al., 2025). The teacher combines this branch with the initial generator and remains frozen throughout training. At latent state z_t and denoising level t, the generator predicts $f_\theta(z_t, c, t)$, while the teacher predicts $T_\phi(z_t, c, t, u)$ in the same parameterization. The teacher's predictions, conditioned on the rendered correction, serve as distillation targets.

At each update, we sample prompts with their correction references and run the current generator from fresh noise. We query the teacher at states visited along the resulting denoising trajectories (Zhao et al., 2026c; Jiang et al., 2026). Both models receive the same state and prompt, while the teacher additionally receives u. Correction references remain fixed within a round, but trajectories are resampled as the generator changes, keeping the training states on-policy. A scene-correspondence check excludes correction supervision for samples whose generated content no longer matches the associated reference; the criteria are specified in Appendix B.

For the set $\mathcal{I}$ of samples passing this check, the distillation objective is

$$\mathcal{L}_{\text{distill}}(\theta) = \frac{1}{\max(1, |\mathcal{I}|)} \sum_{i \in \mathcal{I}} \sum_{t \in \mathcal{T}_i} \omega_{it} \left\| f_\theta(z_{i,t}, c_i, t) - \text{sg}[T_\phi(z_{i,t}, c_i, t, u_i)] \right\|_2^2, \tag{3}$$

where $\mathcal{T}_i$ contains the sampled denoising levels, ω_{it} are nonnegative weights summing to one for each trajectory, and sg denotes stop-gradient. We optimize $\mathcal{L} = \mathcal{L}_{\text{distill}} + \lambda \mathcal{L}_{\text{keep}}$, where $\mathcal{L}_{\text{keep}}$ anchors the generator to its initial predictions on a fixed set of training-only reference states. When $\mathcal{I}$ is empty, the distillation term is zero and only the anchor is applied. Gradients do not propagate through trajectory sampling or teacher predictions; only the generator parameters θ are updated. Timestep weighting and anchor construction are detailed in Appendix B.

3.4 Recursive Self-Improvement

Each round produces an updated generator and feedback on how the world proxy constructed its supervision. Training changes the generator's errors, motivating new correction references in the next round. However, fresh samples provide supervision only when the world proxy can reconstruct, diagnose, and repair them. Gen-RSI therefore uses the round's correction outcomes to revise these procedures before constructing the next round's references.

The agent reviews the reconstruction programs, diagnostic results, repair attempts, and verification feedback collected during the round. It extracts reusable reconstruction and repair routines from successful attempts and revises procedures associated with failures or unresolved checks. These revisions may modify scene templates, constraint checks, or simulation setup code. Candidate revisions undergo a fixed, independent validation procedure before entering the tool library. The validation criteria are maintained outside the agent's revision process, so modifying a diagnostic or repair routine cannot change the criteria used to accept it. This update changes the agent's executable tools while keeping its model weights fixed.

After training and tool validation, we sample fresh videos from the updated generator θ_{r+1} and process them with the revised world proxy. Accepted corrections form the next round's references,

supplemented by a capped, category-balanced selection of earlier references for replay. Thus, each round changes both the samples requiring correction and the procedures available to correct them: the generator learns from verified repairs, and the world proxy incorporates feedback from producing those repairs. Algorithm 1 gives the complete loop. At inference, only the trained text-to-video generator is required.

4 EXPERIMENTS

4.1 EXPERIMENT SETUP

Data Curation. To construct training data, we begin by sampling about 3,000 text prompts that cover four physical domains: mechanics, optics, thermal, and material. Mechanics prompts involve contact, object permanence under occlusion, articulation, and intervention, the failure families illustrated in Figure 2. Optics prompts involve light phenomena such as reflection and refraction. Thermal prompts involve heat-driven processes such as combustion, and material prompts involve chemical changes such as acid-base reactions. In these three domains, the diagnosis targets the process named in the prompt, for example a reflection or refraction inconsistent with the scene geometry, a flame that leaves its fuel unchanged, or an acid-base color change that completes abruptly. At each round r, the current student π_{θ_r} renders every prompt c into a video x_r, which then goes through the reconstruct, diagnose and repair procedure of Section 3.2. Following the notation introduced earlier, this yields an observed proxy P_r^{obs} together with its verified repair P_r^+. We then render P_r^+ into the frames and masks $u(P_r^+)$ that the frozen teacher reads. In this way, we obtain records $(c, P_r^{\text{obs}}, P_r^+)$ for the world bank, shared by on-policy distillation and the RSI rounds and rebuilt from the updated student at every round.

Baselines. All methods share the same Wan2.1 backbone at each scale, so each comparison isolates the training method. We report the base generator and three physics-targeted methods. PhysVid (Pathak et al., 2026) adds local physics conditioning. PhyGDPO (Cai et al., 2025) applies physics-aware groupwise preference optimization. PhyT2V (Xue et al., 2025) refines the prompt through LLM-guided self-refinement. We also report OPD, the Gen-RSI ablation that removes world information from the loop and keeps only on-policy distillation.

Implementation. We use Wan2.1-T2V-1.3B and Wan2.1-T2V-14B (Team Wan, 2025) as students. Each teacher is the VACE variant of the same backbone (Jiang et al., 2025), which takes the world proxy as privileged information (Lopez-Paz et al., 2016). We fine-tune the student with LoRA (rank 64, α=64) and keep the teacher frozen. We optimize with AdamW at a learning rate of 1×10^{-4}. All video clips are resized to 854×480. Gen-RSI runs for 4 self-improvement rounds, with 200

	Method	Mechanics (↑)	Optics (↑)	Thermal (↑)	Material (↑)	Average (↑)
Wan2.1-1.3B	Base generative model	0.350	0.280	0.556	0.183	0.342
	+ PhysVid (Pathak et al., 2026)	0.375	0.247	0.511	0.217	0.338
	+ PhyGDPO (Cai et al., 2025)	0.342	0.373	0.578	0.225	0.380
	+ PhyT2V (Xue et al., 2025)	0.350	0.433	0.600	0.375	0.440
	+ OPD	0.392	0.340	0.600	0.217	0.387
	+ Gen-RSI (ours)	**0.475**	**0.493**	**0.644**	**0.442**	**0.514**
Wan2.1-14B	Base generative model	0.350	0.327	0.600	0.200	0.369
	+ PhysVid (Pathak et al., 2026)	0.375	0.287	0.556	0.233	0.363
	+ PhyGDPO (Cai et al., 2025)	0.342	0.440	0.622	0.250	0.414
	+ PhyT2V (Xue et al., 2025)	0.350	0.507	0.644	0.408	0.477
	+ OPD	0.392	0.400	0.644	0.233	0.417
	+ Gen-RSI (ours)	**0.475**	**0.580**	**0.700**	**0.475**	**0.558**

Table 1: **Physical consistency on PhyGenBench.** Scores on the $[0, 1]$ scale (Meng et al., 2024); Average is the mean of the four categories. OPD is the Gen-RSI ablation without world information. **Blue** and light blue mark the best and second-best result per column for each backbone.

	Method	Mechanics (↑)	Thermotics (↑)	Material (↑)	Average (↑)
Wan2.1-1.3B	Base generative model	71.21	70.29	70.18	70.56
	+ PhysVid (Pathak et al., 2026)	70.68	69.01	72.38	70.69
	+ PhyGDPO (Cai et al., 2025)	76.92	68.61	73.75	73.09
	+ PhyT2V (Xue et al., 2025)	67.46	63.97	69.09	66.84
	+ OPD	71.70	69.74	**74.42**	71.95
	+ Gen-RSI (ours)	**82.20**	**77.85**	73.97	**78.01**
Wan2.1-14B	Base generative model	70.00	64.66	84.62	73.09
	+ PhysVid (Pathak et al., 2026)	56.00	58.90	86.30	67.07
	+ PhyGDPO (Cai et al., 2025)	70.30	62.70	86.00	73.00
	+ PhyT2V (Xue et al., 2025)	78.70	61.70	86.40	75.60
	+ OPD	70.70	63.70	86.71	73.70
	+ Gen-RSI (ours)	**80.00**	**71.00**	**89.00**	**80.00**

Table 2: **Physical plausibility on VBench-2.0.** Percentages on the mechanics, thermotics, and material dimensions (Zheng et al., 2025); Average is their mean, not the full benchmark score. OPD is the Gen-RSI ablation without world information. **Blue** and light blue mark the best and second-best result per column for each backbone.

optimization steps per round. Both scales share this setup, with a global batch size of 64 on 64 NVIDIA A800 GPUs. On-policy rollouts use 10 sampling steps during training, and inference uses 50 steps.

Evaluation. We evaluate on two physics-focused video benchmarks. PhyGenBench (Meng et al., 2024) tests physical commonsense in four categories: mechanics, optics, thermal, and material. We follow its official protocol, which uses GPT-4o (Hurst et al., 2024) as a multimodal judge to score how faithfully a video follows the physical principle implied by its prompt. Each category receives a consistency score in $[0, 1]$, and we report the average of the four. VBench-2.0 (Zheng et al., 2025) targets intrinsic faithfulness beyond surface quality. We adopt its per-dimension protocol on its three physics dimensions: Mechanics (e.g., gravity, buoyancy, stress), Thermotics (e.g., vaporization, freezing), and Material (e.g., color mixing, solubility). Each dimension has a dedicated prompt suite and reports plausibility as a percentage. We average the three, so this average is not the full VBench-2.0 score. Higher is better on every metric.

4.2 Main Results

PhyGenBench. Table 1 compares Gen-RSI against the baselines on both backbones. Adding Gen-RSI to Wan2.1-14B raises the average consistency score from 0.369 to 0.558, the best result in the table, ahead of the strongest baseline PhyT2V at 0.477. Gen-RSI attains the best per-category score on mechanics (0.475), optics (0.580), thermal (0.700), and material (0.475). The largest gains over the base generator appear on material ($+0.275$) and optics ($+0.253$), precisely the categories where appearance-driven generators, lacking an explicit dynamics reference, are weakest. The ordering repeats at 1.3B, where the average rises from 0.342 to 0.514 and material gains $+0.259$. Simulator-verified correction thus helps most where visual priors are weakest, which confirms that the world proxy contributes physical dynamics rather than surface realism.

VBench-2.0. Table 2 reports physical plausibility on VBench-2.0. Wan2.1-14B alone scores 73.09% on average, and adding Gen-RSI raises the average to 80.00%, a gain of 6.91 points over the base model and the best average among all methods. Gen-RSI attains the best mechanics (80.00%), thermotics (71.00%), and material (89.00%) scores, improving the base generator by 10.00, 6.34, and 4.38 points respectively. PhyT2V is the closest competitor on mechanics at 78.70%, but it falls below the base generator on thermotics at both scales. At both scales, the largest gain is on mechanics, at 10.00 points for 14B and 10.99 for 1.3B. All three dimensions improve together instead of trading one against another.

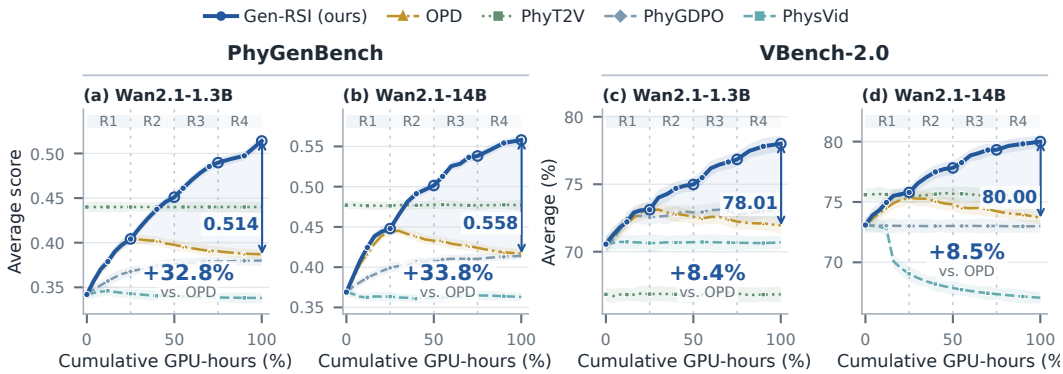


Figure 3: **RSI ablation across benchmarks and backbone scales.** Benchmark score against normalized cumulative GPU-hours under a matched budget. R1–R4 mark the four RSI rounds.The tinted wedge and the arrow show the gap between Gen-RSI and OPD at full budget. Single-shot baselines have no round structure; PhyGDPO and PhysVid are plotted as one training run under the same budget, and PhyT2V, which does not update the generator, stays flat.

Where the gain comes from. On Wan2.1-14B, stripping world information from the loop leaves OPD at 0.417 average on PhyGenBench and 73.70% on VBench-2.0. That recovers under 30% of the Gen-RSI gain on PhyGenBench and under 20% on VBench-2.0, at both backbone scales. On-policy distillation alone therefore accounts for a minority of the improvement, and the verified correction signal accounts for the rest.

4.3 Ablation Study

RSI across rounds. The tables report the endpoint of training, which cannot separate a method that converges early from one that keeps improving. Figure 3 therefore plots the whole budget. Under a matched compute budget, Gen-RSI is still climbing through the fourth round in all four panels. In contrast, OPD declines after the first round, and each single-shot baseline stays within 2% of its second-round score. At full budget it ends 32.8% and 33.8% above OPD on PhyGenBench, and 8.4% and 8.5% above it on VBench-2.0.

OPD runs the same rounds under the same budget but without world information. Its curve peaks near the first round boundary and then declines at both scales on both benchmarks. The gap to Gen-RSI therefore comes from the verified world information rather than from running more rounds, and it widens in every round after the first.

RSI components. Figure 4 removes one component at a time, two for each part of Section 3: the world proxy (skip the repair, or accept every repair without verification), local OPD (drop the world condition, or train on cached off-policy latents), and recursion (fix the world bank after the first round, or freeze the correction machinery). All variants start at 0.34, and Full Gen-RSI gains in every round to reach 0.51. Repair is the critical component. Without it, the score rises to 0.39 and falls back to 0.35, because a target that keeps the failure erodes the student. Without the verifier or on-policy states, the score moves up and down between rounds and ends at 0.40 and 0.38. Without bank refresh or the world condition, it stays nearly flat after the first round, between 0.39 and 0.42. Without machinery revision, it reaches 0.44 but stops short of 0.51.

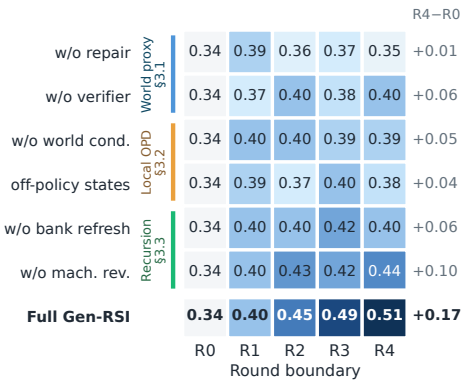


Figure 4: **RSI components.** PhyGenBench average per round on Wan2.1-1.3B after removing one component.



Figure 5: **Qualitative comparison with physics-targeted baselines on material prompts.** All methods use Wan2.1-T2V-1.3B; each row shows one method at the same three time points. Figure 10 adds the mechanics cases.

Local OPD, SFT, and RL. Table 3 keeps the world bank, rounds, and GPU budget fixed and varies only how the student learns from the repairs. SFT fine-tunes the student on the rendered repairs $u(P_r^+)$, so its training states come from the proxy. For RL we adopt PhyGDPO (Cai et al., 2025), which keeps the student's own rollouts but replaces the dense teacher target with groupwise preferences ranked by the physics checker. Local OPD uses both the student's own states and a dense target at every denoising step. SFT reaches 0.441 on PhyGenBench and 74.85% on VBench-2.0, RL reaches 0.468 and 76.12%, and Gen-RSI reaches 0.558 and 80.00%, 1.9 and 2.3 times the gain of RL. Training on the student's own states helps, and a dense target at every step helps further.

Qualitative comparison. Figure 5 compares Gen-RSI with the base model, PhysVid (Pathak et al., 2026), and PhyT2V (Xue et al., 2025) on two material prompts, and Figure 10 in Appendix D.3 adds two mechanics prompts. The baselines show two recurring failure modes. In mechanics scenes, they lose temporally consistent object states: objects change appearance or number across frames, stay suspended against gravity, or interpenetrate on contact. In material scenes, they either omit the physical process in the prompt or render its outcome without a plausible temporal progression. Combustion leaves the material unchanged, or a color change completes abruptly. Gen-RSI preserves object identity and produces dynamics and state transitions consistent with the underlying physics. This matches its design: the supervision is built from the student's own failures and verified in a physics simulator.

Transfer	PhyGenBench (↑)	VBench-2.0 (↑)
Base generative model	0.369	73.09
SFT	0.441	74.85
RL	0.468	76.12
Gen-RSI (ours)	**0.558**	**80.00**

Table 3: **Local OPD, SFT, and RL** on Wan2.1-14B, with the same world bank and budget.

5 Discussion and Conclusion

Gen-RSI turns physical errors in a generator's own videos into explicit, simulator-verified correction targets. Executable world proxies reconstruct, diagnose, and repair generated events in Blender, and OPD transfers accepted repairs at the generator's own denoising states. Across rounds, the agent also revises these procedures, keeping only independently validated revisions, so later rounds can correct failures that earlier procedures missed. Without this recursion, the score peaks after the first round and then declines; with it, the score keeps rising through four rounds. Proxies can still misreconstruct scenes or inherit simulator bias, yet our results show that recursive self-improvement is an effective path toward physically reliable video generation.

AI USE STATEMENT

Generative AI tools were used to assist with manuscript preparation, including language editing, and bibliography checks. Separately, Gen-RSI itself uses an LLM agent, GPT 6 Astra to write Blender proxies (Section 3.2), and PhyGenBench uses GPT as its official judge (Section 4.1). These are parts of the method and the evaluation protocol, not writing aids. The authors reviewed all AI-assisted work and remain responsible for the final text, claims, citations, experimental results, and artifacts in this submission.

REPRODUCIBILITY STATEMENT

The main text and appendix specify the information needed to rebuild and evaluate Gen-RSI. Section 3 defines different components. Algorithm 1 gives the full training loop, and Appendix B lists the implementation rules for each step. Section 4.1 reports data curation, backbones, the teacher, LoRA and optimizer settings, the round schedule, hardware, and sampling steps, and both benchmarks follow their official protocols. Appendix E defines the evidence each diagnostic family requires. Final-evaluation prompts never enter the training loop, and stopping rules use a development split disjoint from final evaluation.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. On-Policy Distillation of Language Models: Learning from Self-Generated Mistakes. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2306.13649.

Sina Alemohammad, Josue Casco-Rodriguez, Lorenzo Luzi, Ahmed Imtiaz Humayun, Hossein Babaei, Daniel LeJeune, Ali Siahkoohi, and Richard G. Baraniuk. Self-Consuming Generative Models Go MAD. *arXiv preprint arXiv:2307.01850*, 2023. URL https://arxiv.org/abs/2307.01850.

Thomas Anthony, Zheng Tian, and David Barber. Thinking Fast and Slow with Deep Learning and Tree Search. *arXiv preprint arXiv:1705.08439*, 2017. URL https://arxiv.org/abs/1705.08439.

Quentin Bertrand, Avishek Joey Bose, Alexandre Duplessis, Marco Jiralerspong, and Gauthier Gidel. On the Stability of Iterative Retraining of Generative Models on their own Data. *arXiv preprint arXiv:2310.00429*, 2023. URL https://arxiv.org/abs/2310.00429.

Kunlin Cai, Rui Song, Jinghuai Zhang, Kaiyuan Zhang, Pranav Bodapati, Alicia Yu, Fnu Suya, Mohammad Rostami, Jiaqi Ma, and Yuan Tian. What-If World: A Causal Benchmark for General World Models in Embodied Scenarios. *arXiv preprint arXiv:2605.27589*, 2026. URL https://arxiv.org/abs/2605.27589.

Yuanhao Cai, Kunpeng Li, Menglin Jia, Jialiang Wang, Junzhe Sun, Feng Liang, Weifeng Chen, Felix Juefei-Xu, Chu Wang, Ali Thabet, Xiaoliang Dai, Xuan Ju, Alan Yuille, and Ji Hou. PhyGDPO: Physics-Aware Groupwise Direct Preference Optimization for Physically Consistent Text-to-Video Generation. *arXiv preprint arXiv:2512.24551*, 2025. URL https://arxiv.org/abs/2512.24551.

Mingguang Chen, Licheng Wang, and Bo Qu. Recursive Self-Improvement in AI: From Bounded Self-Refinement to Autonomous Research Loops. *arXiv preprint arXiv:2607.07663*, 2026a. URL https://arxiv.org/abs/2607.07663.

Yiwen Chen, Guosheng Lin, and Chi Zhang. Code World Model: Coding Agent as World Brain. *arXiv preprint arXiv:2608.25927*, 2026b. URL https://arxiv.org/abs/2608.25927.

Wei Chow, Jiageng Mao, Boyi Li, Daniel Seita, Vitor Guizilini, and Yue Wang. PhysBench: Benchmarking and Enhancing Vision-Language Models for Physical World Understanding. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2501.16411.

Yi Duan, Ying Liu, Zirui Tang, Haodong Chen, Jun Zhou, Yumou Liu, Bangrui Xu, Yukai Wu, Sidi Chen, Yuhan Zhou, Haoyu Wang, Xiaoyou Yu, Shaokun Han, Xuzhou Zhu, Le Zhou, Bolin Lu, Wei Zhou, Jiachen Liu, Nuozhou Fang, Jiaxin Tian, Ruoyu Chen, Yuxuan Li, Kai Zuo, Kaiyan Zhang, Qianyu Yang, Zijie Wang, Jiantao Qiu, Conghui He, Guoliang Li, Bowen Zhou, Zhiyuan Liu, Zhoufutu Wen, Jihua Kang, Xuanhe Zhou, and Fan Wu. The Last AI Built by Humans: Toward Genuine Recursive Self-Improvement. *arXiv preprint arXiv:2609.11873*, 2026. URL https://arxiv.org/abs/2609.11873.

Tianyu Fan and Chao Huang. HELIX: Model-Harness Co-evolution for Recursive Self-Improvement. *arXiv preprint arXiv:2608.13951*, 2026. URL https://arxiv.org/abs/2608.13951.

Pengcheng Fang, Hongli Chen, and Xiaohao Cai. Privileged Foresight Distillation: Zero-Cost Future Correction for World Action Models. *arXiv preprint arXiv:2604.25859*, 2026a. URL https://arxiv.org/abs/2604.25859.

Zhen Fang, Wenxuan Huang, Yu Zeng, Yiming Zhao, Shuang Chen, Kaituo Feng, Yunlong Lin, Lin Chen, Zehui Chen, Shaosheng Cao, and Feng Zhao. Flow-OPD: On-Policy Distillation for Flow Matching Models. *arXiv preprint arXiv:2605.08063*, 2026b. URL https://arxiv.org/abs/2605.08063.

Nate Gillman, Michael Freeman, Daksh Aggarwal, Chia-Hong Hsu, Calvin Luo, Yonglong Tian, and Chen Sun. Self-Correcting Self-Consuming Loops for Generative Model Training. *arXiv preprint arXiv:2402.07087*, 2024. URL https://arxiv.org/abs/2402.07087.

John Gkountouras, Matthias Lindemann, Phillip Lippe, Efstratios Gavves, and Ivan Titov. Language Agents Meet Causality – Bridging LLMs and Causal World Models. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2410.19923.

Irving John Good. Speculations concerning the first ultraintelligent machine. In Franz L. Alt and Morris Rubinoff (eds.), *Advances in Computers*, volume 6, pp. 31–88. Academic Press, New York, 1965.

Caglar Gulcehre, Tom Le Paine, Srivatsan Srinivasan, Ksenia Konyushkova, Lotte Weerts, Abhishek Sharma, Aditya Siddhant, Alex Ahern, Miaosen Wang, Chenjie Gu, Wolfgang Macherey, Arnaud Doucet, Orhan Firat, and Nando de Freitas. Reinforced Self-Training (ReST) for Language Modeling. *arXiv preprint arXiv:2308.08998*, 2023. URL https://arxiv.org/abs/2308.08998.

Yoav HaCohen, Benny Brazowski, Nisan Chiprut, Yaki Bitterman, Andrew Kvochko, Avishai Berkowitz, Daniel Shalem, Daphna Lifschitz, Dudu Moshe, Eitan Porat, Eitan Richardson, Guy Shiran, Itay Chachy, Jonathan Chetboun, Michael Finkelson, Michael Kupchick, Nir Zabari, Nitzan Guetta, Noa Kotler, Ofir Bibi, Ori Gordon, Poriya Panet, Roi Benita, Shahar Armon, Victor Kulikov, Yaron Inger, Yonatan Shiftan, Zeev Melumian, and Zeev Farbman. LTX-2: Efficient Joint Audio-Visual Foundation Model. *arXiv preprint arXiv:2601.03233*, 2026. URL https://arxiv.org/abs/2601.03233.

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self Forcing: Bridging the Train-Test Gap in Autoregressive Video Diffusion. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://arxiv.org/abs/2506.08009.

Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, et al. GPT-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

Dengyang Jiang, Xin Jin, Dongyang Liu, Zanyi Wang, Mingzhe Zheng, Ruoyi Du, Xiangpeng Yang, Qilong Wu, Zhen Li, Peng Gao, Harry Yang, and Steven Hoi. D-OPSD: On-Policy Self-Distillation for Continuously Tuning Step-Distilled Diffusion Models. *arXiv preprint arXiv:2605.05204*, 2026. URL https://arxiv.org/abs/2605.05204.

Zeyinzi Jiang, Zhen Han, Chaojie Mao, Jingfeng Zhang, Yulin Pan, and Yu Liu. VACE: All-in-One Video Creation and Editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 17191–17202, October 2025. URL https://arxiv.org/abs/2503.07598.

Simran Kaur, Narutatsu Ri, Yinghui He, Liam Fowl, and Sanjeev Arora. Rethinking On-Policy Self-Distillation for Thinking Models. *arXiv preprint arXiv:2607.05184*, 2026. URL https://arxiv.org/abs/2607.05184.

Minh-Quan Le, Yuanzhi Zhu, Vicky Kalogeiton, and Dimitris Samaras. What about gravity in video generation? Post-Training Newton's Laws with Verifiable Rewards. *arXiv preprint arXiv:2512.00425*, 2025. URL https://arxiv.org/abs/2512.00425.

Chenyu Li, Oscar Michel, Xichen Pan, Sainan Liu, Mike Roberts, and Saining Xie. PISA Experiments: Exploring Physics Post-Training for Video Diffusion Models by Watching Stuff Drop. *arXiv preprint arXiv:2503.09595*, 2025. URL https://arxiv.org/abs/2503.09595.

Haodong Li, Tianfei Ren, Xiaoxiao Ma, Chunmei Qing, Zhen Fang, Sipeng He, Ziyu Guo, Haoyu Wu, Juanxi Tian, Yihang Zou, Ruichuan An, Dongzhi Jiang, Boxue Yang, Ji Xie, Xu Huang, Wenhao Yan, Jialv Zou, Zhengrong Yue, Yaxin Luo, Xiaotong Li, Yuzhu Wang, Junyan Ye, Jinjing Zhao, Zehui Chen, Lin Chen, Renye Yan, Feng Zhao, and Pheng-Ann Heng. VideoCoCo: Code-as-CoT for Physically-Consistent Video Generation via an Agentic Dual-Engine System. *arXiv preprint arXiv:2607.27380*, 2026a. URL https://arxiv.org/abs/2607.27380.

Quanhao Li, Junqiu Yu, Kaixun Jiang, Yujie Wei, Zhen Xing, Pandeng Li, Ruihang Chu, Shiwei Zhang, Yu Liu, and Zuxuan Wu. DiffusionOPD: A Unified Perspective of On-Policy Distillation in Diffusion Models. *arXiv preprint arXiv:2605.15055*, 2026b. URL https://arxiv.org/abs/2605.15055.

Hongyu Liu, Chun Wang, Feng Gao, Xuanhua He, Yue Ma, Ziyu Wan, Yong Zhang, Xiaoming Wei, and Qifeng Chen. OPSD-V: On-Policy Self-Distillation for Post-Training Few-Step Autoregressive Video Generators. *arXiv preprint arXiv:2607.08766*, 2026. URL https://arxiv.org/abs/2607.08766.

David Lopez-Paz, Léon Bottou, Bernhard Schölkopf, and Vladimir Vapnik. Unifying distillation and privileged information. In *International Conference on Learning Representations*, 2016. URL https://arxiv.org/abs/1511.03643.

Calvin Luo, Zilai Zeng, Mingxi Jia, Yilun Du, and Chen Sun. Self-Improving Loops for Visual Robotic Planning. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2506.06658.

Fanqing Meng, Jiaqi Liao, Xinyu Tan, Wenqi Shao, Quanfeng Lu, Kaipeng Zhang, Yu Cheng, Dianqi Li, Yu Qiao, and Ping Luo. Towards World Simulator: Crafting Physical Commonsense-Based Benchmark for Video Generation. *arXiv preprint arXiv:2410.05363*, 2024. URL https://arxiv.org/abs/2410.05363.

Saman Motamed, Laura Culp, Kevin Swersky, Priyank Jaini, and Robert Geirhos. Do generative video models understand physical principles? *arXiv preprint arXiv:2501.09038*, 2025. URL https://arxiv.org/abs/2501.09038.

NeoHorse Team, Guoliang Cao, Guohao Dai, Tianyu Guo, Kai Han, and Hailin Hu. NeoHorse-1: Towards Recursive Self-Improvement via Agentic Post-Training with Routing Harness. *arXiv preprint arXiv:2609.08183*, 2026. URL https://arxiv.org/abs/2609.08183.

Saurabh Pathak, Elahe Arani, Mykola Pechenizkiy, and Bahram Zonooz. PhysVid: Physics Aware Local Conditioning for Generative Video Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026. URL https://arxiv.org/abs/2603.26285.

Emiliano Penaloza, Dheeraj Vattikonda, Nicolas Gontier, Alexandre Lacoste, Laurent Charlin, and Massimo Caccia. Privileged Information Distillation for Language Models. *arXiv preprint arXiv:2602.04942*, 2026. URL https://arxiv.org/abs/2602.04942.

Tim Rädsch, Yuki M Asano, Hilde Kuehne, Stefan Bauer, Priyank Jaini, Robert Geirhos, and Carsten T. Lüth. Physics-IQ Verified. *arXiv preprint arXiv:2606.18943*, 2026. URL https://arxiv.org/abs/2606.18943.

Maxime Robeyns, Martin Szummer, and Laurence Aitchison. A Self-Improving Coding Agent. *arXiv preprint arXiv:2504.15228*, 2025. URL https://arxiv.org/abs/2504.15228.

Stephane Ross, Geoffrey J. Gordon, and J. Andrew Bagnell. A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15, pp. 627–635, 2011. URL https://arxiv.org/abs/1011.0686.

Jürgen Schmidhuber. Gödel Machines: Self-Referential Universal Problem Solvers Making Provably Optimal Self-Improvements. *arXiv preprint cs/0309048*, 2003. URL https://arxiv.org/abs/cs/0309048.

Samyak Shrestha and Alexander Tessier. Rethinking Privileged Information in On-Policy Self-Distillation. *arXiv preprint arXiv:2608.18271*, 2026. URL https://arxiv.org/abs/2608.18271.

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Yarin Gal, Nicolas Papernot, and Ross Anderson. The Curse of Recursion: Training on Generated Data Makes Models Forget. *arXiv preprint arXiv:2305.17493*, 2023. URL https://arxiv.org/abs/2305.17493.

Achint Soni, Sreyas Venkataraman, Abhranil Chandra, Sebastian Fischmeister, Percy Liang, Bo Dai, and Sherry Yang. VideoAgent: Self-Improving Video Generation. *arXiv preprint arXiv:2410.10076*, 2024. URL https://arxiv.org/abs/2410.10076.

Hongyao Tang, Yi Ma, Pengyi Li, and Yifu Yuan. Generalized Agent Iteration: One Formal Framework for Iterative Policy Improvement and Recursive Self-Improvement. *arXiv preprint arXiv:2609.13406*, 2026. URL https://arxiv.org/abs/2609.13406.

Team Seedance. Seedance 2.0: Advancing Video Generation for World Complexity. *arXiv preprint arXiv:2604.14148*, 2026. URL https://arxiv.org/abs/2604.14148.

Team Wan. Wan: Open and Advanced Large-Scale Video Generative Models. *arXiv preprint arXiv:2503.20314*, 2025. URL https://arxiv.org/abs/2503.20314.

Antonios Tragoudaras, Chenyu Zhang, Daniil Cherniavskii, Antonios Vozikis, Thijmen Nijdam, Derck W. E. Prinzhorn, Mark Bodracska, Nicu Sebe, Andrii Zadaianchuk, and Efstratios Gavves. Evaluating Newtonian Mechanics in Video Generative Models with Real Physical Systems. *arXiv preprint arXiv:2504.02918*, 2025. URL https://arxiv.org/abs/2504.02918.

Hanyang Wang, Yimo Cai, Weiliang Chen, Jiawei Chi, Haowen Sun, Qiyu Dai, Yi-Hsin Hung, Xingzhuo Guo, Jinshan Ren, Runmao Yao, Ziwei Liu, Mingsheng Long, Yueqi Duan, Jun Gao, Jiangran Lyu, Fangfu Liu, and Jialong Wu. Code as Worlds: Agentic Discovery of Executable World Representations for Physical Reasoning. *arXiv preprint arXiv:2608.27549*, 2026a. URL https://arxiv.org/abs/2608.27549.

Ziyue Wang, Shiqi Huang, Weiwen Xu, Bihan Wen, and Xudong Jiang. Video-OPSD: Exploiting Privileged Visual Evidence for On-Policy Self-Distillation in Video Large Language Models. *arXiv preprint arXiv:2608.27065*, 2026b. URL https://arxiv.org/abs/2608.27065.

Keming Wu, Yijing Cui, Wenhan Xue, Qijie Wang, Xuan Luo, Zhiyuan Feng, Zuhao Yang, Sudong Wang, Sicong Jiang, Haowei Zhu, Zihan Wang, Ping Nie, Wenhu Chen, and Bin Wang. WorldReasonBench: Human-Aligned Stress Testing of Video Generators as Future World-State Predictors. *arXiv preprint arXiv:2605.10434*, 2026. URL https://arxiv.org/abs/2605.10434.

Wenzhuo Xu, Yuchen Zhu, Chongjian Ge, Xuan Shen, Jing Shi, Jason Kuen, Yongxin Chen, Molei Tao, Christopher McComb, Noelia Grande Gutiérrez, and Jiuxiang Gu. VeriPhy: Agentic Physical Reasoning for World Model Evaluation and Refinement. *arXiv preprint arXiv:2609.03153*, 2026. URL https://arxiv.org/abs/2609.03153.

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 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025. URL https://arxiv.org/abs/2412.00596.

Shaofeng Yin, Jiaxin Ge, Zora Zhiruo Wang, Chenyang Wang, Xiuyu Li, Michael J. Black, Trevor Darrell, Angjoo Kanazawa, and Haiwen Feng. Vision-as-Inverse-Graphics Agent via Interleaved Multimodal Reasoning. *arXiv preprint arXiv:2601.11109*, 2026. URL https://arxiv.org/abs/2601.11109.

Xunjian Yin, Xinyi Wang, Liangming Pan, Li Lin, Xiaojun Wan, and William Yang Wang. Gödel Agent: A Self-Referential Agent Framework for Recursive Self-Improvement. *arXiv preprint arXiv:2410.04444*, 2024. URL https://arxiv.org/abs/2410.04444.

Jianhao Yuan, Xiaofeng Zhang, Felix Friedrich, Nicolas Beltran-Velez, Melissa Hall, Reyhane Askari-Hemmat, Xiaochuang Han, Nicolas Ballas, Michal Drozdzal, and Adriana Romero-Soriano. Inference-time Physics Alignment of Video Generative Models with Latent World Models. *arXiv preprint arXiv:2601.10553*, 2026. URL https://arxiv.org/abs/2601.10553.

Eliezer Yudkowsky. Artificial intelligence as a positive and negative factor in global risk. In Nick Bostrom and Milan M. Cirkovic (eds.), *Global Catastrophic Risks*, pp. 308–345. Oxford University Press, Oxford, 2008.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. STaR: Bootstrapping Reasoning With Reasoning. *arXiv preprint arXiv:2203.14465*, 2022. URL https://arxiv.org/abs/2203.14465.

Eric Zelikman, Eliana Lorch, Lester Mackey, and Adam Tauman Kalai. Self-Taught Optimizer (STOP): Recursively Self-Improving Code Generation. *arXiv preprint arXiv:2310.02304*, 2023. URL https://arxiv.org/abs/2310.02304.

Jenny Zhang, Shengran Hu, Cong Lu, Robert Lange, and Jeff Clune. Darwin Gödel Machine: Open-Ended Evolution of Self-Improving Agents. *arXiv preprint arXiv:2505.22954*, 2025a. URL https://arxiv.org/abs/2505.22954.

Xiangdong Zhang, Jiaqi Liao, Shaofeng Zhang, Fanqing Meng, Xiangpeng Wan, Junchi Yan, and Yu Cheng. VideoREPA: Learning Physics for Video Generation through Relational Alignment with Foundation Models. *arXiv preprint arXiv:2505.23656*, 2025b. URL https://arxiv.org/abs/2505.23656.

Yongheng Zhang, Guang Yang, Ruihan Hou, Qiguang Chen, Ziang Liu, Xiaolong Liu, Manman Zhang, Yanchao Hao, Zheng Wei, Hao Wu, Libo Qin, Peishan Dai, Yinghui Li, Di Yin, and Xing Sun. Thinking in Video: Can Video Generators Really Reason About the Real World? *arXiv preprint arXiv:2607.17523*, 2026. URL https://arxiv.org/abs/2607.17523.

Liangliang Zhao, Junying Wang, Danni Yang, Yifan Chang, Bin Fu, Yu Qiao, Bowen Zhou, and Yihao Liu. Distilling Physical Priors into Streaming World Models. *arXiv preprint arXiv:2608.07981*, 2026a. URL https://arxiv.org/abs/2608.07981.

Rui Zhao, Kaiming Yang, Jifeng Zhu, Siyang Chen, Ziqi Wang, Weijia Wu, Kevin Qinghong Lin, Heng Wang, and Mike Zheng Shou. Dream.exe: Can Video Generation Models Dream Executable Robot Manipulation? *arXiv preprint arXiv:2606.04811*, 2026b. URL https://arxiv.org/abs/2606.04811.

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-Distilled Reasoner: On-Policy Self-Distillation for Large Language Models. *arXiv preprint arXiv:2601.18734*, 2026c. URL https://arxiv.org/abs/2601.18734.

Dian Zheng, Ziqi Huang, Hongbo Liu, Kai Zou, Yinan He, Fan Zhang, Lulu Gu, Yuanhan Zhang, Jingwen He, Wei-Shi Zheng, Yu Qiao, and Ziwei Liu. VBench-2.0: Advancing Video Generation Benchmark Suite for Intrinsic Faithfulness. *arXiv preprint arXiv:2503.21755*, 2025. URL https://arxiv.org/abs/2503.21755v2.

Tong Zheng, Xidong Wu, Zheng Zhang, Zhankui He, Chaoyi Zhang, Benjamin Coleman, Junchen Jiang, Pan Zhou, Hao Su, and Carl Yang. Dream-RSI: Recursive Self-Improvement through Evolving Worlds. *arXiv preprint arXiv:2609.14858*, 2026. URL https://arxiv.org/abs/2609.14858.

Sihan Zhuang, Xinyuan Chen, Tianfan Xue, and Yaohui Wang. CausalMotion: Structured Physical Reasoning as Keyframe and Trajectory Guidance for Training-Free Video Generation. *arXiv preprint arXiv:2606.14317*, 2026. URL https://arxiv.org/abs/2606.14317.

A More Related Works

This section extends Section 2. Table 4 compares Gen-RSI with its closest methods. Besides the correction signal and the objective, it asks whether a method diagnoses the violated constraint, derives the corrected motion from the model's own samples, supervises at states the current model visits, refreshes its supervision every round, and needs only the generator at inference.

Table 4: **Comparison with related methods.** To our knowledge, only Gen-RSI has all five properties. ◯: diagnosis only against ground truth simulated before training. †: human-motion generation.

Method	Correction signal	Objective	Diagnosis	Motion from own samples	Fresh states	Per-round refresh	Gen. only
PISA (Li et al., 2025)	simulated videos	SFT + reward	◯	–	✓	–	✓
VideoREPA (Zhang et al., 2025b)	VideoMAEv2	relation distillation	–	–	–	–	✓
PhyGDPO (Cai et al., 2025)	VLM	DPO	–	–	–	–	✓
NewtonRewards (Le et al., 2025)	optical flow	verifiable reward	✓	–	✓	–	✓
PhyT2V (Xue et al., 2025)	LLM	prompt rewriting	–	–	–	–	–
VideoCoCo (Li et al., 2026a)	Blender	draft editing	–	–	–	–	–
Self Forcing (Huang et al., 2025)	bidirectional teacher	DMD	–	–	✓	–	✓
Self-correcting loops† (Gillman et al., 2024)	physics tracker	retraining	–	✓	–	✓	✓
★ **Gen-RSI (ours)**	Blender	OPD	✓	✓	✓	✓	✓

Physics evaluation. Benchmarks show that realistic appearance does not imply correct dynamics (Meng et al., 2024; Zheng et al., 2025; Motamed et al., 2025; Tragoudaras et al., 2025). Others test future-state prediction (Wu et al., 2026; Zhang et al., 2026), controlled interventions (Cai et al., 2026) or whether generated motion can be executed by a robot (Zhao et al., 2026b), and audits revisit their protocols (Rädsch et al., 2026). These works measure errors but do not produce correction targets. Many also rely on vision-language judges, whose physical understanding is limited (Chow et al., 2025). Gen-RSI uses such benchmarks only for evaluation and computes its training signal in a simulator.

Training-time and inference-time physics. Most training-time methods fix their supervision before training, through simulated videos (Li et al., 2025), feature alignment (Zhang et al., 2025b), teacher priors (Zhao et al., 2026a) or local conditioning (Pathak et al., 2026). PISA's diagnosis compares generations only with scenes simulated in advance (◯ in Table 4). PhyGDPO and NewtonRewards score the model's own samples (Cai et al., 2025; Le et al., 2025), but a score says that a motion is wrong, not what the correct motion is. Inference-time methods leave the generator unchanged. They rewrite prompts (Xue et al., 2025), plan keyframes and trajectories (Zhuang et al., 2026), steer sampling with a world-model reward (Yuan et al., 2026), or keep an executable world outside the generator (Li et al., 2026a; Chen et al., 2026b). They pay this cost at every generation, so PhyT2V and VideoCoCo fail the last column of the table. Gen-RSI computes the correct motion in Blender at training time and distills it, so only the generator runs at inference.

Executable worlds. Programs give agents an explicit world state to inspect and test (Gkountouras et al., 2025). VIGA rebuilds an image as a Blender program (Yin et al., 2026). Code as Worlds turns text or video into simulation programs that train vision-language models (Wang et al., 2026a), and VeriPhy evaluates generated videos with expert tools (Xu et al., 2026). Gen-RSI extends the VIGA loop from images to videos. Unlike these works, it uses the program to compute a corrected motion for the generator's own sample and distills that motion back into the generator.

On-policy and privileged distillation. On-policy distillation supervises a student at states it visits itself (Ross et al., 2011; Agarwal et al., 2024). Self Forcing uses this idea to close the train-test gap of autoregressive video diffusion (Huang et al., 2025), and DiffusionOPD and Flow-OPD extend it to diffusion and flow-matching models (Li et al., 2026b; Fang et al., 2026b). Privileged distillation gives the teacher information that the student lacks (Lopez-Paz et al., 2016; Penaloza et al., 2026; Fang et al., 2026a). OPSD builds this teacher from the student itself, first in language models (Zhao et al., 2026c) and then in diffusion and video generators (Jiang et al., 2026; Liu et al., 2026) and video language models (Wang et al., 2026b). Such context does not always help (Kaur et al., 2026; Shrestha & Tessier, 2026). In Gen-RSI, the privileged information is a verified simulator repair of

the student's own sample, and the teacher is a frozen VACE branch (Jiang et al., 2025) rather than an EMA copy of the student.

Self-training loops. Self-training fine-tunes a model on its own filtered outputs (Anthony et al., 2017; Zelikman et al., 2022; Gulcehre et al., 2023). Without such filtering, generative models trained on their own samples lose the tails of the distribution or degrade in quality and diversity (Shumailov et al., 2023; Alemohammad et al., 2023), unless real data keep a large share of training (Bertrand et al., 2023). Gillman et al. correct generated human motion with a physics tracker before retraining (Gillman et al., 2024), the closest row in Table 4. Video self-improvement loops for embodied planning refine video models with feedback on their own plans (Soni et al., 2024; Luo et al., 2026). In all these loops, the corrector stays fixed. Gen-RSI never trains on raw generations, and it revises its correction procedures across rounds.

Recursive self-improvement. RSI asks a system to improve the process that improves it (Good, 1965; Yudkowsky, 2008). The Gödel machine accepts a self-rewrite only after proving it useful (Schmidhuber, 2003). Later systems replace proof with empirical validation. They improve scaffolds (Zelikman et al., 2023), agent code (Yin et al., 2024; Robeyns et al., 2025; Zhang et al., 2025a), models together with their harnesses (Fan & Huang, 2026; NeoHorse Team et al., 2026), or the worlds in which agents learn (Zheng et al., 2026). Surveys and frameworks organize this field (Chen et al., 2026a; Tang et al., 2026; Duan et al., 2026). Nearly all of this work improves language or coding agents. Gen-RSI brings RSI to video post-training. The generator improves through distillation, the agent improves its tools rather than its weights, and the validation criteria stay outside both (Section 3.4).

B Method Details

This section lists the implementation rules behind Section 3. Algorithm 1 gives the full training loop, and the paragraphs below follow the order of the main text.

Reconstruction. The operator $\mathcal{A}$ keeps an evolving multimodal memory across its edit steps, as in VIGA (Yin et al., 2026). VIGA fits a single image. $\mathcal{A}$ fits a video, so it also recovers a per-frame state trace and the events that connect frames. The proxy is executable: running b in Blender reproduces τ. The same program later renders the frames and masks used by the teacher. P_r^{obs} is never required to satisfy the constraints tested next, since an automatically corrected reconstruction would hide the failure. Reconstruction may consult the agent's archive of earlier proxies and rejections. No archived verdict is adopted without the checks below.

Diagnosis evidence. Each violation also records the observation evidence behind it. Typical violations are overlap depth at contact, an identity missing after a verified reappearance opportunity, a latch that changes without a release event, and an unchanged trajectory under a support-removal pair whose effect simulation has established. Two abstention rules apply. A hidden object alone does not establish disappearance. An unchanged intervention outcome is an error only when the tested cause should matter. The fit-quality gate must pass before any verdict is assigned to the source video, because successful Blender execution is not evidence of a correct reconstruction.

Repair and verification. The repair constraints in Section 3.2 describe a search, not an exact solver. A repair leaves every keyframe outside the diagnosed segment untouched. The verifier checks both the targeted constraints and the unchanged factors. An accepted record stores both proxies, their code diff, the evidence and the rendered frames. Repair may reuse previously verified operators, but reuse never replaces re-execution. Finite checks certify only the declared tests.

Teacher interface. The teacher is frozen in all rounds, including its context adapter. Its backbone is the frozen Wan2.1-T2V network of the same scale as the student (Jiang et al., 2025); only the context branch differs. Unlike OPSD with an EMA teacher (Jiang et al., 2026; Liu et al., 2026), the teacher does not track the student. Student and teacher must share compatible latent coordinates, noise schedules, frame indexing and output parameterization. Any conversion is fixed before training. The Wan/VACE pairing is accepted only after this interface test passes. The design does not

Algorithm 1 Gen-RSI training. Discovery, repair and machinery revision run once per round. Distillation resamples student states at every update. Final-evaluation prompts never enter the loop.

Require: student θ_0; frozen VACE teacher T_ϕ; machinery $\mathcal{M}_0$; training prompts $\mathcal{C}$; reference set ν_0; rounds R; updates per round U; step size η
Ensure: student θ_R, deployed alone

$\mathcal{B}_0 \leftarrow \emptyset$
for $r = 0, \dots, R-1$ **do**
  Round boundary: current errors become world proxies, using $\mathcal{M}_r$ (Sec. 3.2)
  $\mathcal{Q}_r \leftarrow \emptyset$
  for each discovery prompt $c \in \mathcal{C}$ **do**
    $x \sim \pi_{\theta_r}(\cdot \mid c)$; $P^{\text{obs}} \leftarrow \mathcal{A}(c, x)$ ▷ keep the error, Sec. 3.2
    $(\mathcal{F}, \mathcal{U}) \leftarrow D(P^{\text{obs}})$ ▷ log abstentions
    if $\mathcal{F} \neq \emptyset$ **then**
      $P^+ \leftarrow$ constrained repair of P^{obs} ▷ Sec. 3.2
      if the verifier re-executes P^+ and accepts **then**
        $\mathcal{Q}_r \leftarrow \mathcal{Q}_r \cup \{(c, P^{\text{obs}}, P^+)\}$
      else
        log the rejection
      end if
    end if
  end for
  $\mathcal{B}_{r+1} \leftarrow \mathcal{Q}_r \cup \text{Replay}_\rho(\mathcal{B}_r)$ ▷ Sec. 3.4
  $\mathcal{M}_{r+1} \leftarrow \text{Meta}(\mathcal{M}_r, \mathcal{E}_r)$ ▷ $\mathcal{E}_r$: accepts, rejections, abstentions; Sec. 3.4
  Local OPD on fresh student states (Sec. 3.3)
  $\theta_{r,0} \leftarrow \theta_r$
  for $j = 0, \dots, U-1$ **do**
    draw $\{(c_i, P_i^+)\}_i$ from $\mathcal{B}_{r+1}$; $u_i \leftarrow u(P_i^+)$ ▷ rendered condition only
    $z_{i,T:0} \leftarrow$ run $\pi_{\theta_{r,j}}(\cdot \mid c_i)$ from fresh noise ▷ latents are never cached
    $g_i \leftarrow$ correspondence gate of z_i against P_i^+
    query $T_\phi(z_{i,t}, c_i, t, u_i)$ at the same states, $t \in \mathcal{T}_i$ ▷ frozen, detached
    $\theta_{r,j+1} \leftarrow \theta_{r,j} - \eta \nabla_\theta \mathcal{L}_{r,j}(\theta_{r,j})$ ▷ Eq. 3; anchor only if $N_g = 0$
  end for
  $\theta_{r+1} \leftarrow \theta_{r,U}$
end for
return θ_R

assume a same-capacity or EMA teacher. VACE carries additional conditioning components, and we report their cost and capacity.

Gate and update. The student output is decoded for correspondence checking. The gate also requires the earlier fit and repair checks to pass. Acceptance and abstention are recorded separately. Eq. 3 is a semi-gradient objective on freshly collected states. It is not the full derivative of an expectation through the state distribution induced by θ. GKD, DiffusionOPD and diffusion OPSD use the same stopped-rollout convention (Agarwal et al., 2024; Li et al., 2026b; Jiang et al., 2026). Flow-OPD instead derives its velocity regression from a per-step KL between Gaussian transitions (Fang et al., 2026b); we do not rely on that derivation, and ω is a design choice rather than a KL weight. Acceptance frequency is part of the cost and coverage report.

Round transition. The boundary that refreshes the bank also revises the machinery as described in Section 3.4, so the data and the mechanism that builds it change together. An empty bank produces anchor-only updates on the reference set, and zero accepted corrections are recorded. The procedure has an explicit refresh interval, not continuous adaptation. New records replace exhausted or incompatible ones at the next boundary. A rejection is never counted as a successful repair.

Budgets and stopping. All variants predeclare discovery attempts, local updates, repair retries and a stopping rule on a development split disjoint from final evaluation. No-progress rounds, failed programs, rejected candidates and teacher queries all count toward the resource ledger. The default study stops at a fixed round budget and reports every checkpoint. An optional development-based stopping policy is evaluated separately. Unchanged inference inputs do not prove unchanged runtime or visual quality, so we measure both.

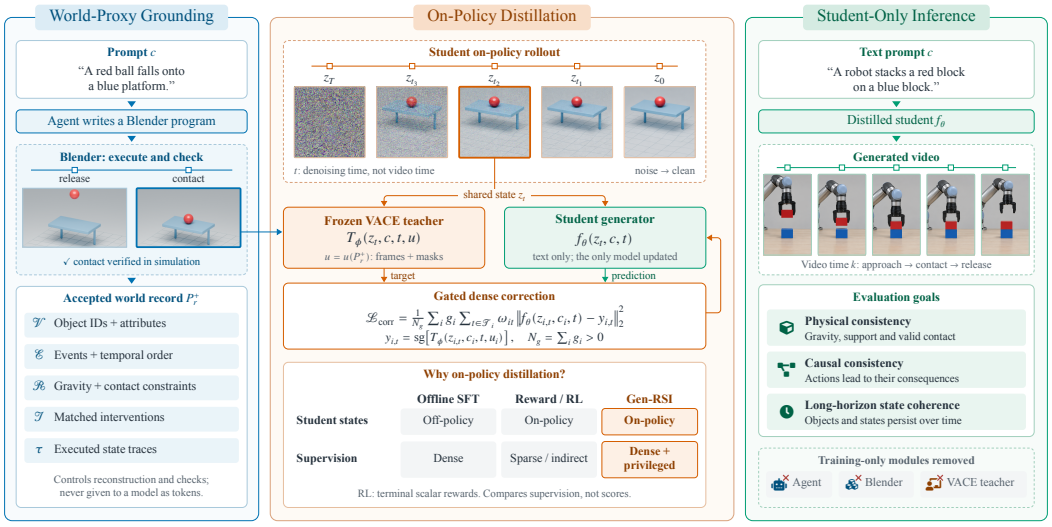


Figure 6: **Local OPD inside the recursive framework. World-Proxy Grounding:** an agent writes a Blender program, and execution checks the event to yield an accepted record P_r^+; only its rendered frames and masks $u(P_r^+)$ reach the frozen VACE teacher. **On-Policy Distillation:** teacher and student are queried at the same fresh student state z_t (t is denoising time), and detached targets correct only the student through the gated term of Eq. 3. **Student-Only Inference:** generation uses only text and the student; the robot strip advances video time k. Blender frames are simulation renders; denoising and robot frames are illustrations.

Meta operator. The inputs of Meta are outcomes the loop itself produced. The revision is therefore internal to the coupled system, not an external retraining of the agent. Each candidate criterion is stored as an executable check together with the evidence it requires. Candidates that fail validation are recorded and rejected, not silently dropped. Every revision is versioned in the run ledger, including revisions that later prove harmful.

Competence estimation. The proxy-development split is disjoint from final evaluation and from the failure families used for training. The frozen-machinery control keeps the data-refresh schedule and removes only the machinery revision of Section 3.4. It tests whether revision contributes. The self-validating control tests whether the external anchor is necessary.

C Human Study

C.1 Protocol

The automatic scores in Section 4.2 are produced by a multimodal judge, so they inherit whatever that judge is insensitive to. To check the main claim against human perception we ran a rating study over the same prompts, using the five dimensions that the world-proxy checker itself scores: *visible event*, *process*, *causality*, *continuity*, and *physical plausibility*. Re-using the checker's own rubric keeps the machine and human measurements commensurable — both describe the same failure modes, so a disagreement between them is informative rather than a difference of vocabulary.

Interface. Raters review one clip per page in the interface of Figure 7. Each page shows a single generated video, the text prompt it was generated from, and one 0–5 scale per dimension, where 0 is very poor and 5 is excellent. The prompt is shown because three of the five dimensions — visible event, process, and causality — are defined relative to the event the prompt asks for and cannot be judged from the video alone. Clips are grouped by method so that a rater holds one generator's behaviour in mind at a time. A blind mode hides the method name and permutes the presentation order with a fixed seed; the seed is fixed so that a rater who reloads mid-session resumes the same sequence instead of re-rating one clip and skipping another.

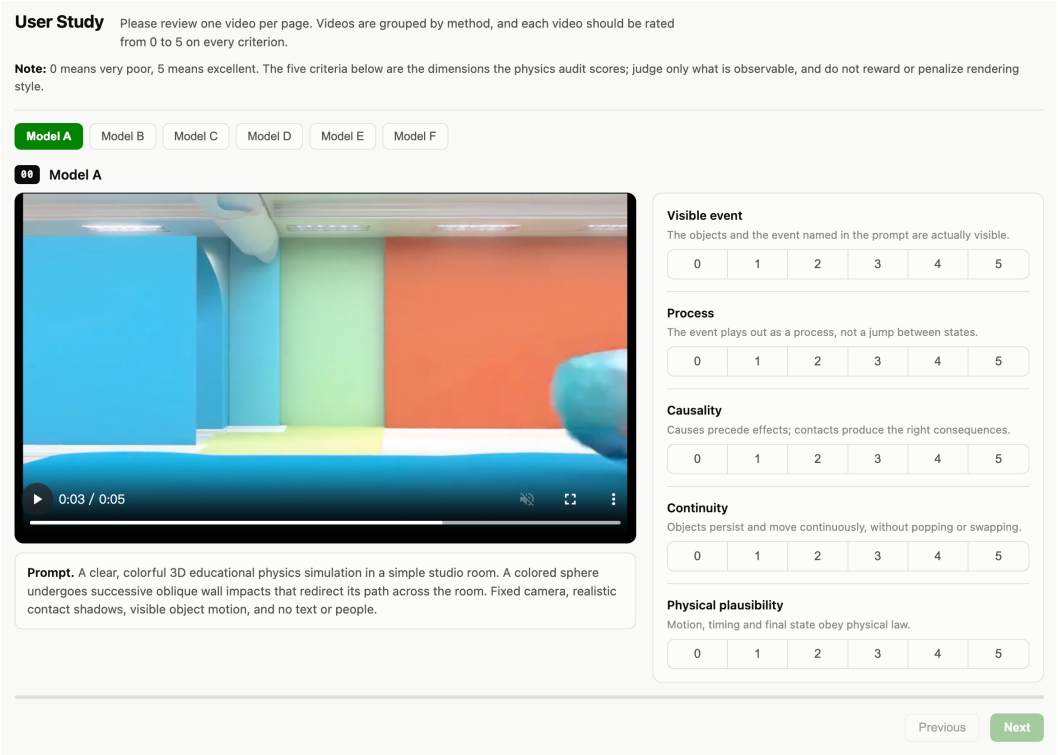


Figure 7: **The rating interface.** One clip per page with its prompt, and a 0–5 scale for each of the five checker dimensions. Methods are selected by the tabs above; a blind mode hides the method name and permutes the order.

Method	Visible	Process	Causality	Continuity	Physical	Average
Base generative model	2.68	2.31	2.15	2.44	1.97	2.31
+ PhysVid (Pathak et al., 2026)	2.60	2.24	2.08	2.39	1.89	2.24
+ PhyGDPO (Cai et al., 2025)	2.85	2.57	2.49	2.71	2.48	2.62
+ PhyT2V (Xue et al., 2025)	3.16	3.02	2.95	3.10	3.02	3.05
+ OPD	2.94	2.70	2.58	2.86	2.47	2.71
+ Gen-RSI (ours)	**3.92**	**3.85**	**3.90**	**3.78**	**3.90**	**3.87**

Table 5: **Human ratings on the five checker dimensions.** Mean rating on a 0–5 scale (higher is better); all methods use the Wan2.1-1.3B backbone and the same prompts. **Blue** and light blue mark the best and second-best result per column.

Instructions. Raters are told to judge observable evidence only, and explicitly not to reward or penalise rendering style, colour choice, or resolution. This matters here: the methods differ in visual polish as well as in physics, and without the instruction a rater tends to score the better-looking clip higher on dimensions that have nothing to do with appearance.

C.2 Results

Table 5 reports the mean rating per dimension. The ordering reproduces Table 1: Gen-RSI is best on every dimension, PhyT2V is the strongest baseline, and PhysVid sits marginally below the base generator. That last point is the one worth stating plainly — local physics conditioning does not, on its own, buy a rating a human will confirm, and the automatic and human protocols agree about it.

The gap between Gen-RSI and the strongest baseline is widest on *causality* and *physical plausibility*, and narrowest on *visible event*. This is the shape the method predicts. Whether the named objects

appear at all is something a text-conditioned generator already does well, so there is little headroom on that dimension; whether contacts produce the right consequences is exactly what the simulator checks and the proxy corrects, so that is where supervision from a verified world proxy should show up. A method that merely improved appearance would lift the first dimension and leave the last two flat.

D LOOP ROUNDS

This appendix shows what a repair round does. Section D.1 follows four cases across rounds, Section D.2 gives the revision request a round issues — what the agent is shown, what it may change, and what it must leave alone — and Section D.3 places the final round beside the baselines.

D.1 ROUND-BY-ROUND VISUALISATION

Figure 8 follows four prompts through the loop. Reading along a row shows whether the event happens at all; reading down a block shows what each repair round changed. The four are chosen because they fail in different ways.

Damped pendulum — a missing constraint. The prompt-only row is the same picture six times: the brush hangs with no pivot, so nothing swings. Round 0 attaches it and the brush moves, but it hangs from a rod that swings through an arc no pivot would allow. By round 2 the row sweeps a full arc and the amplitude falls across the six frames, which is the damping the prompt asks for. A single frame per round would show a brush in a slightly different place each time and none of this.

Refraction — an effect that is never rendered. The prompt-only row is a flat two-dimensional fill that does not change. Round 0 builds the tank and the tap and starts filling it. Round 1 fills further. Only in round 2 does the submerged part of the strip sit visibly offset from the part above the water line, which is the optical effect the prompt is about. The geometry was right from round 0; the later rounds are spent making the consequence visible.

Wall bounce — objects that do not persist. The prompt-only row has three balls that drift without contacting anything. The round-0 row starts with three and is down to one by the third frame: the repair recovered the room and the camera but lost objects partway through the clip. Rounds 1 and 2 keep all three for the whole clip and separate them by the wall contact. This failure is invisible in a single frame, which is why it survived into round 0 in the first place.

Center-of-mass toppling — correct event, wrong timing. The prompt-only row shows a ramp and a pipe that never interact. In round 0 the ball rolls the length of the plank and the pillar stands untouched through all six frames — the event the prompt names does not occur. In round 1 the pillar is already down in the first frame, so it topples before the ball could have reached it: the object now responds, but not to anything. Round 2 has the pillar upright while the ball travels and falling only after it passes. The three rows are the same scene with the contact absent, mistimed, and then correct.

D.2 THE PER-ROUND REVISION REQUEST

A round boundary issues exactly one request. The agent is shown three things — the prompt the case was authored from, a block of measurements comparing this round's render against the reference clip, and the complete Blender program that produced that render — and returns one thing: the complete revised program. Nothing else crosses the boundary. The instruction is reproduced verbatim below, reflowed to the page width.

```
You are the dynamic-scene VIGA revision agent. Revise the complete
Blender Python program below. The candidate is compared to a source
video, not merely a still image. Preserve camera/appearance unless
metrics require it. Make concrete changes to animation/physics:
velocity, initial velocity, gravity, friction, restitution,
collision bodies, keyframe timing, simulation step. Ensure
```

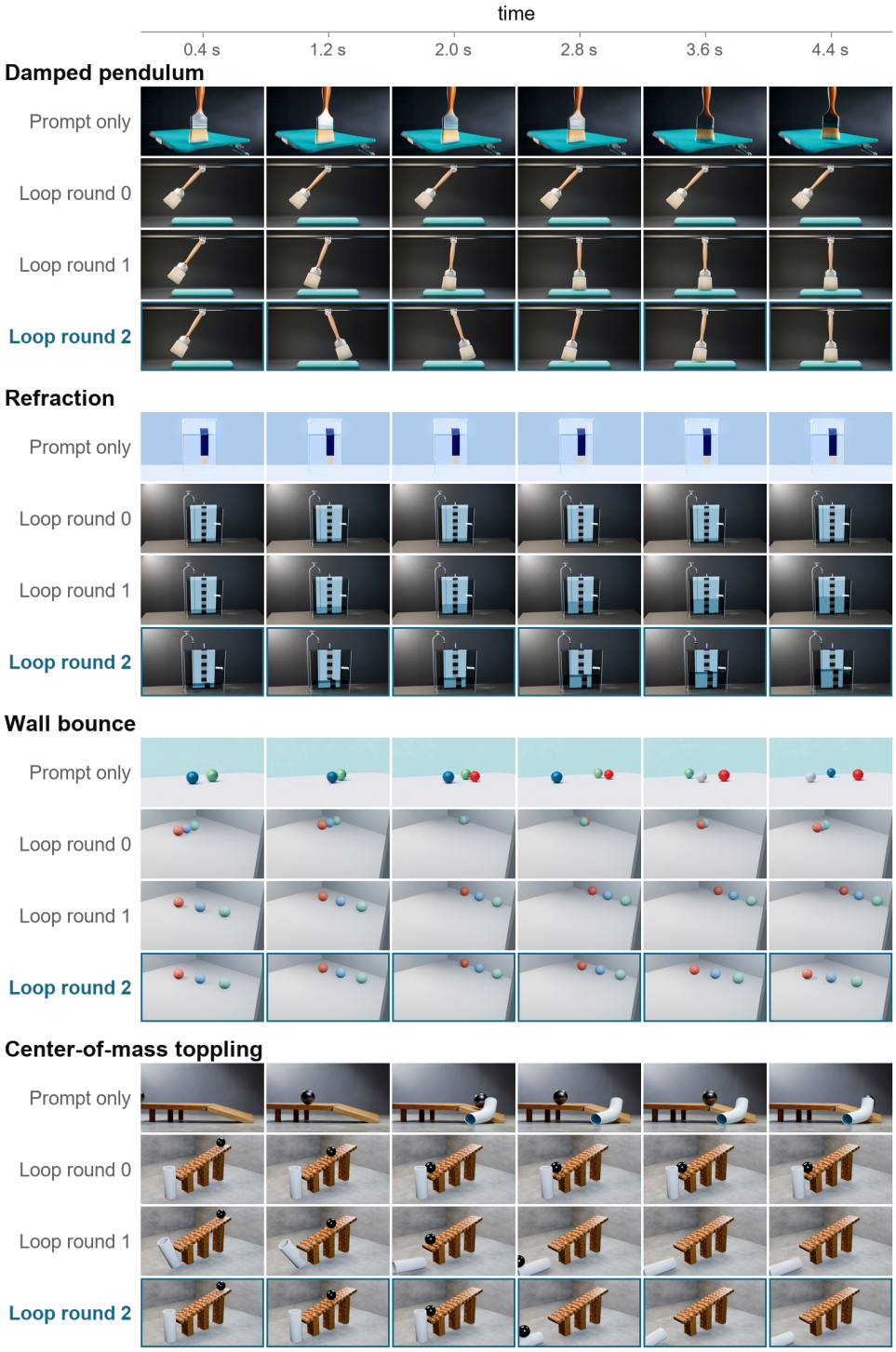


Figure 8: **Four prompts across repair rounds.** Each block is one prompt, and each row is one round: *prompt only*, then loop rounds 0, 1 and 2. Columns are fixed instants of the five-second clip (scale above), so rows can be compared column by column. The outlined last row is the render used in the rest of the paper.

```
frame_start=1, frame_end=80 and actual state changes across frames.
Return ONLY the complete Python source, no markdown.

DESCRIPTION:
  <the prompt this case was authored from>

METRICS:
  {"first_frame_error": ..., "last_frame_error": ...,
   "frame_diff_abs_error": ..., "flow_abs_error": ...,
   "energy_abs_error": ...}

CODE:
  <the complete program that produced this round's render>
```

What the measurements are. Both clips are decoded to at most eighty frames at 256×144 and reduced to a few aggregates; the block handed to the agent is the absolute gap between the reference and the candidate on each.

Field	Quantity
`first_frame_error`	grayscale difference between the two first frames
`last_frame_error`	grayscale difference between the two last frames
`frame_diff_abs_error`	gap in mean inter-frame difference
`flow_abs_error`	gap in mean optical-flow magnitude
`energy_abs_error`	gap in mean inter-frame energy

What the request fixes. Three constraints make the revision checkable rather than open-ended. The editable set is enumerated in the instruction — velocity, initial velocity, gravity, friction, restitution, collision bodies, keyframe timing, simulation step — and camera and appearance are excluded from it unless a measurement forces the change, so a round cannot improve its numbers by relighting or re-framing the shot. The timing contract is fixed at frames 1 to 80, and the render is rejected outright unless exactly eighty frames are produced, which stops a revision from shortening the event to reduce the discrepancy. Finally the reply must be a complete program containing Blender API calls; a fragment, a diff or prose is rejected before it is executed. A revision that survives all three is rendered and measured, and becomes the next round's input.

What the agent is not given. The quantities the instruction asks for are physical, but the evidence it is shown is not: the five fields above are image statistics, and no threshold accompanies them. The agent is told that the render differs from the reference and by how much on each aggregate, and is asked to change physics in response; it is not told which constraint was violated, because at this stage nothing has computed one. This bounds what the loop can repair — a failure that leaves the aggregates unchanged is invisible to it — and it is the reason the verified corrections of Section 3.2 are produced by the simulator rather than by this request.

D.3 Qualitative results

Figure 9 places the final round beside the baselines on matched prompts. Two failure modes recur across the baseline columns. The first is substitution: the generator produces a well-rendered scene that is not the one the prompt describes — a cup of sweetcorn in place of the damped pendulum, a sphere resting against a pipe in place of the toppling block. The second is an event without its consequence: the broom in the combustibility row catches light in three of the four columns, but only the last shows the flame developing and consuming the bristles.

Both modes are invisible to a metric that scores appearance, and both are what the world proxy is there to catch: the first fails the proxy's object and event checks before any render happens, and the second fails its state-change check.

Figure 10 shows the same comparison over time and extends Figure 5 with two mechanics prompts. Each row is one method at three time points, so the failures that a single frame cannot show become visible: objects that change across frames, stay suspended, or interpenetrate on contact, and material changes that never start or complete abruptly. Section 4.3 discusses these failure modes.

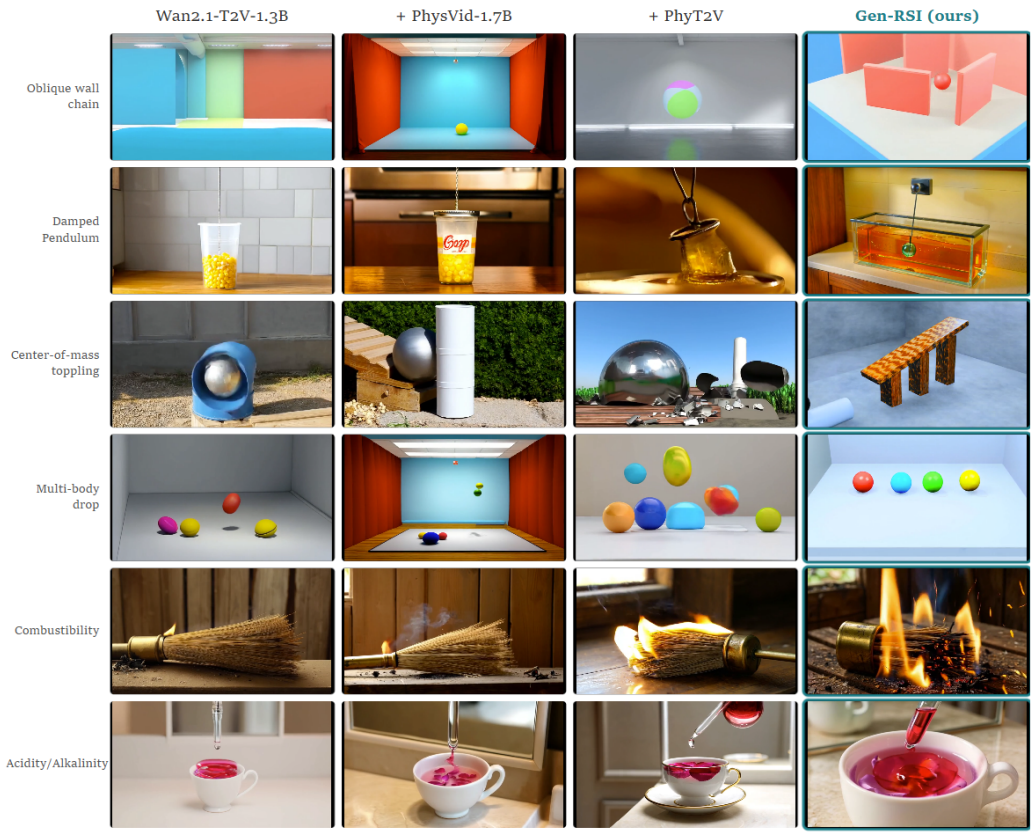


Figure 9: **Final round against the baselines.** One prompt per row, matched frames. The baselines either render a plausible scene without the event (rows 1–4) or render the event without its physical consequence (rows 5–6). Gen-RSI, conditioned on a proxy the simulator accepted, shows both.

E Diagnostic Examples and Evidence Requirements

The four examples in Figure 11 are generated conceptual illustrations, not Blender measurements or student outputs. Actual study records must pair source video evidence with recovered states and reproducible checks. Contact uses signed separation or overlap under a declared geometry tolerance. Permanence requires identity and visibility evidence before and after an occlusion interval, not merely a hidden frame. A latch test records joint and latch states, release events, and persistence over a specified interval. A matched intervention holds camera, materials, initial state, and random factors fixed while changing one causal variable; the expected trajectory difference is validated under the declared physical assumptions. The fourth family is a paired protocol applicable to the other three, not an independent physical law.

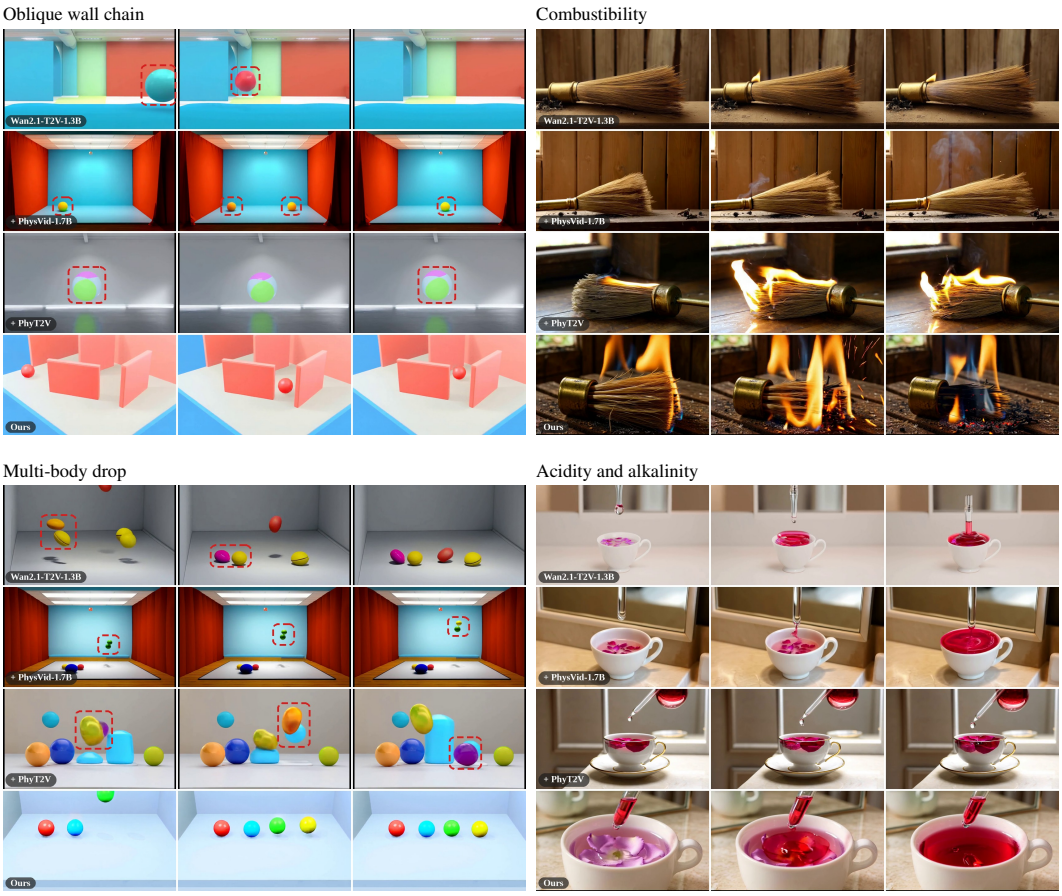


Figure 10: **Qualitative comparison with physics-targeted baselines.** All methods use Wan2.1-T2V-1.3B; each row shows one method at the same three time points. Red dashed boxes mark baseline failures in the mechanics cases.

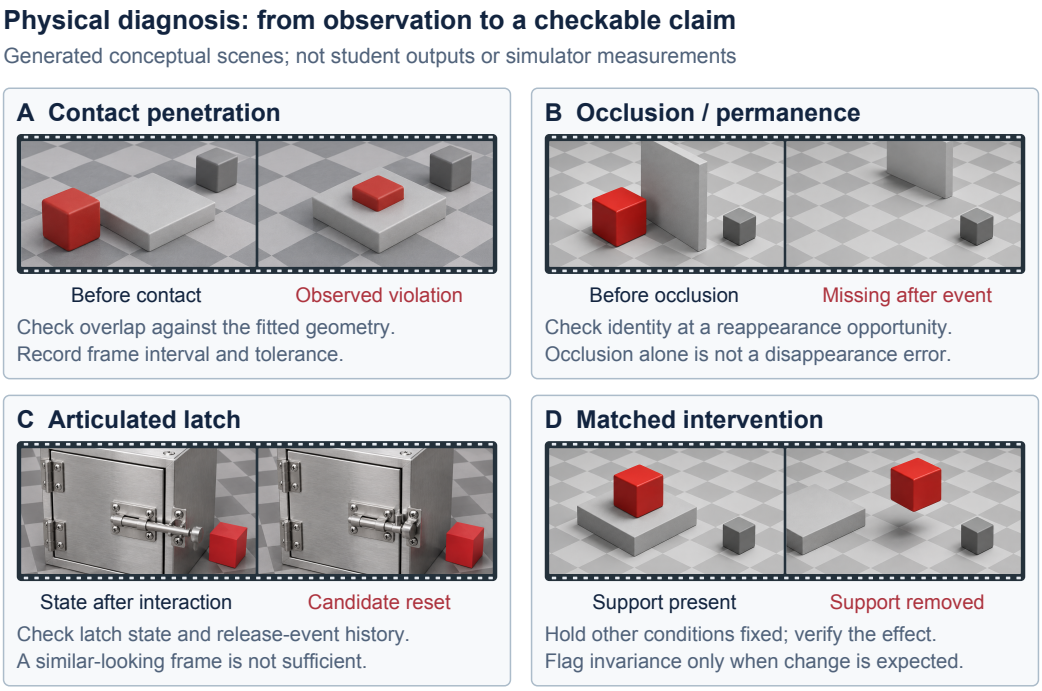


Figure 11: **Four diagnostic families and their evidence requirements.** An observed proxy retains the candidate failure; before repair, an independent check must establish correspondence, the relevant constraint, and its scope. Panels are conceptual illustrations, not measured trajectories.