MIRABENCH: EVALUATING ACTION-CONDITIONED RELIABILITY IN ROBOTIC WORLD MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Action-conditioned world models are increasingly used as scalable simulators for robot learning, yet current evaluations provide limited evidence that their predictions are reliable under the actions they condition on. Existing benchmarks largely emphasize visual fidelity, leaving unclear whether predicted futures are physically plausible, faithful to commanded actions, and calibrated to failure when actions should not succeed. We introduce MIRABENCH, a benchmark that defines *action-conditioned reliability* as a core evaluation target for robotic world models. MiraBench evaluates three complementary dimensions: *Physics Adherence*, which evaluates reference-free physical consistency; *Action-Following Fidelity*, which measures whether predictions respect task-relevant action inputs; and *Optimism Bias Detection*, which probes the tendency to predict successful outcomes under failure-inducing actions. To support this evaluation, we curate a human-annotated corpus with over 16,000 judgments across tasks, failure categories, and leading world models. We evaluate 12 representative model configurations spanning vector-conditioned robotic world models, text-conditioned generative world models, open-weight systems, closed-source systems, and multiple model scales. Across this broad model landscape, MiraBench reveals three central findings: visual fidelity is a poor proxy for action fidelity; increasing model scale does not reliably improve action following; and optimism bias is pervasive across current systems. By shifting evaluation from appearance to action-conditioned reliability, MiraBench provides a diagnostic foundation for assessing and improving robotic world models as faithful simulators.

1 INTRODUCTION

Scalable interaction data remains a central bottleneck for embodied intelligence. Real-world robot data are expensive and slow to collect, difficult to diversify across environments, and particularly sparse for rare failures and safety-critical interactions (Khazatsky et al., 2025; Collaboration et al., 2025). As robots expand into manufacturing, healthcare, and everyday assistance (Brohan et al., 2023; Team et al., 2024; Collaboration et al., 2025), learned world models offer an attractive way to generate interaction trajectories beyond what can be practically collected in the physical world. Recent action-conditioned world models have shown rapid progress in robotic manipulation (Gao et al., 2026; NVIDIA et al., 2025a; Zhu et al., 2025a; Guo et al., 2026) and related domains such as autonomous driving (Hu et al., 2023; Gao et al., 2024). Their value as simulators ultimately depends on whether predicted futures remain faithful to the physical world and to the produced actions.

Existing benchmarks largely emphasize visual quality, perceptual realism, and generic video-generation fidelity (Shang et al., 2026; Li et al., 2025; Qin et al., 2024; Huang et al., 2023). These criteria provide limited evidence about action-conditioned behavior. A rollout can look plausible while violating physical constraints, incompletely executing the commanded motion, or producing nearly the same outcome after a consequential action change. The last case is especially important for robotics. Robot datasets are dominated by successful demonstrations (Mandlekar et al., 2021; Walke et al., 2024; Collaboration et al., 2025), while failures are relatively sparse or filtered during data collection. This imbalance can induce a strong prior toward task completion, causing a model to predict success even when its action input should lead to failure, which we call *Optimism Bias*.

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.

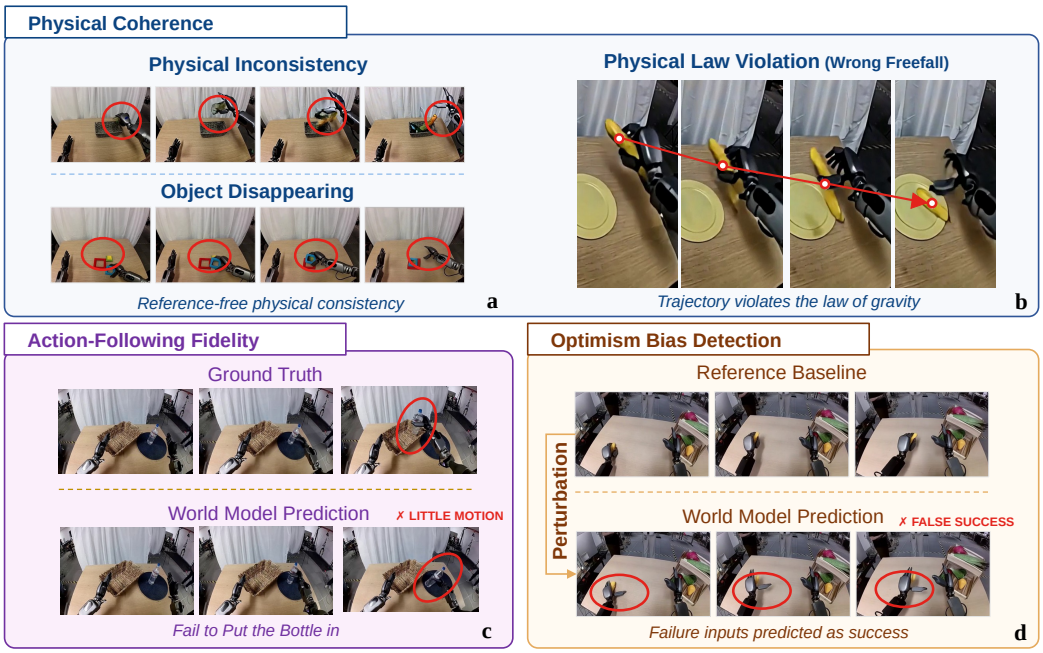


Figure 1: Representative failure modes targeted by MiraBench. (a–b) Physics Adherence failures include object morphing, disappearance, and implausible free-fall dynamics. (c) Action-Following failures arise when predicted motion is incomplete or inconsistent with the commanded action. (d) Optimism Bias arises when a failure-inducing action is mapped to a successful predicted outcome.

Evaluating action-conditioned reliability raises several measurement challenges. Physical correctness cannot rely on a unique reference future, since multiple outcomes may be visually valid. Action fidelity must also accommodate different conditioning interfaces, from low-level motor commands to natural-language instructions. Most critically, diagnosing optimism bias requires separating sensitivity to a failure-inducing action from generic variation in video generation. Scalable evaluation further requires automated judgments that remain consistent with human assessments.

To address these challenges, we introduce *MiraBench*. For *Physics Adherence*, MiraBench combines reference-free temporal consistency checks with trajectory-based kinematic tests, allowing physical violations to be detected without matching a single ground-truth future. For *Action-Following Fidelity*, it supports both precise motor actions and descriptive action instructions, and evaluates task completion, target-object grounding, and motion quality under the corresponding conditioning interface. For *Optimism Bias Detection*, MiraBench constructs paired nominal and counterfactual inputs by perturbing a successful action into a physically interpretable failure-inducing action while keeping the initial scene fixed, enabling direct measurement of whether the predicted outcome changes with the action. Finally, we ground the automated evaluators in 16,704 structured human judgments over 906 generated videos and use metric-specific temporal aggregation to obtain scalable yet human-aligned measurements. The three dimensions are scored separately to distinguish physical inconsistency, weak action grounding, and failure insensitivity.

We evaluate 12 representative model configurations spanning vector-conditioned robotic world models, text-conditioned generative world models, open-weight and closed-source systems, and multiple model scales, including systems developed by NVIDIA, Alibaba, and Kuaishou. Across this broad model landscape, three patterns emerge consistently. Visual fidelity is a poor proxy for action fidelity, increasing model scale does not reliably improve action following, and optimism bias remains widespread under failure-inducing actions. These findings show that gains in visual generation quality and model scale can substantially overstate progress toward faithful robotic simulators. They also identify action grounding and failure sensitivity as distinct unresolved challenges.

Our contributions are threefold:

- We formulate *action-conditioned reliability* as an evaluation target for robotic world models and formalize *Optimism Bias* as a failure mode in which failure-inducing actions are mapped to successful predicted outcomes.
- We introduce *MiraBench*, which operationalizes action-conditioned reliability through reference-free physical assessment, action-grounded evaluation across heterogeneous conditioning interfaces, and controlled failure-inducing interventions, with evaluators validated against 16,704 structured human judgments.
- Across 12 representative configurations, including systems from NVIDIA, Alibaba, and Kuaishou, we reveal a consistent gap between visual quality and action fidelity, limited gains from model scaling, and widespread optimism bias under failure-inducing actions.

2 Related Work

Action-conditioned world models are rooted in model-based reinforcement learning (Sutton, 1991; Ha & Schmidhuber, 2018), including Dreamer (Hafner et al., 2020; 2022; 2024) and pixel-space models such as IRIS (Micheli et al., 2023) and DIAMOND (Alonso et al., 2024). Recent systems increasingly model visually rich robotic interactions, including UniSim (Yang et al., 2024), IRASim (Zhu et al., 2025a), CtrlWorld (Guo et al., 2026), and DreamDojo (Gao et al., 2026), with related developments in driving (Hu et al., 2023; Wang et al., 2023; Gao et al., 2024) and open-world environments (Bruce et al., 2024; Zhou et al., 2024). Their evaluation has largely evolved from generic video metrics such as FVD (Unterthiner et al., 2019), VBench (Huang et al., 2023), EvalCrafter (Liu et al., 2024), and T2V-CompBench (Sun et al., 2025), toward world-model benchmarks such as WorldSimBench (Qin et al., 2024), WorldModelBench (Li et al., 2025), WorldScore (Duan et al., 2025), and WorldArena (Shang et al., 2026), which additionally consider physics and instruction following. MiraBench focuses on a complementary question: Whether generated futures remain reliable under the specific actions used to condition them, including which may lead to failure.

Physical reasoning benchmarks such as IntPhys (Riochet et al., 2020), CLEVRER (Yi et al., 2020), ComPhy (Chen et al., 2022), Physion (Bear et al., 2022), Physion++ (Tung et al., 2023), and PhyWorldBench (Gu et al., 2026) evaluate whether video models capture physical structure and dynamics. MiraBench connects physical judgment directly to action-conditioned consequences, such as whether an object falls when a release action is given. This requirement is also central to model-based planning (Schrittwieser et al., 2020; Hansen et al., 2022; Janner et al., 2021) and synthetic-data generation for robotics (Yu et al., 2023; Wang et al., 2024b;a), where predicted transitions must preserve the relationship between action and outcome. Research on domain randomization and sim-to-real transfer studies fidelity gaps in conventional simulators (Tobin et al., 2017; OpenAI et al., 2019). WorldArena (Shang et al., 2026) and CtrlWorld (Guo et al., 2026) further evaluate manipulation action following, while MiraBench explicitly constructs failure-inducing actions and measuring whether models preserve the resulting failures, enabling direct diagnosis of optimism bias.

3 Problem Formulation

World model as a conditional generator. Let $\mathcal{W}$ denote a robotic world model and $\mathcal{E}$ the true environment dynamics. Given an initial observation o_0, an action sequence $\mathbf{a}_{1:T}$:

$$\hat{\mathbf{v}}_{1:T} \sim p_{\mathcal{W}}(\cdot \mid o_0, \mathbf{a}_{1:T}), \qquad \mathbf{v}^*_{1:T} \sim p_{\mathcal{E}}(\cdot \mid o_0, \mathbf{a}_{1:T}), \tag{1}$$

where $\hat{\mathbf{v}}_{1:T}$ is the predicted trajectory and $\mathbf{v}^*_{1:T}$ is the environment trajectory. A faithful world model should preserve the consequences of the conditioning signal: its rollout should be physically admissible, semantically consistent with the specified action, and calibrated to failure when the action implies failure. We refer to this property as *action-conditioned reliability*.

Physical Adherence. Let $\Phi = \{\phi_1, \ldots, \phi_K\}$ denote a set of physical invariants that should hold for any realizable trajectory, including object persistence, coherent motion, contact causality, occlusion continuity, and simple physical laws. Rather than treating each invariant as a binary predicate, we associate each ϕ_k with a normalized violation degree $\delta_k(\hat{\mathbf{v}}) \in [0, 1]$, where 0 means the invariant

is satisfied and 1 indicates a severe violation. We define *Physical Adherence* as

$$\mathrm{PA}(\hat{\mathbf{v}}) = 1 - \frac{1}{K}\sum_{k=1}^{K}\delta_k(\hat{\mathbf{v}}). \tag{2}$$

It remains *reference-free*: physical validity is assessed from the generated rollout itself, without a ground-truth video. Thus, a prediction can appear visually realistic while still receiving low score if it violates persistence, contact dynamics, or basic kinematic constraints.

Optimism Bias. Let $\mathbf{a}^+$ denote a nominal action sequence that succeeds in the environment, and let $\mathbf{a}^- = \mathrm{Perturb}(\mathbf{a}^+, \tau, s)$ be a perturbation of type τ and severity s that should induce failure under true dynamics. Let $Y(\mathbf{v}, g) \in \{0, 1\}$ denote whether trajectory $\mathbf{v}$ depicts task success under goal g. By construction, $Y(\mathbf{v}^{*-}, g) = 0$ for $\mathbf{v}^{*-} \sim p_{\mathcal{E}}(\cdot \mid o_0, \mathbf{a}^-, g)$. We define *Optimism Bias* as:

$$\mathrm{OB}(\mathcal{W}) = \mathbb{E}_{o_0,g,\tau,s}\left[Y(\hat{\mathbf{v}}^-, g) \;\middle|\; \hat{\mathbf{v}}^- \sim p_{\mathcal{W}}(\cdot \mid o_0, \mathbf{a}^-, g),\ Y(\mathbf{v}^{*-}, g) = 0\right]. \tag{3}$$

A model with $\mathrm{OB}(\mathcal{W}) = 0$ preserves failure outcomes, whereas $\mathrm{OB}(\mathcal{W}) = 1$ always predicts success despite failure-inducing actions. Equivalently, we report *Failure Preservation* as $1-\mathrm{OB}(\mathcal{W})$, so higher values indicate better reliability.

4 Motivating Study

The decoupling hypothesis. We begin with a small-scale motivating study to test whether failures in action-conditioned reliability are already observable in current world models. Our central hypothesis is that success-dominated robot data can induce a decoupling between visual plausibility and action faithfulness. Let $\mathcal{D}^+ = \{(o_0^{(i)}, \mathbf{a}^{+(i)}, \mathbf{v}^{*(i)})\}$ denote a training distribution consisting primarily of successful demonstrations. Since counterfactual failures are rarely observed, a learned world model may explain future videos using a strong scene-conditioned success prior and only weak residual dependence on the action:

$$p_{\mathcal{W}}(\hat{\mathbf{v}} \mid o_0, \mathbf{a}) \propto p_{\text{prior}}(\hat{\mathbf{v}} \mid o_0) \cdot p_{\text{residual}}(\hat{\mathbf{v}} \mid o_0, \mathbf{a}), \tag{4}$$

where p_{prior} captures likely successful futures implied by the initial observation, and p_{residual} captures the remaining action-conditioned variation. Under this view, a failure-inducing action $\mathbf{a}^-$ can be overwhelmed by the success prior, causing the model to generate a plausible successful trajectory rather than the correct failure outcome. This hypothesis leads to two empirical predictions. First, models should exhibit measurable differences in physical consistency, since physical incoherence can mask action sensitivity. Second, visual quality should not be sufficient evidence of reliability: a model may produce plausible videos while still predicting success under actions that should fail.

Study setting. We test these predictions through a controlled human annotation study on four representative models: DreamDojo-2B (Gao et al., 2026), DreamDojo-14B (Gao et al., 2026), Wan2.1-14B (Wan et al., 2025), and Happy Horse (Alibaba Inc., 2026). This set covers different model scales, access types, and generation regimes, allowing us to examine whether the hypothesized failures are tied to a single model family or appear across heterogeneous systems. Trained annotators evaluated generated videos using structured rubrics rather than overall preference scores. For physical consistency, annotators scored fine-grained dimensions such as object shape, size, material behavior, background stability, and action-effect coherence. For optimism bias, annotators compared baseline and failure-perturbed predictions using multi-item judgments covering perturbation sensitivity, task-success prediction, false-success behavior, and failure preservation. This design allows the pilot study to distinguish physical incoherence from success-biased prediction, and to verify whether visually plausible generations can still be action-conditionally unreliable. An overview is provided in Fig.2. Full questionnaires, grading scales, and annotation instructions are provided in Appendix G. The annotation coverage and quality-control procedure are documented in Appendix G.5, while the corresponding VLM–human grounding and evaluator sanity checks are reported in Appendix D.3.

(a) Physical consistency varies sharply across models. The left panel of Table 1 reports physical-consistency pass rates across four dimensions on 30 GR-1 episodes. Even the strongest model, Happy Horse, does not achieve perfect consistency, while DreamDojo-2B passes only 17–30% of cases across dimensions. The stable ranking across dimensions, Happy Horse $>$ Wan2.1 $>$

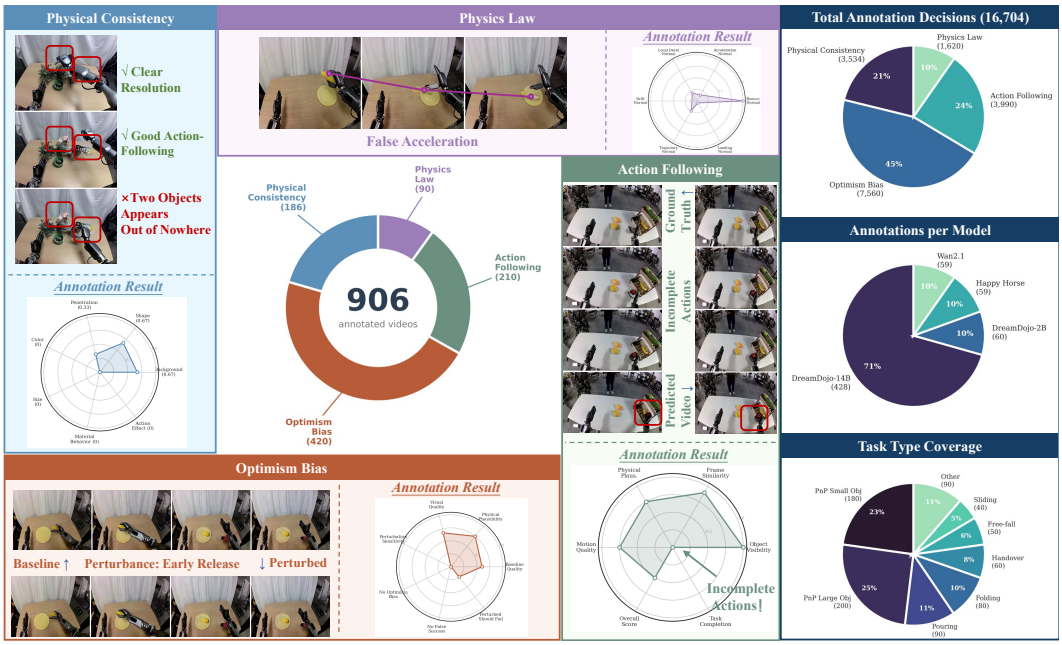


Figure 2: Overview of MiraBench's human annotation corpus. The corpus contains 906 generated videos and 16,704 structured human annotation decisions across three modules: (a) *Physical Adherence*, which includes *Physical Consistency* and *Physics Law Compliance*, (b) *Action Following*, and (c) *Optimism Bias Detection*. These annotations are collected on representative model outputs and provide per-indicator supervision for validating MiraBench evaluators, which are then used to assess a broader set of world models.

Table 1: Pilot human evaluation on GR-1 episodes. Physical consistency reports pass rates; the failure perturbation probe reports visual quality, physical plausibility, and optimism bias.

	Physical Consistency					Failure Perturbation Probe		
Model	Shape (%)↑	Size (%)↑	Material (%)↑	Background (%)↑	Average (%)↑	Visual Quality↑	Physical Plausibility (%)↑	Optimism Bias (%)↑
Happy Horse	90.0	90.0	93.3	100.0	93.3	2.2	96.6	35.7
Wan2.1-14B	46.7	53.3	50.0	76.7	56.7	0.5	6.9	50.0
DreamDojo-14B	36.7	43.3	34.5	53.3	42.0	2.8	93.1	15.0
DreamDojo-2B	16.7	30.0	23.3	26.7	24.2	1.8	40.0	23.3

DreamDojo-14B > DreamDojo-2B, suggests that these annotations capture model-level differences in physical fidelity rather than isolated artifacts.

(b) Optimism bias is prevalent and decoupled from visual quality. The right panel of Table 1 reports per-model optimism bias rates under six implicit failure perturbations. Happy Horse achieves high physical plausibility, yet still exhibits a high bias rate, indicating that strong generation quality does not guarantee faithful failure prediction. *DreamDojo-14B* shows a similar pattern, combining relatively high visual quality with substantial optimism bias. Conversely, *Wan2.1* performs better on the optimism-bias evaluation, but this should not be interpreted as overall reliability: its low visual quality and physical plausibility limit its ability to produce recognizable success predictions. These results support the need to evaluate action-conditioned reliability separately from visual realism.

Implications for benchmark design. The motivating study suggests three design requirements for MiraBench. First, physical consistency should be evaluated reference-free, since counterfactual actions often lack ground-truth videos. Second, optimism bias should be probed with failure-inducing perturbations whose correct outcomes are known under basic task physics, rather than with per-

turbations that merely degrade visual quality. Third, physical coherence, action sensitivity, and optimism bias should be reported separately and interpreted together: poor physical coherence can make action-sensitivity and optimism-bias scores ambiguous.

5 DESIGN OF MIRABENCH

5.1 CONSTRUCTION OF TEST CASES

MiraBench operationalizes *action-conditioned reliability* through the pipeline in Fig. 3. MiraBench is built from manipulation episodes in the GR-1 humanoid corpus (Gao et al., 2026) and the Lingchu bimanual dataset, covering contact-rich tasks such as pick-and-place, pouring, folding, bimanual handover, sliding, and free-fall interactions.

- For nominal evaluation, models receive an initial observation together with either a raw motor command sequence or a natural-language task instruction.
- For failure evaluation, we construct paired *intended-failure counterfactuals* by applying mechanically interpretable perturbations to nominal successful actions while keeping the initial scene fixed. The perturbations are chosen to remove or violate a physical precondition for successful task completion, such as sufficient grasp closure, correct release timing, or valid approach geometry. For descriptive-action models, the same failure modes are expressed as natural-language instructions.

The above design supports two complementary action modalities, *precise actions* and *descriptive actions*, allowing MiraBench to localize failures in motor grounding, semantic task understanding, or success-biased prediction. The resulting corpus contains 906 generated videos and 16,704 human annotation decisions across four annotation modules. Based on the corpus, MiraBench evaluates action-conditioned reliability along three complementary dimensions: *Physics Adherence*, *Action-Following Fidelity*, and *Optimism Bias Detection*.

Physics Adherence. Physics Adherence evaluates whether a generated rollout is physically meaningful, independently of the action-fidelity score. **Obj.** (Object Consistency) assesses whether manipulated objects retain a coherent appearance over time, while **Occ.** (Occlusion Consistency) assesses whether their identity is preserved through occlusion. **Rule** (Physical Rule Adherence) assesses whether object motion and contact behavior follow plausible physical dynamics.

Action-Following Fidelity. Action-Following Fidelity tests whether a generated rollout reflects the conditioning action. **TCR** (Task Completion Rate) measures whether the predicted rollout achieves the intended nominal task. **OPS** (Object Preservation Score) assesses whether task-relevant objects remain intact and recognizable in the prediction. **Gen.** (Generalization) captures how well task-completion performance transfers from the GR-1 setting to out-of-distribution settings. To avoid overfitting the evaluation to a single embodiment, MIRABENCH includes cross-embodiment settings spanning *GR-1*, *Unitree G1*, *DROID* (Khazatsky et al., 2025), and bimanual manipulation.

Optimism Bias Detection. Optimism Bias Detection probes whether a model preserves failure when the action implies failure. We define six failure-inducing perturbation families grounded in common manipulation failures: insufficient grip force, premature release, carry slip, contact oscillation, wrist tilt, and approach overshoot. Each perturbation keeps the visual scene fixed while changing the action-implied outcome. For each episode, the model generates a nominal rollout and a counterfactual rollout. **Bias Res.** (Bias Resistance) measures whether the model preserves the failure implied by a perturbed action instead of predicting the nominal successful outcome.

See Table 8 for metric definitions and Appendix D for detailed scoring procedures.

5.2 HUMAN-GROUNDED EVALUATOR

MIRABENCH uses human annotation as supervision for automated evaluators. We obtain structured human annotations on physical consistency, physics-law compliance, action-following outcomes, and optimism-bias indicators. For automated evaluation, we deploy frontier-scale VLM models as zero-shot judges and use task-specific temporal aggregation. Physical consistency compares 10 pairs formed from 20 evenly spaced frames; task completion is judged once from a 16-frame sequence, while object preservation averages 16 frame-pair judgments; optimism-bias detection uses

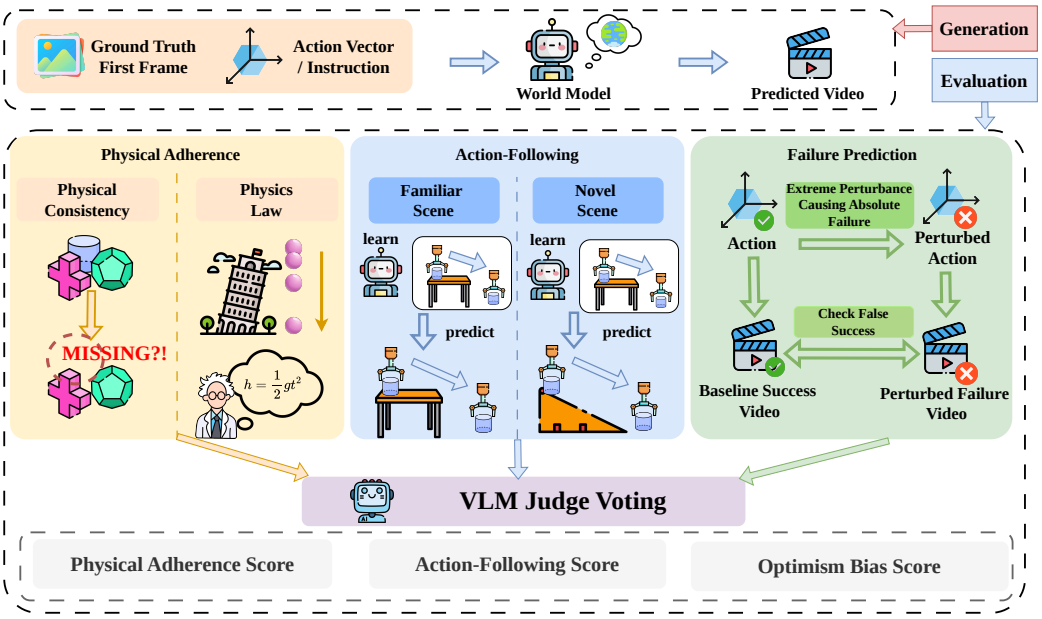


Figure 3: Overview of MIRABENCH. From robotic manipulation episodes, MIRABENCH constructs nominal and failure-inducing action inputs, evaluates generated rollouts along three complementary dimensions of action-conditioned reliability, and uses VLM evaluators validated against structured human annotations. Rule-based kinematic checks complement VLM for physics-law compliance.

a majority vote over late-phase frame comparisons. These procedures reduce sensitivity to individual frame noise while preserving temporal context. On held-out expert-labeled samples, per-metric VLM–human agreement ranges from 85.1% to 92.3%, with 95% bootstrap confidence intervals reported in Table 11. The annotated validation set spans DreamDojo-2B/14B, WAN2.1, and Happy Horse across model scales, vector/text conditioning, and open/closed access; model-group agreement ranges from 87.3% to 96.2% (Table 12), supporting high evaluator consistency across these tested model types. Annotation-team QA, low-ambiguity labeling criteria, model/interface coverage, and manual evaluator sanity checks are documented in Appendices D.3 and G.5.

6 EXPERIMENTS AND ANALYSIS

We evaluate MIRABENCH on 12 representative model configurations spanning vector-conditioned robotic world models and text-conditioned generative world models. The vector-conditioned group includes *DreamDojo*, *DreamDojo-GR1*, and *Cosmos-GR1* at 2B and 14B scales, all conditioned on raw action vectors and an initial frame. The text-conditioned group includes *Cosmos-GR1* in text mode, *Wan2.1/2.2*, *WanX*, *Happy Horse*, and *Kling*, conditioned on natural-language task instructions and an initial frame. This suite covers conditioning paradigms, model scales, access types, and systems released by major industrial and academic groups. All scores are produced by MIRABENCH's human-grounded evaluators, and Table 2 reports the complete scores.

Finding 1: Action-following is decoupled from failure preservation. In Fig. 4(a), MIRABENCH reveals a clear gap between performance under nominal actions and reliability under failure-inducing actions. Models with strong action-following scores can still fail to preserve failure outcomes. For example, *DreamDojo-GR1-14B* achieves high task completion and object preservation, yet its failure-preservation score remains among the lowest. Happy Horse similarly performs strongly on physics and action following, but moderately on failure preservation. Thus, successful-looking rollouts do not imply a world model preserves the causal relationship between actions and outcomes.

Table 2: Main results on MiraBench across representative vector-conditioned and text-conditioned world models. Results are grouped by three complementary reliability dimensions. Higher is better for all scores.

Model	Scale / Access	Physics Adherence			Action-Following			Failure Preservation
		Obj.↑	Occ.↑	Rule↑	TCR↑	OPS↑	Gen.↑	Bias Res.↑
Vector-conditioned models								
DreamDojo	2B	21.3	58.7	7.0	14.0	90.0	100.0	48.7
DreamDojo-GR1	2B	43.3	64.0	25.5	92.0	98.0	42.6	23.1
DreamDojo	14B	38.7	79.0	7.8	18.0	92.0	100.0	12.8
DreamDojo-GR1	14B	53.0	68.3	22.7	92.0	96.0	42.6	12.8
Cosmos-GR1	2B	30.5	51.0	17.0	68.0	98.0	54.2	17.5
Cosmos-GR1	14B	30.3	53.7	14.0	54.0	96.0	71.2	15.4
Text-conditioned models								
Cosmos-GR1	14B	10.3	42.3	10.9	76.0	56.0	97.4	30.8
WAN 2.1	14B	51.7	67.3	18.1	58.0	74.0	100.0	87.2
WAN 2.2	5B	65.2	81.4	10.6	92.0	90.0	72.6	56.4
Happy Horse	Closed	92.0	94.0	12.7	96.0	94.0	97.4	56.4
WanX	Closed	77.2	87.9	10.3	70.0	90.0	100.0	46.2
Kling 3.0 Omni	Closed	85.7	100.0	14.6	100.0	94.0	100.0	35.9

Obj. = Object Consistency; Occ. = Occlusion Consistency; Rule = Physical Rule Adherence; TCR = Task Completion Rate; OPS = Object Preservation Score; Gen. = Generalization; Bias Res. = Bias Resistance. Vector-conditioned Cosmos-GR1 entries use 6000 fine-tuning steps.

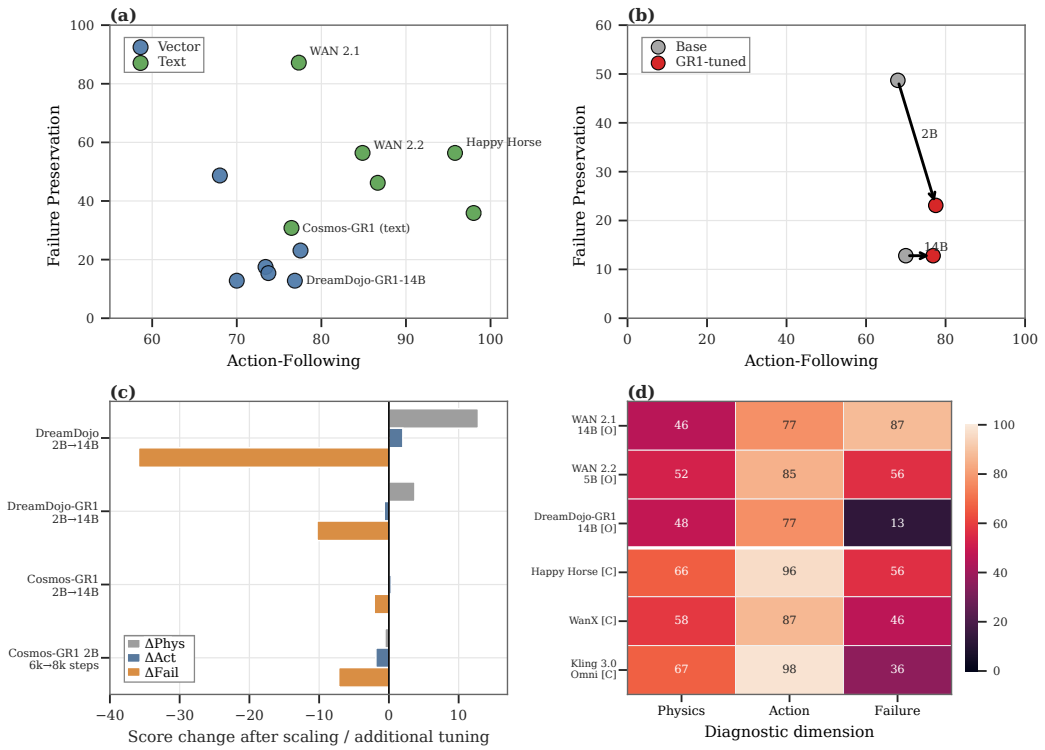


Figure 4: (a) Action-following does not guarantee failure preservation. (b) GR1 post-training improves task execution while weakening failure preservation at fixed scale. (c) Scaling and additional fine-tuning produce non-uniform changes across the three diagnostic dimensions (ΔPhys, ΔAct, and ΔFail denote changes in physics adherence, action following, and failure preservation). (d) Open-weight [O] and closed-source [C] systems show complementary strengths, but no model dominates all dimensions.

Finding 2: Success-only post-training improves the wrong capability. Fig.4(b) shows GR1 post-training substantially improves nominal action-following but not failure preservation. For *DreamDojo-2B*, task completion increases from 14.0 to 92.0 and object preservation from 90.0 to 98.0 after post-training, while failure preservation drops from 48.7 to 23.1. At 14B scale, task

completion rises from 18.0 to 92.0, but failure preservation remains low. Success-dominated post-training strengthens normal task execution without teaching the model to represent failed outcomes.

Finding 3: Scaling and extra fine-tuning are not reliable fixes. Fig.4(c) compares changes under scaling and additional fine-tuning. Within the *DreamDojo* family, the 14B model improves some physical-consistency metrics over the 2B model, but failure preservation drops from 48.7 to 12.8. *DreamDojo-GR1* shows nearly unchanged nominal action-following from 2B to 14B, while failure preservation again decreases. *Cosmos-GR1* also shows limited evidence that scale alone improves failure preservation. The *Cosmos-GR1* step ablation further shows that increasing fine-tuning from 6000 to 8000 steps changes the action-following profile without producing a monotonic gain in action-conditioned reliability. These results indicate that the observed failures are not merely capacity or optimization deficits, but reflect missing supervision for action-outcome contingency.

Finding 4: Model families differ, but no current system performs best across all dimensions. As summarized in Fig. 4(d), text-conditioned, vector-conditioned, open-weight, and closed-source models exhibit distinct failure profiles. Some text-conditioned systems achieve stronger failure preservation, while several vector-conditioned models achieve high nominal action-following under raw action inputs. However, no evaluated model dominates Physics Adherence, Action-Following Fidelity, and Failure Preservation simultaneously. *Wan2.1* attains the strongest failure-preservation score but is not uniformly strongest in physics or nominal action-following, while Happy Horse performs strongly on physics and action-following but still leaves a gap in failure preservation. This heterogeneity motivates MIRABENCH's multidimensional design: a single aggregate visual-quality score would obscure whether a model fails through physical incoherence, action insensitivity, or success-biased prediction.

Further Implications. These findings show that generating successful-looking videos is not equivalent to learning a faithful action-conditioned simulator. Mitigating optimism bias will likely require failure-aware training data, objectives that contrast nominal and failure-inducing actions, and physical priors that rule out impossible trajectories. MIRABENCH provides a diagnostic testbed for measuring which aspects of reliability such interventions improve.

Case study and human validation. Figure 1 shows representative failures corresponding to the quantitative patterns above, including object persistence errors, physics-law violations, action mismatch, and success-biased failure suppression. Additional galleries are provided in Appendix J, Appendix K, and Appendix L. All automated scores are grounded in the human annotation corpus described in Section 5, which contains 906 generated videos and 16,704 structured annotation decisions across *Physical Adherence* (*Physical Consistency* and *Physics Law Compliance*), *Action Following*, and *Optimism Bias Detection*. Dataset examples and corpus statistics are summarized in Fig. 2, with full rubrics and subset breakdowns in Appendix G and Appendix H.

7 CONCLUSION

We introduce MIRABENCH for evaluating *action-conditioned reliability* in robotic world models. By examining Physics Adherence, Action-Following Fidelity, and Optimism Bias Detection as complementary dimensions, MiraBench shifts evaluation from visual plausibility to simulator faithfulness. Its fine-grained human annotation corpus of 906 videos and 16,704 structured decisions provides supervision for human-grounded VLM evaluators, enabling scalable assessment of 12 representative model configurations. Our results show that current world models can appear visually plausible while failing to preserve action-outcome contingency: visual quality is a weak proxy for action fidelity, scaling and post-training do not reliably improve failure preservation, and optimism bias remains widespread. We hope MiraBench provides a diagnostic foundation for building world models which are not only realistic to watch, but reliable as simulators for embodied AI.

AI USE STATEMENT

We used AI assistance to aid or polish writing through language editing and readability improvements, and for generating synthetic datasets: Gemini-2.5-Pro was used to generate natural-language task and failure-perturbation prompts, the evaluated world models generated synthetic rollout videos for the benchmark, and VLMs served as structured evaluators for the released videos. The authors

designed the data-generation protocol and evaluation rules, manually verified generated prompts, checked evaluator outputs against human annotations, performed the final data filtering and analysis, reviewed all AI-assisted text, and take responsibility for the final wording, claims, datasets, analyses, and artifacts.

REFERENCES

Alibaba Inc. Happyhorse. https://www.happyhorse.cn/, 2026. Accessed: 2026-05-06.

Eloi Alonso, Adam Jelley, Vincent Micheli, Anssi Kanervisto, Amos Storkey, Tim Pearce, and François Fleuret. Diffusion for world modeling: Visual details matter in atari, 2024. URL https://arxiv.org/abs/2405.12399.

Daniel M. Bear, Elias Wang, Damian Mrowca, Felix J. Binder, Hsiao-Yu Fish Tung, R. T. Pramod, Cameron Holdaway, Sirui Tao, Kevin Smith, Fan-Yun Sun, Li Fei-Fei, Nancy Kanwisher, Joshua B. Tenenbaum, Daniel L. K. Yamins, and Judith E. Fan. Physion: Evaluating physical prediction from vision in humans and machines, 2022. URL https://arxiv.org/abs/2106.08261.

Anthony Brohan, Noah Brown, Justice Carbajal, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control, 2023. URL https://arxiv.org/abs/2307.15818.

Jake Bruce, Michael Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, Yusuf Aytar, Sarah Bechtle, Feryal Behbahani, Stephanie 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, 2024. URL https://arxiv.org/abs/2402.15391.

Remi Cadene, Simon Aliberts, Francesco Capuano, Michel Aractingi, Adil Zouitine, Pepijn Kooijmans, Jade Choghari, Martino Russi, Caroline Pascal, Steven Palma, Mustafa Shukor, Jess Moss, Alexander Soare, Dana Aubakirova, Quentin Lhoest, Quentin Gallouédec, and Thomas Wolf. Lerobot: An open-source library for end-to-end robot learning, 2026. URL https://arxiv.org/abs/2602.22818.

Zhenfang Chen, Kexin Yi, Yunzhu Li, Mingyu Ding, Antonio Torralba, Joshua B. Tenenbaum, and Chuang Gan. Comphy: Compositional physical reasoning of objects and events from videos, 2022. URL https://arxiv.org/abs/2205.01089.

Open X-Embodiment Collaboration, Abby O'Neill, Abdul Rehman, et al. Open x-embodiment: Robotic learning datasets and rt-x models, 2025. URL https://arxiv.org/abs/2310.08864.

Haoyi Duan, Hong-Xing Yu, Sirui Chen, Li Fei-Fei, and Jiajun Wu. Worldscore: A unified evaluation benchmark for world generation, 2025. URL https://arxiv.org/abs/2504.00983.

Shenyuan Gao, Jiazhi Yang, Li Chen, Kashyap Chitta, Yihang Qiu, Andreas Geiger, Jun Zhang, and Hongyang Li. Vista: A generalizable driving world model with high fidelity and versatile controllability, 2024. URL https://arxiv.org/abs/2405.17398.

Shenyuan Gao, William Liang, Kaiyuan Zheng, Ayaan Malik, Seonghyeon Ye, Sihyun Yu, Wei-Cheng Tseng, Yuzhu Dong, Kaichun Mo, Chen-Hsuan Lin, Qianli Ma, Seungjun Nah, Loic Magne, Jiannan Xiang, Yuqi Xie, Ruijie Zheng, Dantong Niu, You Liang Tan, K. R. Zentner, George Kurian, Suneel Indupuru, Pooya Jannaty, Jinwei Gu, Jun Zhang, Jitendra Malik, Pieter Abbeel, Ming-Yu Liu, Yuke Zhu, Joel Jang, and Linxi "Jim" Fan. Dreamdojo: A generalist robot world model from large-scale human videos, 2026. URL https://arxiv.org/abs/2602.06949.

Jing Gu, Xian Liu, Yu Zeng, Ashwin Nagarajan, Fangrui Zhu, Daniel Hong, Yue Fan, Qianqi Yan, Kaiwen Zhou, Ming-Yu Liu, and Xin Eric Wang. "phyworldbench": A comprehensive evaluation of physical realism in text-to-video models, 2026. URL https://arxiv.org/abs/2507.13428.

Yanjiang Guo, Lucy Xiaoyang Shi, Jianyu Chen, and Chelsea Finn. Ctrl-world: A controllable generative world model for robot manipulation, 2026. URL https://arxiv.org/abs/2510.10125.

David Ha and Jürgen Schmidhuber. World models. 2018. doi: 10.5281/ZENODO.1207631. URL https://zenodo.org/record/1207631.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination, 2020. URL https://arxiv.org/abs/1912.01603.

Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models, 2022. URL https://arxiv.org/abs/2010.02193.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models, 2024. URL https://arxiv.org/abs/2301.04104.

Nicklas Hansen, Xiaolong Wang, and Hao Su. Temporal difference learning for model predictive control, 2022. URL https://arxiv.org/abs/2203.04955.

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, 2023. URL https://arxiv.org/abs/2309.17080.

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, 2023. URL https://arxiv.org/abs/2311.17982.

Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization, 2021. URL https://arxiv.org/abs/1906.08253.

Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, Peter David Fagan, Joey Hejna, Masha Itkina, Marion Lepert, Yecheng Jason Ma, Patrick Tree Miller, Jimmy Wu, Suneel Belkhale, Shivin Dass, Huy Ha, Arhan Jain, Abraham Lee, Youngwoon Lee, Marius Memmel, Sungjae Park, Ilija Radosavovic, Kaiyuan Wang, Albert Zhan, Kevin Black, Cheng Chi, Kyle Beltran Hatch, Shan Lin, Jingpei Lu, Jean Mercat, Abdul Rehman, Pannag R Sanketi, Archit Sharma, Cody Simpson, Quan Vuong, Homer Rich Walke, Blake Wulfe, Ted Xiao, Jonathan Heewon Yang, Arefeh Yavary, Tony Z. Zhao, Christopher Agia, Rohan Baijal, Mateo Guaman Castro, Daphne Chen, Qiuyu Chen, Trinity Chung, Jaimyn Drake, Ethan Paul Foster, Jensen Gao, Vitor Guizilini, David Antonio Herrera, Minho Heo, Kyle Hsu, Jiaheng Hu, Muhammad Zubair Irshad, Donovon Jackson, Charlotte Le, Yunshuang Li, Kevin Lin, Roy Lin, Zehan Ma, Abhiram Maddukuri, Suvir Mirchandani, Daniel Morton, Tony Nguyen, Abigail O'Neill, Rosario Scalise, Derick Seale, Victor Son, Stephen Tian, Emi Tran, Andrew E. Wang, Yilin Wu, Annie Xie, Jingyun Yang, Patrick Yin, Yunchu Zhang, Osbert Bastani, Glen Berseth, Jeannette Bohg, Ken Goldberg, Abhinav Gupta, Abhishek Gupta, Dinesh Jayaraman, Joseph J Lim, Jitendra Malik, Roberto Martín-Martín, Subramanian Ramamoorthy, Dorsa Sadigh, Shuran Song, Jiajun Wu, Michael C. Yip, Yuke Zhu, Thomas Kollar, Sergey Levine, and Chelsea Finn. Droid: A large-scale in-the-wild robot manipulation dataset, 2025. URL https://arxiv.org/abs/2403.12945.

Dacheng Li, Yunhao Fang, Yukang Chen, Shuo Yang, Shiyi Cao, Justin Wong, Michael Luo, Xiaolong Wang, Hongxu Yin, Joseph E. Gonzalez, Ion Stoica, Song Han, and Yao Lu. Worldmodelbench: Judging video generation models as world models, 2025. URL https://arxiv.org/abs/2502.20694.

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, 2024. URL https://arxiv.org/abs/2310.11440.

Ajay Mandlekar, Danfei Xu, Josiah Wong, Soroush Nasiriany, Chen Wang, Rohun Kulkarni, Li Fei-Fei, Silvio Savarese, Yuke Zhu, and Roberto Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation, 2021. URL https://arxiv.org/abs/2108.03298.

Vincent Micheli, Eloi Alonso, and François Fleuret. Transformers are sample-efficient world models, 2023. URL https://arxiv.org/abs/2209.00588.

NVIDIA, Niket Agarwal, Arslan Ali, et al. Cosmos world foundation model platform for physical ai, 2025a. URL https://arxiv.org/abs/2501.03575.

NVIDIA, Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi "Jim" Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, Joel Jang, Zhenyu Jiang, Jan Kautz, Kaushil Kundalia, Lawrence Lao, Zhiqi Li, Zongyu Lin, Kevin Lin, Guilin Liu, Edith Llontop, Loic Magne, Ajay Mandlekar, Avnish Narayan, Soroush Nasiriany, Scott Reed, You Liang Tan, Guanzhi Wang, Zu Wang, Jing Wang, Qi Wang, Jiannan Xiang, Yuqi Xie, Yinzhen Xu, Zhenjia Xu, Seonghyeon Ye, Zhiding Yu, Ao Zhang, Hao Zhang, Yizhou Zhao, Ruijie Zheng, and Yuke Zhu. Gr00t n1: An open foundation model for generalist humanoid robots, 2025b. URL https://arxiv.org/abs/2503.14734.

OpenAI, Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, Jonas Schneider, Szymon Sidor, Josh Tobin, Peter Welinder, Lilian Weng, and Wojciech Zaremba. Learning dexterous in-hand manipulation, 2019. URL https://arxiv.org/abs/1808.00177.

Yiran Qin, Zhelun Shi, Jiwen Yu, Xijun Wang, Enshen Zhou, Lijun Li, Zhenfei Yin, Xihui Liu, Lu Sheng, Jing Shao, Lei Bai, Wanli Ouyang, and Ruimao Zhang. Worldsimbench: Towards video generation models as world simulators, 2024. URL https://arxiv.org/abs/2410.18072.

Ronan Riochet, Mario Ynocente Castro, Mathieu Bernard, Adam Lerer, Rob Fergus, Véronique Izard, and Emmanuel Dupoux. Intphys: A framework and benchmark for visual intuitive physics reasoning, 2020. URL https://arxiv.org/abs/1803.07616.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, Timothy Lillicrap, and David Silver. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839), December 2020. ISSN 1476-4687. doi: 10.1038/s41586-020-03051-4. URL http://dx.doi.org/10.1038/s41586-020-03051-4.

Yu Shang, Zhuohang Li, Yiding Ma, Weikang Su, Xin Jin, Ziyou Wang, Lei Jin, Xin Zhang, Yinzhou Tang, Haisheng Su, Chen Gao, Wei Wu, Xihui Liu, Dhruv Shah, Zhaoxiang Zhang, Zhibo Chen, Jun Zhu, Yonghong Tian, Tat-Seng Chua, Wenwu Zhu, and Yong Li. Worldarena: A unified benchmark for evaluating perception and functional utility of embodied world models, 2026. URL https://arxiv.org/abs/2602.08971.

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, 2025. URL https://arxiv.org/abs/2407.14505.

Richard S. Sutton. Dyna, an integrated architecture for learning, planning, and reacting. *SIGART Bull.*, 2(4), July 1991. ISSN 0163-5719. doi: 10.1145/122344.122377. URL https://doi.org/10.1145/122344.122377.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, You Liang Tan, Lawrence Yunliang Chen, Pannag Sanketi, Quan Vuong, Ted Xiao, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy, 2024. URL https://arxiv.org/abs/2405.12213.

Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world, 2017. URL https://arxiv.org/abs/1703.06907.

Hsiao-Yu Tung, Mingyu Ding, Zhenfang Chen, Daniel Bear, Chuang Gan, Joshua B. Tenenbaum, Daniel LK Yamins, Judith E Fan, and Kevin A. Smith. Physion++: Evaluating physical scene understanding that requires online inference of different physical properties, 2023. URL https://arxiv.org/abs/2306.15668.

Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges, 2019. URL https://arxiv.org/abs/1812.01717.

Homer Walke, Kevin Black, Abraham Lee, Moo Jin Kim, Max Du, Chongyi Zheng, Tony Zhao, Philippe Hansen-Estruch, Quan Vuong, Andre He, Vivek Myers, Kuan Fang, Chelsea Finn, and Sergey Levine. Bridgedata v2: A dataset for robot learning at scale, 2024. URL https://arxiv.org/abs/2308.12952.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models, 2025. URL https://arxiv.org/abs/2503.20314.

Lirui Wang, Yiyang Ling, Zhecheng Yuan, Mohit Shridhar, Chen Bao, Yuzhe Qin, Bailin Wang, Huazhe Xu, and Xiaolong Wang. Gensim: Generating robotic simulation tasks via large language models, 2024a. URL https://arxiv.org/abs/2310.01361.

Xiaofeng Wang, Zheng Zhu, Guan Huang, Xinze Chen, Jiagang Zhu, and Jiwen Lu. Drivedreamer: Towards real-world-driven world models for autonomous driving, 2023. URL https://arxiv.org/abs/2309.09777.

Yufei Wang, Zhou Xian, Feng Chen, Tsun-Hsuan Wang, Yian Wang, Katerina Fragkiadaki, Zackory Erickson, David Held, and Chuang Gan. Robogen: Towards unleashing infinite data for automated robot learning via generative simulation, 2024b. URL https://arxiv.org/abs/2311.01455.

Sherry Yang, Yilun Du, Kamyar Ghasemipour, Jonathan Tompson, Leslie Kaelbling, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators, 2024. URL https://arxiv.org/abs/2310.06114.

Kexin Yi, Chuang Gan, Yunzhu Li, Pushmeet Kohli, Jiajun Wu, Antonio Torralba, and Joshua B. Tenenbaum. Clevrer: Collision events for video representation and reasoning, 2020. URL https://arxiv.org/abs/1910.01442.

Tianhe Yu, Ted Xiao, Austin Stone, Jonathan Tompson, Anthony Brohan, Su Wang, Jaspiar Singh, Clayton Tan, Dee M, Jodilyn Peralta, Brian Ichter, Karol Hausman, and Fei Xia. Scaling robot learning with semantically imagined experience, 2023. URL https://arxiv.org/abs/2302.11550.

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, 2025. URL https://arxiv.org/abs/2503.21755.

Siyuan Zhou, Yilun Du, Jiaben Chen, Yandong Li, Dit-Yan Yeung, and Chuang Gan. Robodreamer: Learning compositional world models for robot imagination, 2024. URL https://arxiv.org/abs/2404.12377.

Fangqi Zhu, Hongtao Wu, Song Guo, Yuxiao Liu, Chilam Cheang, and Tao Kong. Irasim: A fine-grained world model for robot manipulation, 2025a. URL https://arxiv.org/abs/2406.14540.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, Zhangwei Gao, Erfei Cui, Xuehui Wang, Yue Cao, Yangzhou Liu, Xingguang Wei, Hongjie Zhang, Haomin Wang, Weiye Xu, Hao Li, Jiahao Wang, Nianchen Deng,

Songze Li, Yinan He, Tan Jiang, Jiapeng Luo, Yi Wang, Conghui He, Botian Shi, Xingcheng Zhang, Wenqi Shao, Junjun He, Yingtong Xiong, Wenwen Qu, Peng Sun, Penglong Jiao, Han Lv, Lijun Wu, Kaipeng Zhang, Huipeng Deng, Jiaye Ge, Kai Chen, Limin Wang, Min Dou, Lewei Lu, Xizhou Zhu, Tong Lu, Dahua Lin, Yu Qiao, Jifeng Dai, and Wenhai Wang. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models, 2025b. URL https://arxiv.org/abs/2504.10479.

APPENDIX

A OFFICIAL DATASET INTRODUCTION

We leverage two large-scale robot manipulation datasets for training and evaluation: the **NVIDIA Arena-GR1-Manipulation-Task** dataset and the **PsiBot SynData** dataset. Together, they provide complementary coverage of embodied manipulation settings: the former supplies simulation-generated humanoid robot trajectories with precise joint-space control signals, while the latter contributes real-world full-modality bimanual hand demonstration data at scale. Figure 5 illustrates the respective data collection pipelines.

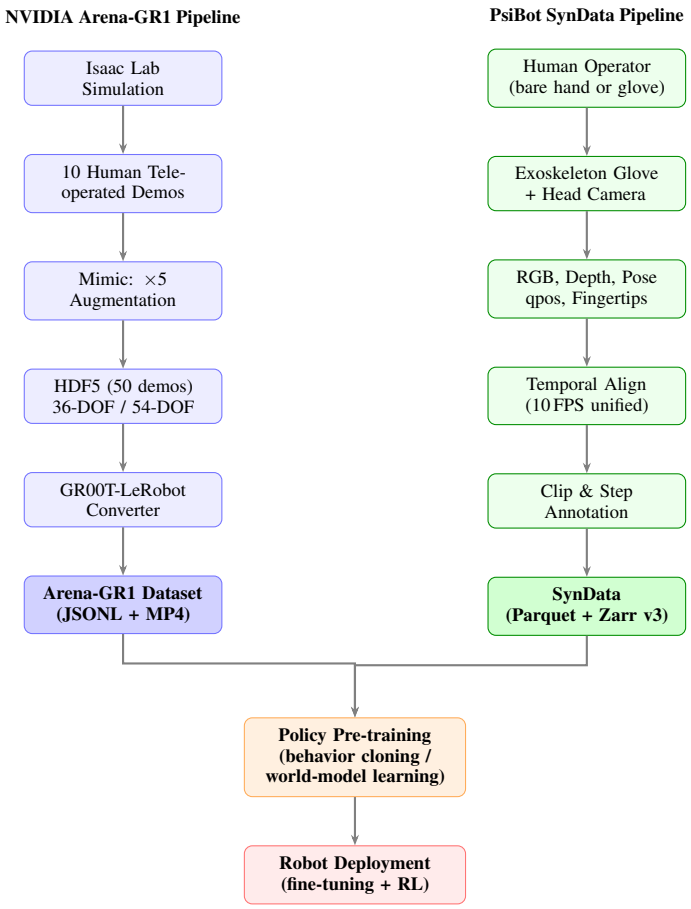


Figure 5: Data collection and processing pipelines. **Left (blue):** Arena-GR1 starts from 10 tele-operated demonstrations in Isaac Lab, expands to 50 via Mimic, and is stored as JSONL metadata and MP4 videos in GR00T-LeRobot format. **Right (green):** SynData captures real-world bimanual demonstrations with a full-modality exoskeleton rig, temporally aligns all streams to 10 FPS, and stores trajectory data in Zarr v3 with Parquet-based indexing and clip-/step-level annotations. Both datasets feed a shared policy pre-training stage followed by robot-specific fine-tuning.

A.1 NVIDIA ARENA-GR1-MANIPULATION-TASK DATASET

Overview. The Arena-GR1-Manipulation-Task dataset NVIDIA et al. (2025b) is an open-source robot manipulation benchmark released by NVIDIA under the CC-BY-4.0 license as part of the Isaac GR00T initiative. It is designed for training and evaluating generalist humanoid manipulation policies via behavior cloning and simulation-to-reality transfer research. The dataset conforms to the GR00T-LeRobot v2.0 specification Cadene et al. (2026), making it directly compatible with GR00T N1.5 post-training pipelines.

Robot Platform and Task. Data are collected in the **IsaacLab-Arena** simulation environment using the **Fourier GR-1** humanoid robot NVIDIA et al. (2025b). The dataset focuses on a single contact-rich manipulation task: *opening a microwave door*. Visual input is captured at 512×512 resolution from a first-person-view RGB camera. All demonstrations are generated at 50 Hz.

Data Collection. The dataset combines two complementary acquisition strategies:

- **Human teleoperation.** 10 demonstrations are collected by a human operator controlling the GR-1 within Isaac Lab via a depth camera and keyboard interface.
- **Synthetic augmentation.** 50 demonstrations are generated automatically from the 10 seed trajectories using MimicGen(5 per seed), forming the complete training corpus. The 10 human-annotated seeds and the 50 MimicGen-generated demonstrations are stored as two separate HDF5 files.

All demonstrations are generated at 50 Hz and subsequently converted from HDF5 to the GR00T-LeRobot format for downstream training.

Data Format and Structure. The released dataset provides three artefacts:

- `arena_gr1_manipulation_dataset_annotated.hdf5` the 10 human-annotated seed demonstrations.
- `arena_gr1_manipulation_dataset_generated.hdf5` the full 50 Mimic-generated demonstrations in raw HDF5.
- `lerobot/` the GR00T-LeRobot formatted version of the 50 demonstrations, organised as `data/chunk-{id}/episode_{id}.parquet` and `videos/chunk-{id}/`.

The per-frame schema of the LeRobot artefact is detailed in Table 3.

Table 3: Per-frame data schema of the Arena-GR1-Manipulation-Task (LeRobot format).

Field	Type	Description
`action`	`float64`	Desired joint positions (36 DoF)
`observation.state`	`float64`	Measured joint positions (54 DoF)
`timestamp`	`float64`	Simulation time in seconds
`episode_index`	`int64`	Episode (demo) identifier
`task_index`	`int64`	Multi-task loader index (always 0)
`annotation.human.action.task_description`	`int64`	Index to language instruction in metadata
`annotation.human.action.valid`	`int64`	Annotation validity flag

Dataset Statistics. The full dataset comprises 50 demonstrations (50 RGB videos) with a total storage footprint of 5.16 GB. All trajectories are generated at 50 Hz. Metadata is provided through four files: `episodes.jsonl` (episode list and lengths), `tasks.jsonl` (task list), `modality.json` (modality configuration), and `info.json` (dataset-level statistics).

Train / Validation Split. Following standard practice, we reserve 5 demonstrations (10%) as a held-out validation set and use the remaining 45 for training.

A.2 PsiBot SynData Dataset

Overview. SynData is a large-scale real-world multimodal dataset released by **PsiBot**, publicly available on Hugging Face. Despite its name, SynData consists entirely of *real-world human demonstrations* "Syn" refers to its *synchronized* full-modality capture pipeline, not synthetic generation. It is designed to serve as a foundational pre-training corpus for embodied intelligence models, with a focus on data quality, multimodal consistency, and temporal alignment.

Data Collection and Sensing Platform. Demonstrations are collected from human operators performing everyday bimanual manipulation tasks using two complementary capture modes:

- **Exoskeleton glove mode.** A proprietary exoskeleton glove achieves millimeter-level positioning accuracy and captures the full degrees of freedom of both hands and arms while preserving natural operator behavior.
- **Bare-hand mode.** Unencumbered natural interaction data is collected to complement the structured glove recordings and enrich behavioral diversity.

Modalities. Every clip simultaneously records the following sensor streams, all temporally aligned to a unified 10 FPS timeline:

Table 4: Modalities in the SynData dataset.

Modality	Description
`head_rgb`	RGB image sequence from the head camera
`head_depth`	Depth image sequence from the head camera
`head_camera_intrinsics`	Intrinsic parameters of the head camera
`head_tracker2head_camera`	Extrinsic: head tracker $\rightarrow$ head camera
`head_pose`	Head 6-DOF pose in world frame
`left_wrist_pose` / `right_wrist_pose`	Left/right wrist 6-DOF pose
`left_qpos` / `right_qpos`	Left/right hand joint states
`left_fingertip` / `right_fingertip`	Fingertip keypoints (left/right hand)

Data Format and Organisation. SynData adopts a clip-based organisation stored in Zarr v3 format. Each clip represents one complete task execution and is the minimum training unit. The directory layout is:

- `task.json` task ID-to-name mapping.
- `index/clips.parquet` global clip index with fields `clip_id`, `task_key`, `volume_id`, `rel_path`, `start_idx`, `end_idx`, and `num_frames`.
- `annotations/clip_annotations.parquet` clip-level semantic descriptions.
- `annotations/clip_steps.parquet` step-level annotations for long-horizon task decomposition and hierarchical policy learning.
- `tasks/<task_id>/<volume_id>.zarr` multimodal data volumes distributed as `.zarr.tar` packages, each storing modalities as independent arrays with the time dimension first.

Task Coverage. The dataset covers a wide spectrum of everyday bimanual manipulation tasks, including clothes sorting, box sorting, fragile-item packing, and further categories spanning contact-rich household and industrial scenarios. Step-level annotations support long-horizon task modeling and hierarchical policy learning.

Train / Validation Split. We apply stratified sampling across task categories, reserving 10% of clips per task as a held-out validation set and using the remaining 90% for pre-training.

B Implicit Failure Perturbation Taxonomy

We define six implicit failure perturbations that modify the GR-1 action vector to encode physically interpretable manipulation failures. The GR-1 action space is a 384D vector whose first 29 dimensions are active, organized as: left arm (7 joints), right arm (7 joints), left hand (6 joints), right hand (6 joints), and waist (3 joints); the remaining 355 dimensions are zero-padded. All perturbations target specific joint groups within this 29D active subspace while preserving overall trajectory structure and numerical plausibility. Each perturbation uses severity $s = 0.5$ in all experiments.

B.1 PERTURBATION DEFINITIONS

We denote the action at time t restricted to a joint group g as $a_{t,g}$, where $g \in \{\text{L-arm}, \text{R-arm}, \text{L-hand}, \text{R-hand}, \text{L-wrist}, \text{R-wrist}\}$.

1. Grip Force Insufficient (`grip_force_weak`). Left hand joint commands scaled to $(1 - s)$ from $t_0 = 0.40T$ onward. At $s = 0.5$, grip force is halved; the object should slip during transport.

$$a'_{t,\,\text{L-hand}} = (1 - s) \cdot a_{t,\,\text{L-hand}}, \quad t \geq \lfloor 0.40T \rfloor \tag{5}$$

2. Premature Release (`premature_release`). Left hand joints reduced to 2% during carry phase ($0.40T$ to $0.80T$), before reaching placement target. Object should fall mid-transport.

$$a'_{t,\,\text{L-hand}} = 0.02 \cdot a_{t,\,\text{L-hand}}, \quad \lfloor 0.40T \rfloor \leq t \leq \lfloor 0.80T \rfloor \tag{6}$$

3. Grip Carry Slip (`grip_carry_slip`). Left hand timing advanced by $\Delta = \lfloor T(0.15 + 0.20s) \rfloor$ frames; arm trajectory unchanged. Gripper opens before arm reaches target.

$$a'_{t,\,\text{L-hand}} = a_{\min(t+\Delta,\,T-1),\,\text{L-hand}} \tag{7}$$

4. Contact Oscillation (`contact_oscillation`). 3-cycle sinusoidal injection on both left and right arm joints during contact phase ($0.25T$ to $0.70T$), amplitude $A = 0.4 \cdot \text{std}(a_{:,\,\text{L-arm}})$. Prevents stable grasp formation.

$$a'_{t,\,\text{L/R-arm}} = a_{t,\,\text{L/R-arm}} + A \cdot \sin\left(\frac{6\pi(t - t_0)}{t_1 - t_0}\right) \tag{8}$$

5. Wrist Tilt During Grasp (`wrist_tilt_grasp`). Both left and right wrist joints (2 per wrist) offset by $+0.8$ rad from $0.15T$ to $0.85T$, causing incorrect contact geometry.

$$a'_{t,\,\text{L/R-wrist}} = a_{t,\,\text{L/R-wrist}} + 0.8 \tag{9}$$

6. Approach Overshoot (`approach_overshoot`). Left arm joint trajectory scaled $\times 1.30$ during approach ($0.10T$ to $0.75T$); end-effector overshoots object before gripper closes.

$$a'_{t,\,\text{L-arm}} = 1.30 \cdot a_{t,\,\text{L-arm}}, \quad \lfloor 0.10T \rfloor \leq t \leq \lfloor 0.75T \rfloor \tag{10}$$

B.2 PER-TASK PERTURBATION ASSIGNMENT

Each task receives 3 perturbations (plus baseline), with 2 mandatory types applied universally and 1 task-specific type selected based on task affordance:

Table 5: Perturbation schedule per task. All tasks share `grip_force_weak` and `premature_release`; the third varies.

Task	Third Perturbation
gr1_pnp_apple, fold_cloth	`wrist_tilt_grasp`
gr1_pnp_mango, gr1_egodex, pnp_corn, pnp_dragonfruit	`contact_oscillation`
gr1_pnp_pear, pour_items	`approach_overshoot`
pnp_cucumber	`grip_carry_slip`

B.3 TRANSLATION TO NATURAL LANGUAGE (DESCRIPTIVE ACTION MODALITY)

For the text-instruction-conditioned evaluation pipeline, each vector-level perturbation is translated into a natural-language prompt that describes the same failure mode semantically. Prompts are generated by Gemini-2.5-Pro given the task context and perturbation specification, then **manually verified by human annotators** against the corresponding ground-truth video to ensure factual accuracy and physical plausibility.

Each task–episode pair has a `prompts.json` file containing one baseline prompt and one prompt per perturbation condition. Below are representative examples for the `gr1_pnp_apple` task:

Table 6: Example natural-language prompts for the descriptive action modality (task: pick-and-place apple).

Condition	Prompt
Baseline	"From a first-person perspective, picks up a red apple from the center of a wooden table and carefully places it into the bottom shelf of a two-tiered wooden crate on the right."
`grip_force_weak`	"From a first-person perspective, a robot with black hands attempts to pick up a red apple from a wooden table using only 0.1 newton of grip force."
`premature_release`	"From a first-person perspective, a robot with black hands picks up a red apple from a wooden table but releases its grip prematurely while moving it towards the wooden crate on the right, releasing it before reaching the destination."
`wrist_tilt_grasp`	"From a first-person perspective, awkwardly grasps a red apple from a wooden table, its wrist bent sideways at an unnatural angle, before moving the unstably held apple to place it into the bottom shelf of the wooden crate."

Prompt design principles.

1. **Failure mode explicit**: the prompt explicitly states the physical failure (e.g., "0.1 newton of grip force", "releases prematurely") so that any text-conditioned model with adequate language understanding should generate the corresponding failure outcome.
2. **Task context preserved**: the prompt retains full scene description (objects, spatial layout, robot morphology) so the model is not confused about what task is being attempted.
3. **Human-verified**: all prompts are reviewed by annotators who confirm (a) the described failure matches the vector-level perturbation effect, and (b) the scene description matches the first frame the model will receive as conditioning.

B.4 Summary Table

Table 7: Consolidated perturbation reference.

Type	Target Joints	Phase ($\%T$)	Operation	Expected Failure
`grip_force_weak`	L-hand (6D)	40–100	$\times(1-s)$	Object slip
`premature_release`	L-hand (6D)	40–80	$\times 0.02$	Mid-air drop
`grip_carry_slip`	L-hand (6D)	all	shift $+\Delta$	Early release
`contact_oscillation`	L/R-arm (7D+7D)	25–70	$+A\sin(\cdot)$	Unstable contact
`wrist_tilt_grasp`	L/R-wrist (2D+2D)	15–85	$+0.8$ rad	Misaligned grasp
`approach_overshoot`	L-arm (7D)	10–75	$\times 1.30$	Missed object

Design principles. The perturbations satisfy three criteria: (1) *Physical interpretability*: each corresponds to a named failure mode in manipulation literature; (2) *Numerical plausibility*: perturbed vectors remain within or near the normalized $[-1, 1]$ range; (3) *Guaranteed failure*: under correct physics, each perturbation necessarily prevents task completion. A world model that produces success under these conditions is exhibiting optimism bias by definition.

C Natural Language Prompt Generation

MiraBench uses natural-language prompts in two distinct roles: (1) as *evaluation prompts* for VLM-based scoring across all three diagnostic dimensions, and (2) as *conditioning prompts* for the descriptive action modality in action-following and optimism-bias evaluation. Both are generated by Gemini-2.5-Pro and human-verified; their design differs in language, length, and purpose.

C.1 Evaluation Prompts

The VLM evaluators for the three diagnostic dimensions receive sampled video frames alongside structured natural-language queries. Rather than hand-crafting these queries, we adopt a *video-to-prompt inversion* pipeline: Gemini-2.5-Pro watches the video (uniformly sampled to 16 frames) and generates a factual description of the observed content, which then serves as context for the evaluation query.

Generation protocol. Gemini receives a system prompt enforcing strict writing constraints:

- **Positive-only language**: no negations or evaluation-style phrases (e.g., "no penetration", "no deformation").
- **Verb-driven structure**: "who does what to which object, in what direction, along what path."
- **Precise visual grounding**: colors, materials, and spatial relations must be explicitly stated (e.g., "dark green transparent glass mug" rather than "cup").
- **Observable-only**: no subjective intent, no unobservable quantities (forces, internal states).

The output is a prompt in Wan2.1-native style (for physics evaluation on Chinese-prompt models) structured as: scene + subject + object (with color/position) + action (with direction) + static constraints. A companion `negative_prompt` lists both generic quality negatives and video-specific physical violation terms.

Few-shot calibration. Each Gemini call includes 3 randomly sampled seed examples (drawn from 5 human-written references spanning different task categories) to enforce consistent style, length, and structure across the full dataset.

Human verification. All generated prompts are reviewed by annotators who verify (a) factual accuracy against the video content, and (b) absence of evaluation-biasing language. Prompts failing verification are regenerated with additional constraints.

C.2 Conditioning Prompts for Descriptive Action Modality

For the text-instruction-conditioned pipeline, each task–episode pair requires a set of prompts: one baseline prompt describing normal task execution, plus one prompt per perturbation condition describing the corresponding failure mode. These prompts serve as the model's *input* (conditioning signal), not as evaluation queries.

Baseline prompts. Generated by Gemini-2.5-Pro in English, describing what the robot does in the ground-truth video from a first-person perspective. Structure: viewpoint + robot description + object identification + action sequence + placement target. Example: "From a first-person perspective, picks up a red apple from the center of a wooden table and carefully places it into the bottom shelf of a two-tiered wooden crate on the right."

Perturbation prompts. Each perturbation type is translated from its vector-level specification into a semantically equivalent natural-language description. The prompt explicitly states the failure mechanism while preserving full scene context:

- `grip_force_weak` → "... attempts to pick up [object] using only 0.1 newton of grip force."
- `premature_release` → "... picks up [object] but releases its grip prematurely ... releasing it before reaching the destination."
- `grip_carry_slip` → "... the [object] slips in its grasp ..."
- `contact_oscillation` → "... its fingers oscillate and tremble with unsteady pressure before attempting to lift ..."
- `wrist_tilt_grasp` → "... its wrist bent sideways at an unnatural angle ..."
- `approach_overshoot` → "... overshoots past the [object] before correcting its position to grasp it ..."

Table 8: Definitions and scoring summary for the seven main metrics. All model-level scores use a 0–100 scale.

Metric	Definition and meaning	Computation
Obj./Occ.	Object and occlusion consistency: whether objects remain coherent and retain their identity through occlusion.	InternVL3-78B judges 10 midcut pairs from 20 sampled frames; each inconsistent pair subtracts 10 points from the respective score.
Rule	Physical rule adherence: whether object motion satisfies kinematic and contact constraints.	10% VLM motion-quality gate and 90% SAM2.1 trajectory and kinematic scoring.
TCR	Task Completion Rate: whether the nominal rollout completes the intended task.	Binary whole-video judgment from 16 frames, averaged over nominal episodes.
OPS	Object Preservation Score: whether relevant objects remain intact and grounded relative to the reference.	Binary judgments on 16 prediction/reference frame pairs; an episode passes at ≥ 0.70 confidence.
Gen.	Generalization: whether task completion transfers beyond the GR-1 setting.	Exponential penalty on the GR-1/OOD TCR gap, capped at 100.
Bias Res.	Bias Resistance: whether nominal and perturbed rollouts preserve distinct outcomes.	Late-phase object-state comparisons; a majority of "Same" votes flags optimism bias. The ungated score is 100 minus the bias rate.

Generation and verification. Perturbation prompts are generated by Gemini-2.5-Pro given: (1) the baseline prompt, (2) the perturbation type name and its physical description, and (3) the first frame of the episode for visual grounding. All generated prompts are then **manually reviewed by human annotators** who verify:

1. The described failure mode is physically consistent with the vector-level perturbation (e.g., "0.1 newton" correctly reflects the $\times 0.5$ grip force reduction).
2. The scene description (objects, colors, spatial layout) matches the first frame the model will receive.
3. The prompt does not inadvertently reveal evaluation criteria or contain ambiguous language.

Cross-task consistency. Prompts across all 9 tasks follow a uniform structure (viewpoint + agent + failure description + object + intended action), ensuring that differences in model performance across tasks reflect task difficulty rather than prompt quality variation. The full set of prompts is released as part of the benchmark data.

D Framework Scoring

Each MiraBench diagnostic dimension employs a dedicated scoring pipeline that combines a VLM-based perceptual judge with task-specific inference strategies. The pipelines share a common design principle: rather than fine-tuning a domain-specific model, we leverage a frontier-scale VLM as a zero-shot judge and stabilize its outputs through structured prompting and task-specific aggregation. This appendix documents the complete scoring logic, prompt templates, and validation results for each diagnostic dimension.

D.1 Physics Adherence

Physics Adherence provides context for interpreting action-following and optimism-bias scores, but is scored independently of them. It has two components: *Physical Consistency Detection* (behavioural coherence; Section D.1.1) and *Physics-Law Compliance* (quantitative kinematic validity under a single conservative force; Section D.1.2). Both operate *reference-free*: the score depends only on the predicted rollout $\hat{v}$, since physical laws hold unconditionally and the metrics must be evaluable on counterfactual action sequences for which no reference exists.

D.1.1 PHYSICAL CONSISTENCY DETECTION

We measure behavioural-level physical consistency through two orthogonal indicators, both implemented as zero-shot pairframe binary judgements on InternVL3-78B (Zhu et al., 2025b):

- **D1 Object Consistency.** The manipulated object should preserve plausible shape, material, size, and colour during continuous visibility (Zheng et al., 2025).
- **D2 Occlusion Consistency.** When the object is occluded and reappears, it should remain the same object.

The two are by design orthogonal — object physics vs. object identity. The combined **PCS** score is the equal-weight mean of the two sub-scores. Since both indicators share the same frame-extraction front-end, their N/A flags are perfectly correlated; PCS is therefore either a real number in $[0, 1]$ or `None` when both indicators abstain.

Pipeline (shared by D1 and D2). We extract 20 evenly-spaced frames and form 10 *midcut* pairs (frame i paired with frame $i + 10$, $i = 0, \ldots, 9$). Each pair is vertically stacked into a single image and submitted to InternVL3-78B, which returns a binary verdict on that pair (`A`=consistent, `B`=inconsistent). The video-level score on a 0–100 scale rewards each pair the VLM judges consistent:

$$s_d = 100 \cdot \left(1 - \tfrac{1}{n} \sum_{i=1}^{n} \mathbb{1}[\text{vote}_i = \texttt{B}]\right) \in [0, 100], \qquad n = 10, \tag{11}$$

i.e. each `B` vote subtracts $100/n = 10$ points from a perfect score of 100. A categorical label `B`, set whenever any pair is judged `B`, is also reported but does *not* enter the PCS aggregation.

D1: Object Consistency prompt.

> These two frames are from a robot manipulation video. Frame 1 is on top, Frame 2 is on bottom.
>
> Identify the MAIN OBJECT — the item being grasped or moved by the robot arm (not the arm itself).
>
> Carefully compare the main object between the two frames across ALL of the following:
>
> - **SHAPE:** Does it keep the same overall form? Any unexpected deformation, collapse, bending, or structural change → inconsistent.
> - **MATERIAL:** Does it behave like the same material type? A rigid object that looks soft / rubbery, or a flexible object that looks stiff → inconsistent. Even subtle changes in how the object deforms or holds its shape count.
> - **SIZE:** Is it approximately the same size? Any unexplained sudden change in scale → inconsistent.
> - **COLOUR:** Is the dominant colour stable? Sudden hue change unexplained by lighting → inconsistent.
>
> Judge whether the object remains PHYSICALLY PLAUSIBLE across the two frames:
>
> - **A. Consistent:** shape, material, size, and colour are stable — differences are clearly explained by viewpoint, grip angle, or distance only.
> - **B. Inconsistent:** something physically implausible changed. **Do NOT default to A** — if you notice any suspicious change that cannot be explained by viewpoint alone, choose B.
>
> Output only `A` or `B` on the first line, then one sentence describing what you observed.

D2: Occlusion Consistency prompt. The Occlusion Consistency indicator uses the same pipeline as D1 but a different prompt that explicitly handles occlusion and asks the VLM to judge *identity* rather than physical attributes. Instructing the VLM to reason over the visible portions rather than abstaining when one frame is partially hidden removes the up-front "occlusion-event present?" gate that earlier two-stage partial-vs-full designs (Zheng et al., 2025) required and reduces the N/A rate from $\sim$70% to $<$5%.

Table 9: PCS(Physical Consistency Score) per-indicator alignment with human annotations on 128 videos. Pairwise accuracy is calculated via consistency matching between model outputs and human judgements.

Indicator	Backbone	Agreement (%)	95% bootstrap CI
Obj. Cons.	InternVL3-78B	87.3	[77.8, 95.0]
Occ. Cons.	InternVL3-78B	85.1	[77.9, 91.3]

> These two frames are from a robot manipulation video. Frame 1 is on top, Frame 2 is on bottom.
>
> **Context:** one or both frames may involve occlusion — the object may be partially or fully hidden by the robot arm or another object.
>
> Focus on the main object (the item being grasped or moved, not the arm itself). If one frame shows it partially hidden, judge only the visible portions.
>
> Judge whether the object looks like the SAME object in both frames — same identity, colour, shape, and appearance:
>
> - **A. Consistent:** the visible portions look the same — any differences are explained by occlusion angle, viewpoint, or grip only.
> - **B. Inconsistent:** the visible portions look genuinely different in a way that occlusion alone cannot explain — changed colour, shape, texture, or appears to be a different object.
> - Do NOT default to A. If you are unsure, choose B.
> - Err on the side of flagging inconsistency.
>
> Output only `A` or `B` on the first line, then one sentence about what you observed.

Combined PCS score.

$$\mathrm{PCS}(\hat{v}) = \tfrac{1}{2}\big(s_{\text{obj}}(\hat{v}) + s_{\text{occ}}(\hat{v})\big) \;\in\; [0, 100], \qquad s_d \in [0, 100]. \tag{12}$$

Equal weighting reflects the orthogonality of the two properties. Table 9 reports per-indicator pairwise accuracy against human annotations on the 68-pair occ68 head-to-head set.

Key design decisions for Physical Consistency. (1) Orthogonal two-dimensional decomposition. Decomposing "physical consistency" into object physics and object identity lets us trace any low PCS score back to a specific failure dimension, each individually verifiable in our released human-annotation corpus, and maps cleanly onto cognitive-science principles — object permanence under occlusion is grounded in infant-physics studies. **(2) Pairframe binary judgement.** Earlier four-tier (A/B/C/D) VLM indicators (Zheng et al., 2025) disproportionately answer "A" on diffusion content, collapsing four-tier signal to binary anyway. We use binary judgement directly with an adversarial "do not default to A" prompt, giving a cleaner threshold and lower agreement noise. **(3) Midcut pairing with threshold 1.** Random pairing biases toward A by leaving most pairs at adjacent frames where the object has barely moved; midcut spans half the video duration to expose accumulated changes. Threshold $= 1$ (any inconsistent pair flags the whole video) reflects the strict requirement that any local violation invalidates the entire generation.

D.1.2 Physics-Law Compliance

Where Physical Consistency tests behavioural coherence, Physics-Law Compliance stress-tests *quantitative kinematic compliance* under a single conservative force — gravity (free fall) or friction (horizontal push). This component uses an explicit kinematic-computation pipeline: *no language-model judgement enters the kinematic score*. A 10% Video Quality Score (VQS) contribution gates whether the video's physical premise (any object exhibits a clear translational motion) is satisfied at all.

Task definition. Two motion regimes are dispatched automatically: **vertical free-fall** ($a = g$, $y(t) = \frac{1}{2}gt^2 + v_0 t + y_0$) and **horizontal push** ($a = -\mu g$, $x(t) = -\frac{1}{2}\mu g\, t^2 + v_0 t + x_0$). Expected motion is therefore quadratic on the appropriate axis; the evaluator scores how well each scoring segment fits this quadratic form plus three discrete event features.

Pipeline. **Stage 0 (Video Quality Score).** A VLM inspects six evenly-spaced frames horizontally tiled into a single image and answers two binary questions in JSON: `video_ok` (recognisable robot scene) and `has_motion` (any object exhibits a clear translational motion — free-fall, slide, push, roll, etc. — at some point in the clip). The two answers map to a discrete VQS $\in \{0, 5, 10\}$ (both False / video_ok only / both True), reported on the same 0–100 scale as the final PhysLawScore. When VQS$= 0$ or VQS$= 5$, the kinematic component is forced to 0; only when VQS$= 10$ does the kinematic score enter the final aggregation. This prevents SAM2 + polyfit from producing spuriously high scores on videos where no scorable motion actually occurs (e.g. a static held object yields a degenerate but smooth trajectory). Because the gate asks about *any* translational motion — not free-fall specifically — the same Stage 0 check applies uniformly to both the vertical (free-fall) and horizontal (push / slide) sub-pipelines. **Stages 1–5 (kinematic).** A VLM (or any open-vocabulary detector) returns the object bounding box in frame 0; SAM2.1 forward propagation yields a mask-centroid trajectory; an axis classifier dispatches to vertical / horizontal / mixed; the active axis is segmented into fall / rise / rest (vertical) or push / slide / rest (horizontal); each scorable segment is fit with a degree-2 polynomial.

Per-segment scoring. Each scorable segment yields a fitted acceleration $\hat{a}$. The segment score is the product of three independent physical-validity factors, each in $[0, 1]$:

$$\text{seg_score} = \text{sign_ok} \cdot \text{magnitude_ok} \cdot \text{uniformity_ok}. \tag{13}$$

sign_ok verifies that the conservative force has the correct direction (positive $\hat{a}$ for fall, negative for slide). magnitude_ok is 1.0 when $|\hat{a}|$ is within $[0.3, 3] \times \frac{2|\Delta y|}{\Delta t^2}$ of the kinematic expectation, with linear decay outside this range. uniformity_ok halves the segment, fits each half, and computes the coefficient of variation between the two fitted accelerations: $\text{half_cv} \leq 0.15 \rightarrow 1.0$, with linear decay to 0.0 at $\text{half_cv} = 0.80$. The polynomial R^2 is recorded as a diagnostic but does *not* enter the score.

The trajectory-level curve score is the length-weighted mean of segment scores with a coverage factor and a hard-violation override:

$$\text{curve} = \begin{cases} 0 & \text{if any fall/rise segment has } \text{sign_ok} = 0 \\ \text{cov} \cdot \dfrac{\sum_s n_s \cdot \text{seg_score}_s}{\sum_s n_s} & \text{otherwise,} \end{cases} \tag{14}$$

where $\text{cov} = \min(1, \text{coverage}/0.3)$.

Event features and gated fusion. Four discrete features — velocity drop at impact $(v_{\text{before}} - v_{\text{after}})/v_{\text{before}}$, post-landing drift $\Delta y_{\text{post}}/0.10$, a binary has_impact indicator, and bounce decay h_2/h_1 (physical < 1) — are combined with weights $(0.30, 0.20, 0.30, 0.20)$ to yield event. The kinematic score gates curve against event:

$$\text{kinematic_score} = \begin{cases} 0.30 \cdot \text{curve} + 0.70 \cdot \text{event} & \text{if curve} \geq 0.30 \\ 0.70 \cdot \text{curve} + 0.30 \cdot \text{event} & \text{if curve} < 0.30, \end{cases} \tag{15}$$

preferring curve when it has flagged a clear violation and event otherwise — avoiding the case where a noise-sensitive curve fit drags down a video whose event-level physics is in fact correct.

Final score. On the 0–100 scale, the final PhysLaw score combines the kinematic component (on $[0, 1]$) with VQS:

$$\text{PhysLaw}(\hat{v}) \;=\; 0.9 \cdot \text{effective_physics} + \text{VQS}, \tag{16}$$

where

$$\text{effective_physics} = \begin{cases} 100 \cdot \text{kinematic_score} & \text{if } \texttt{has_motion} = \text{True}, \\ 0 & \text{otherwise.} \end{cases} \tag{17}$$

A video the VLM judges to lack any scorable motion therefore cannot exceed $5/100$, regardless of trajectory smoothness.

Physics-Law VQS prompt.

> You are evaluating a generated video that should show a physics-scorable scene (free-fall, slide / push along a surface, or any other clear, sustained translational motion). The image shows several equally-spaced frames tiled left to right. Answer ONLY with valid JSON:
>
> ```
> { "video_ok": true/false,
>   "has_motion": true/false,
>   "reason": "one short sentence" }
> ```
>
> **video_ok:** the video shows a recognisable robot-arm scene (not all-black, not corrupted, not static). **has_motion:** ANY object visible in the scene undergoes a clear translational motion at some point — free-fall, sliding, being pushed, rolling, or otherwise traversing space. Be LENIENT; set to `false` ONLY when the entire clip is essentially static.

Test-set performance and key design decisions for Physics-Law Compliance. On the 89 graded videos in the released Physics-Law set, PhysLaw matches well with human Grade A/B/C/D; per-object breakdown in Section H.2. **(1) Explicit kinematic computation, no LLM as judge:** the 90% component is a quadratic fit and three explicit factors — an auditor can trace any low score back to a specific segment, axis, and factor. **(2) VLM as quality gate, not as judge:** a pure SAM2 + polyfit pipeline can score highly on videos where no scorable motion occurs (a static held object yields a degenerate but smooth trajectory); the Stage 0 VQS detects this premise mismatch and caps Physics Law at $5/100$ irrespective of trajectory smoothness. **(3) Multiplicative factor decomposition:** $\mathrm{seg_score}$ is the product of three independent checks rather than a single R^2, which would conflate three distinct failure modes (anti-gravity, wrong magnitude, non-uniform acceleration). **(4) Length-weighted segment averaging instead of** $\min$**:** a naive $\min$ lets a short, noise-sensitive rebound segment force the trajectory to 0; the length-weighted mean is robust while still flagging hard violations through the explicit "any sign_ok = 0 $\rightarrow$ 0" override.

D.2 Action Following

D.2.1 Task Completion Rate (TCR)

TCR is a binary per-episode metric that asks whether the world model's predicted video demonstrates successful completion of the manipulation task.

Task definition. Given a predicted video $\hat{\mathbf{v}}$ and the corresponding natural-language task instruction ℓ, determine whether the task goal is achieved in $\hat{\mathbf{v}}$ (TCR = 1) or not (TCR = 0).

Model. We use **InternVL3-78B** as the automated judge, deployed across 3 GPUs with tensor-parallel split. No fine-tuning is applied; the model is used zero-shot with a task-specific prompt.

Inference: 16-frame whole-sequence judge. We uniformly sample 16 frames from the predicted video and present all 16 frames to the VLM in a single call, along with the task instruction. The VLM outputs a single binary judgment for the entire video—no per-frame voting, no aggregation:

$$\text{TCR} = v \in \{0, 1\} \tag{18}$$

This holistic approach lets the VLM reason over the full temporal context, catching cases where task completion occurs mid-video or where motion progress across frames disambiguates single-frame uncertainty.

Importantly, **no ground-truth frames are shown to the judge**. Removing the GT reference eliminates a systematic bias where the VLM anchors on arm-pose similarity between predicted and ground-truth frames, which unfairly penalises models whose motion trajectories diverge from the GT while still achieving the task goal.

Prompt design.

```
You are evaluating a robot manipulation video.
Here are 16 frames sampled uniformly from the predicted video, in
chronological order.

Task instruction:  "{instruction}"

Looking at the entire video:  did the robot complete the task
instruction?
• Track object positions across the full sequence to confirm goal
achievement.
• Do NOT require any specific arm pose -- focus on whether the
target object reaches the correct final state.

Respond ONLY with 0 or 1.
1 = task completed in this video
0 = task not completed
```

Model-level score. The TCR score for a model on a given split is the fraction of episodes with TCR = 1, reported as a percentage:

$$\text{TCR score} = \frac{1}{N}\sum_{n=1}^{N}\text{TCR}_n \times 100 \tag{19}$$

Key design decisions.

1. **Whole-sequence over per-frame**: Presenting all 16 frames at once allows the VLM to reason about temporal progression—e.g. recognising that an object was successfully placed mid-video even if the arm subsequently moves away. Per-frame voting schemes risk losing this context and introduce sensitivity to aggregation hyper-parameters (tail weighting, vote threshold).
2. **No ground-truth reference**: Showing GT frames alongside predicted frames creates an implicit arm-pose anchor that biases the judge against models whose trajectories differ from GT, even when the task goal is achieved. Removing GT frames eliminates this bias and evaluates the predicted video on its own merits.
3. **Single binary output**: A straightforward 0/1 judgment avoids the need for confidence calibration, vote aggregation, or threshold tuning, making the metric simple and reproducible.

Human consistency validation (GR1 split, 48 episodes). We validate the automated evaluator against human-verified labels on 48 GR1 episodes. The evaluator achieves an accuracy of **87.5%** (42/48; 95% bootstrap CI [77.1, 95.8]) and a Spearman rank correlation of $\boldsymbol{\rho = 0.545}$ ($p < 0.001$) between the per-episode confidence score and the ground-truth binary label.

D.2.2 Object Preservation Score (OPS)

OPS is a binary per-episode metric that assesses whether the world model maintains physically plausible object appearance throughout the predicted video, independent of whether the task is completed.

Task definition. Given a predicted video $\hat{\mathbf{v}}$ and the corresponding ground-truth video $\mathbf{v}^*$, along with the task instruction ℓ, determine whether the objects in $\hat{\mathbf{v}}$ are visually coherent throughout the sequence (OPS = *preserved*) or exhibit artefacts such as deformation, disappearance, or unnatural pop-in/pop-out events (OPS = *flawed*).

Model. We use **InternVL3-78B** as the automated judge, deployed across 3 GPUs with tensor-parallel split. No fine-tuning is applied; the model is used zero-shot with a task-specific prompt.

Inference: 16-frame uniform sampling with mean aggregation. We uniformly sample 16 frames from both the predicted and ground-truth video, producing 16 temporally aligned frame pairs. Each pair is independently judged by the VLM, yielding a binary quality vote $q_i \in \{0, 1\}$.

Unlike TCR, temporal position is not expected to affect object quality, so votes are aggregated by simple mean:

$$\text{confidence} = \frac{1}{16}\sum_{i=1}^{16} q_i \tag{20}$$

The per-episode label is then:

$$\text{OPS} = \begin{cases} \textit{preserved} & \text{if confidence} \geq 0.70 \\ \textit{flawed} & \text{if confidence} < 0.70 \end{cases} \tag{21}$$

Prompt design.

```
You are evaluating the visual quality of a single predicted robot
video frame.
• Frame1 = frame from the PREDICTED video (world model output --
evaluate this)
• Frame2 = frame from the GROUND TRUTH video at the same timestamp
(reference only)
Task instruction:  "{instruction}"
Check Frame1 against Frame2 for the following issues:
1.  Is the target object (the object being manipulated) clearly
visible in Frame1,
without unexpected blurring, occlusion, or disappearance?
2.  Are all objects in Frame1 free of distortion or unnatural
deformation?
3.  Are there no objects that pop in or pop out unnaturally between
frames
(appearing or vanishing without physical cause)?
Respond ONLY with 0 or 1.  No explanation.
1 = Frame1 passes all checks (high quality, matches GT object
presence)
0 = Frame1 fails at least one check (object issue, distortion, or
pop artifact)
```

Model-level score. The OPS score for a model on a given split is the fraction of *preserved* episodes:

$$\text{OPS score} = \frac{|\{n : \text{OPS}_n = \textit{preserved}\}|}{N} \times 100 \tag{22}$$

Human consistency validation (GR1 split, 26 human-annotated episodes). We validate the automated evaluator against human annotations using the binary *preserved* vs. *flawed* split. The evaluator achieves a **92.3%** accuracy (24 of 26 episodes agree with human judgment; 95% bootstrap CI [80.8, 100.0]), confirming that the 0.70 threshold reliably separates episodes that humans judge as object-coherent from those they do not.

Key design decisions.

1. **Uniform temporal weighting**: Object preservation is a frame-level property that should hold throughout the video, not just at the end. Simple mean aggregation avoids biasing the score towards any particular phase of the manipulation.
2. **Binary label with conservative threshold**: A single *preserved*/*flawed* dichotomy reflects the fact that object artefacts are both highly salient and disqualifying for downstream use. The 0.70 confidence threshold requires a clear majority of frames to pass, catching videos where artefacts appear intermittently.
3. **Object-focused prompt with three explicit checks**: Decomposing the quality judgment into three distinct failure modes (visibility, deformation, pop artefacts) guides the VLM towards consistent and interpretable binary decisions, reducing false negatives that arise from a holistic "looks bad" judgment.

D.2.3 GENERALIZABILITY (GEN)

GEN measures how well a model generalises from its training split to unseen episodes, quantifying the gap between in-distribution and out-of-distribution TCR performance.

Definition. Let TCR_{GR1} and TCR_{Gen} denote a model's Task Completion Rate on the GR1 (in-distribution) and generalizability (out-of-distribution) splits, respectively. The raw generalisation gap is:

$$\Delta = \text{TCR}_{\text{GR1}} - \text{TCR}_{\text{Gen}} \tag{23}$$

A positive Δ indicates overfitting to the training distribution. To convert this gap into a score where higher is better and the range is bounded on $[0, 100]$, we apply an exponential decay with clipping:

$$\text{GEN} = \min\Big(100,\ 100 \cdot e^{-\Delta/100}\Big) \tag{24}$$

Interpretation.

- $\Delta \leq 0$ (no overfitting): GEN ≥ 100, clipped to 100%. The model generalises at least as well as it performs in-distribution.
- $\Delta > 0$ (overfitting): GEN decays exponentially, penalising larger gaps while remaining non-zero. For example, $\Delta = 2.7$ yields GEN $= 97.4\%$; $\Delta = 85.3$ yields GEN $= 42.6\%$.

Why exponential rather than linear? A linear transform $(100 - \Delta)$ would assign GEN $= 0$ for $\Delta \geq 100$, creating a hard floor. The exponential decay ensures that even severely overfit models retain a non-zero score proportional to their gap, while small gaps remain close to 100%. The divisor of 100 scales the decay so that a 10-point gap causes only $\sim$10% loss, matching the intuition that moderate overfitting should not be penalised as severely as extreme overfitting.

D.3 OPTIMISM BIAS DETECTION

Optimism Bias Detection uses a separate evaluation pipeline designed specifically for the binary optimism-bias judgment.

Task definition. Given a baseline prediction $\hat{\mathbf{v}}^+$ (under nominal action) and a perturbed prediction $\hat{\mathbf{v}}^-$ (under failure-inducing action), determine whether the world model ignores the perturbation (Y = optimism bias present) or correctly reflects it (N = no bias).

Model. We use **InternVL3-78B** as the automated judge, with dynamic resolution (no distortion) at native video resolution. No fine-tuning is applied; the model is used zero-shot with a task-specific prompt.

Inference: 7-frame majority voting. Rather than processing the full video, we extract 7 frames at progress percentages $[81\%, 83\%, 85\%, 87\%, 90\%, 95\%, 97\%]$, focusing on the late phase of the manipulation where perturbation effects are most visible. Each frame is independently judged as "Same" or "Different" by the VLM. The final prediction is determined by majority vote:

$$\text{pred} = \begin{cases} \text{Y (bias)} & \text{if } \sum_{i=1}^{7} \mathbb{1}[\text{vote}_i = \text{Same}] > 3 \\ \text{N (no bias)} & \text{otherwise} \end{cases} \tag{25}$$

Input format. For action-conditioned models (DreamDojo), baseline and perturbed frames are presented side-by-side in a two-column layout (left = baseline, right = perturbed). For text-conditioned models with different rendering styles (Happy Horse), original high-resolution separate videos are concatenated to avoid confounding visual style differences with perturbation effects.

Prompt design. Two prompt variants are used depending on the model type:

```
This is one frame from a robot manipulation video.
LEFT column = baseline prediction.  RIGHT column = perturbed
prediction.

Ignore blur, color, or rendering differences.
Look at the manipulated object only.

Is the object in the same general location/state in LEFT and RIGHT?
(Same container/area = Same.  Dropped/missing/wrong place =
Different.)

Output ONLY: Same or Different
```

Lenient Prompt (text-conditioned models)

```
This is one frame from a robot manipulation video.
LEFT = baseline.  RIGHT = perturbed version.

Ignore all visual quality differences:  blur, color, lighting,
noise, rendering style.

Compare ONLY the general action trend:
- Is the robot doing roughly the same thing in both?
- Is the object in a roughly similar situation?

Be lenient:  minor differences in exact position, timing, or
appearance do NOT matter.
Only say Different if the action is fundamentally different.

Output ONLY: Same or Different
```

The lenient prompt is necessary for text-conditioned models (Happy Horse) because their rendering style differs substantially from action-conditioned baselines, creating visual differences unrelated to the perturbation.

Test-set performance (Y/N only, excluding borderline Y?). Table 10 reports per-model accuracy against human annotations. Figure 6 visualizes the per-model breakdown and voting distribution.

Table 10: Optimism bias evaluator accuracy on human-annotated test set (Y and N labels only; Y? borderline cases excluded).

Model	Prompt	Accuracy	Y Recall	N Recall
DreamDojo-14B	Standard	87.1%	88.8%	80.0%
DreamDojo-2B	Standard	81.5%	78.3%	100.0%
Happy Horse	Lenient	75.0%	33.3%	100.0%
Wan2.1	Standard	100.0%	–	100.0%
DreamDojo-14B (Lingchu)	Standard	100.0%	100.0%	100.0%
Overall		**87.8%**	**87.1%**	**89.6%**

Cross-metric agreement and uncertainty. Table 11 consolidates the per-metric VLM–human agreement results. Each interval is a 95% bootstrap confidence interval computed from 200,000 resamples; the reported intervals are asymmetric because agreement is bounded in $[0, 100]$. These values quantify agreement with human labels and should not be interpreted as inter-annotator agreement.

Human grounding and validation coverage. The automated evaluators are zero-shot, structured VLM judges grounded in human labels rather than preferences from a single model family. The validation coverage includes vector-conditioned open-weight models at two scales (DreamDojo-2B and DreamDojo-14B), a text-conditioned open-weight model (WAN2.1), a closed-source text-conditioned model (Happy Horse), and a DreamDojo-14B/Lingchu subset for task-domain transfer. This covers conditioning interface, access type, model scale, and task transfer; the representative model-group agreement values are summarized in Table 12. The evaluator uses structured temporal evidence: physical consistency uses 10 midcut pairs from 20 frames, task completion uses

Table 11: Per-metric agreement between the automated VLM evaluators and human labels. The 95% confidence intervals use 200,000 bootstrap resamples.

Metric	Agreement (%)	95% bootstrap CI (%)
Obj. Consistency	87.3	[77.8, 95.0]
Occ. Consistency	85.1	[77.9, 91.3]
Task Completion (TCR)	87.5	[77.1, 95.8]
Object Preservation (OPS)	92.3	[80.8, 100.0]
Optimism Bias	87.8	[84.0, 91.5]

Table 12: Representative model-group stratification of VLM–human agreement. This is a descriptive stratification of the validation set, not a leave-one-model-family-out estimate.

Validation group	Interface / access	Agreement (%)
DreamDojo-14B	Vector / open	89.7
DreamDojo-2B	Vector / open	87.7
Happy Horse	Text / closed	87.3
WAN2.1	Text / open	96.2

a 16-frame whole-sequence judgment, object preservation averages 16 frame-pair judgments, and optimism-bias detection uses a seven-frame late-phase majority vote. Trained platform reviewers manually inspected representative evaluator decisions and confirmed that they followed the intended observable criteria, including object-state change, physically plausible action outcomes, and whether perturbed predictions collapsed into nominal success. The full annotation coverage, quality-control procedure, and release format are documented in Appendix G.5.

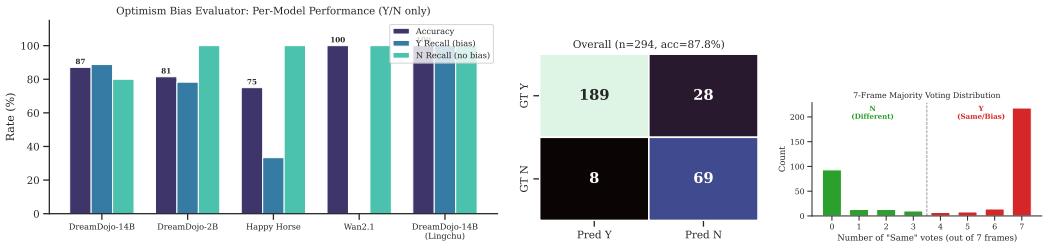


Figure 6: Optimism bias evaluator performance. **Left**: Per-model accuracy, Y recall, and N recall. **Center**: Overall confusion matrix ($n = 294$, accuracy 87.8%). **Right**: Distribution of "Same" vote counts across all 376 samples; the bimodal pattern confirms that most samples produce clear majority decisions, supporting the 7-frame voting design.

Key design decisions.

1. **Late-phase frame sampling**: Frames are extracted at 81–97% progress because perturbation effects (e.g., premature release, grip slip) manifest in the transport and placement phases, not the approach phase.
2. **Object-focused prompt**: The prompt explicitly directs the model to focus on the manipulated object's state rather than overall visual similarity, reducing false positives from rendering differences.
3. **Separate prompt for text-conditioned models**: Text-conditioned models produce stylistically different videos even without perturbation, requiring the lenient prompt to avoid conflating style differences with perturbation effects.
4. **Zero-shot 78B model vs fine-tuned 8B**: We find that a large zero-shot model (InternVL3-78B) outperforms the fine-tuned smaller model (Qwen3-VL-8B SFT) on this binary task, likely because the task requires spatial comparison rather than domain-specific scoring.

E PHYSICAL CONSISTENCY METRIC CALIBRATION AND PHYSICS LAW COMPLIANCE DETAILS

This appendix gathers the calibration constants, candidate-selection process, and validation evidence for the two evaluators released with MiraBench: the *physical-consistency* evaluator and the *physics-law* evaluator. Every numeric threshold and weight cited below is exposed as a named constant in the released source.

E.1 PHYSICAL CONSISTENCY EVALUATOR (PCS)

The released physical-consistency evaluator scores each video on two orthogonal dimensions and aggregates them by an equal-weight mean. No four-tier (A/B/C/D) sub-indicator weighting and no calibration-fitted coefficients enter the released formula; an earlier 9-indicator $\mathrm{PhysConsScore}$ aggregator is retained in the codebase as an ablation baseline only and is not used in the headline numbers.

D1 — Object Consistency. A midcut paired-frame binary judgement on InternVL3-78B. We extract 20 evenly-spaced frames, form 10 pairs by midcut (frame i with frame $i{+}10$), vertically stack each pair, and submit each composite to the VLM with an adversarial prompt that explicitly discourages defaulting to the "A" (consistent) label and asks the VLM to compare shape, material, size, and colour. Let b_{obj} be the number of pairs voted "B" (inconsistent); the dimension score is

$$s_{\mathrm{obj}} \;=\; 1 \;-\; b_{\mathrm{obj}}/10, \qquad b_{\mathrm{obj}} \in \{0, 1, \ldots, 10\}. \tag{26}$$

The violation threshold of 1 (any single "B" flags the video) is intentional: a single clearly-inconsistent pair is sufficient evidence of physical violation in the subject.

D2 — Occlusion Consistency. The same paired-frame pipeline as D1, but with a prompt that asks the VLM to judge *identity* (rather than physical attributes) and to explicitly handle partial-visibility cases. The unified pairframe formulation, in which the VLM itself adjudicates whether visibility is sufficient for a verdict, reduces the N/A rate from $\sim 70\%$ in earlier two-stage partial-vs-full designs to below 5% on the validation set:

$$s_{\mathrm{occ}} \;=\; 1 \;-\; b_{\mathrm{occ}}/10. \tag{27}$$

Aggregation. Both indicators share the same frame-extraction front-end, so their N/A flags are perfectly correlated; the PCS score is therefore either a real number in $[0, 1]$ or `None`:

$$\mathrm{PCS} \;=\; \tfrac{1}{2}\big(s_{\mathrm{obj}} + s_{\mathrm{occ}}\big). \tag{28}$$

Backbones and inference. D1 and D2 share a single InternVL3-78B instance loaded once per evaluation run; no fine-tuning is applied. Each pair is rendered as a vertical stack of two frames at $448{\times}448$ and submitted as a single image with the corresponding text prompt. Trials whose response cannot be parsed are discarded.

Table 13: Hyperparameters of the released PCS physical-consistency evaluator.

Symbol	Meaning	Value
N_{pair}	Paired frames per video	10 (from 20 evenly-spaced)
τ_B	Violation threshold (# B votes)	1
T_{vlm}	VLM call timeout (s)	90

E.2 PHYSICS LAW COMPLIANCE DETAILS

The physics-law evaluator scores each video by an explicit kinematic fit on a single tracked object trajectory, with a single VLM call used as a video-quality gate before the kinematic pipeline runs. No language model is used inside the kinematic computation; the only VLM call (besides the optional first-frame object localizer) is the Stage 0 video-quality check, which contributes 10% of the final score.

Stage 0: Video Quality Score (VQS). Six evenly-spaced frames are tiled horizontally and submitted to a VLM that returns two binary fields in JSON: `video_ok` (the video shows a recognisable robot-arm scene) and `has_motion` (any object exhibits a clear translational motion — free-fall, slide, push, roll, etc. — at any point in the video). The two answers map to a discrete VQS:

$$\text{VQS} = \begin{cases} 0 & \text{if } \texttt{video_ok} = \text{False}, \\ 5 & \text{if } \texttt{video_ok} = \text{True} \wedge \texttt{has_motion} = \text{False}, \\ 10 & \text{if both True.} \end{cases} \tag{29}$$

The kinematic pipeline contributes only when $\text{VQS} = 10$; otherwise $\text{effective_physics} = 0$ and the final score is set by VQS alone, capping a non-motion video at $5/100$. This prevents the SAM2 + polyfit downstream from producing spuriously high scores on videos where no scorable motion actually occurs. The same gate covers both the vertical free-fall and horizontal push / slide sub-pipelines.

Tracking and dispatch. A SAM2.1 (Hiera-Large) mask is propagated forward from a first-frame prompt, yielding a centroid trajectory $\{(x_t, y_t)\}_{t=0}^{T-1}$ in normalised image coordinates. With $\Delta x = \max_t x_t - \min_t x_t$ and analogously for Δy, the trajectory is routed to the gravity (vertical) pipeline if $\Delta y > 1.5\,\Delta x$ and $\Delta y > 0.05$, to the friction (horizontal) pipeline if $\Delta x > 1.5\,\Delta y$ and $\Delta x > 0.05$, and to both pipelines otherwise.

Coverage analysis. On a 50-video free-fall validation set, the curve branch alone (segment polynomial fits) is evaluable on $32/50$ videos; introducing the event branch raises overall coverage to $48/50$, since `has_impact` only requires a clean stop in the trajectory and does not need a fittable fall segment. The remaining 2 unevaluable videos lack both a fittable trajectory and a detectable impact frame.

Trajectory segmentation. Frame-to-frame velocities are tagged `move` or `rest` using the threshold $\tau_v = \max(0.08,\, 0.15\,|v|_{p95})$. After merging consecutive same-label runs, sign flips that survive a two-frame confirmation window split each `move` block into sign-labelled sub-segments: `lift / fall / rise / rest` on the vertical axis (`lift` is the externally driven phase before the first fall and is excluded from scoring), `push / slide / rest` on the horizontal axis (`push` is excluded). Segments shorter than four points or with span below 0.03 on the relevant axis are demoted to `rest`. Pseudocode:

```
def segment(centroids):
    v = diff(y) / diff(t)
    v_ref = percentile(|v|, 95)
    move_thresh = max(0.08, 0.15 * v_ref)
    min_v       = max(0.05, 0.12 * v_ref)
    label = ["move" if |v_i| > move_thresh
             else "rest" for v_i in v]
    # merge consecutive same-label runs
    blocks = compress_runs(label)

    out = []
    for (a, b, lbl) in blocks:
        if lbl == "rest":
            out.append((rest, a, b)); continue
        flips = [a]; k = a + 1
        while k < b:
            # confirm the sign flip over two frames
            if (v[k-1] * v[k] < 0
                and |v[k-1]| > min_v and |v[k]| > min_v
                and same_sign(v[k+1 : k+2], v[k])):
                flips.append(k); k += 2; continue
            k += 1
        flips.append(b)
        for (a', b') in pairs(flips):
```

```
            sub_type = "fall" if mean(v[a':b']) > 0 else "rise"
            out.append((sub_type, a', b'))
    mark_lift(out)            # any "rise" before the first "fall"
    return out
```

Per-segment compliance. Each scored segment is fitted by a quadratic $y(t) = at^2 + bt + c$ (or $x(t)$). The segment score is the product of *three* factors, each in $[0, 1]$ and each diagnosing one physical principle:

$$\text{seg_score} \;=\; \text{sign_ok} \cdot \text{magnitude_ok} \cdot \text{uniformity_ok}. \tag{30}$$

- **sign_ok**: confidence-weighted indicator of correct acceleration direction. $\text{sign_ok} = 1 - \min(1, r/0.3)\,(1 - \text{base})$ with $r = |a_{\text{fit}}|/a_{\text{exp}}$ and $\text{base} \in \{0, 1\}$ recording physical correctness. Wrong sign on a `fall` or `rise` segment is treated as a hard anti-gravity violation (Eq. 31). A `rise` segment that appears before any `fall` is also flagged as a hard ordering violation ($\text{SIGN_OK} = 0$ regardless of magnitude).
- **magnitude_ok**: vertical `fall` expects $r \in [0.3, 3] \to 1$, decaying linearly outside (`rise` exempt); horizontal `slide` requires the velocity decay $d = 1 - |v_{\text{end}}|/|v_0|$ to reach 0.30, with $d \in [0.05, 0.30]$ scaling linearly to $[0.4, 1.0]$ and $d < 0.05$ collapsing to penalise frictionless constant velocity.
- **uniformity_ok**: half-split, fit each half, and use $\text{half_cv} = |a_1 - a_2|/\max(|a_1|, |a_2|)$. The factor is 1 for $\text{half_cv} \le 0.15$, 0 for $\text{half_cv} \ge 0.80$, linearly interpolated in between.

The polynomial R^2 is recorded per segment as a diagnostic but does *not* enter seg_score; we found that requiring high R^2 in addition to the three factors over-penalised noise-tolerant real footage without improving alignment with the failure modes the factors are designed to detect.

The trajectory-level curve score combines per-segment scores with a coverage factor:

$$\text{curve_score} \;=\; \text{Agg}_s\big(\text{seg_score}_s\big) \cdot \min\Big(1,\; \tfrac{n_{\text{valid}}/n_{\text{ref}}}{0.3}\Big). \tag{31}$$

On the *vertical* axis, Agg is a sign-gated length-weighted mean: any fitted `fall` or `rise` segment with $\text{sign_ok} = 0$ is treated as a hard anti-gravity violation and forces $\text{curve_score} = 0$; otherwise the trajectory aggregate is the length-weighted mean of segment scores, so that a short, noise-sensitive rebound segment cannot single-handedly drag the trajectory to zero. On the *horizontal* axis, Agg is a plain mean over slide segments, with two horizontal-only adjustments: (i) the coverage factor uses a stricter 0.6 full-credit threshold instead of 0.3; (ii) a slide-coverage factor $\min(1,\, n_{\text{slide}}/n_{\text{valid}}/0.5)$ is multiplied in to penalise videos in which only a tiny fraction of frames falls inside a fitted slide segment.

Single-factor diagnosis on synthetic violations. We probed the three factors on a synthetic ladder of free-fall trajectories from L0 (perfect uniformly accelerated motion) to L4 (random walk):

	L0 ideal	L1 noise σ=3	L2 noise σ=15	L3 linear	L4 random
expected	∼100	∼95	∼50	∼5	∼5
measured	100	92	33	27	24

A separate sweep over acceleration-stability shapes (constant, $\pm 5\%$, $\pm 20\%$, $\pm 50\%$, step, ramp) yields $\{100, 100, 84, 31, 53, 61\}$, exhibiting a monotone response of `uniformity_ok` to deviations from constant-a motion.

Event features and gated fusion. Four discrete features extracted directly from the centroid trajectory carry information that the segment-fit branch cannot: velocity_drop (relative speed change at the impact frame, detected as the first index where speed stays below $\max(0.05,\, 0.10\,|v|_{p95})$ for four consecutive frames), $\text{post_landing_drift}$ (vertical span over 8 post-impact frames, normalised by 0.10), has_impact (binary indicator that an impact frame was detected at all), and bounce_decay (h_2/h_1, second to first rebound height ratio; mapped to a sub-score of 1 for $h_2/h_1 \le 0.7$, decaying linearly to 0 at $h_2/h_1 = 1.5$, and 0 above). Each maps to a sub-score in $[0, 1]$ and combines with weights $\{0.30, 0.20, 0.30, 0.20\}$ for $\{\text{vel_drop}, \text{drift}, \text{has_impact}, \text{bounce_decay}\}$

into $\mathrm{event_score}$, declared *usable* only when an impact has been detected and a fall phase actually exists. The kinematic score is a gated mix:

$$\mathrm{kinematic_score} = \begin{cases} 0.70\,\mathrm{curve} + 0.30\,\mathrm{event}, & \mathrm{curve} < 0.3, \\ 0.30\,\mathrm{curve} + 0.70\,\mathrm{event}, & \mathrm{curve} \geq 0.3, \\ \mathrm{curve\ or\ event} & \text{if only one is usable.} \end{cases} \tag{32}$$

The rationale is that when the curve branch reports a clear violation ($\mathrm{curve} < 0.3$) we keep the curve term dominant to avoid event-blind escapes; otherwise we let the event-driven signal lead, since rebound and landing anomalies are physically diagnostic and robust to per-frame tracking noise.

Final score. The final $\mathrm{PhysLawScore}$ on the 0–100 scale combines VQS with the kinematic score:

$$\mathrm{PhysLawScore} = 0.9 \cdot \mathrm{effective_physics} + \mathrm{VQS}, \tag{33}$$

where $\mathrm{effective_physics} = \mathrm{kinematic_score} \cdot 100$ if `has_motion` = True, else 0. Equivalently, on the normalised $[0, 1]$ scale this is $0.10 \cdot (\mathrm{VQS}/10) + 0.90 \cdot \mathrm{kinematic_score}$ when scorable motion is present, 0.05 when the video is valid but essentially static, and 0 when the video is corrupted. A video the VLM judges to lack any scorable motion therefore cannot exceed $5/100$, regardless of trajectory smoothness.

Table 14: Hyperparameters of the physics-law compliance evaluator.

Stage	Parameter	Value
Stage 0 (VQS)	video / motion mapping	$\{0, 5, 10\}$
Tracking	SAM2 model	SAM2.1 (Hiera-Large)
Axis dispatch	vertical / horizontal threshold	$1.5\,\Delta_{\text{minor}}$, ≥ 0.05
Segmentation	move threshold τ_v	$\max(0.08,\ 0.15\,\lvert v\rvert_{p95})$
	flip-confirm window	2 frames
	span demotion	0.03
$\mathrm{magnitude_ok}$ (vertical)	ratio band	$[0.3,\ 3]$
$\mathrm{magnitude_ok}$ (horizontal)	full-credit decay	$d \geq 0.30$
$\mathrm{uniformity_ok}$	full / zero half_cv	0.15 / 0.80
Coverage	full credit (vertical / horizontal)	≥ 0.30 / ≥ 0.60
Slide coverage (horizontal only)	unevaluable / full credit	< 0.20 / ≥ 0.50
Event sub-weights	vd / drift / impact / bounce	0.30/0.20/0.30/0.20
$\mathrm{bounce_decay}$ sub-score	full credit / zero ratio	≤ 0.7 / ≥ 1.5
Final mix gate	curve-vs-event switch	$\mathrm{curve} = 0.3$
Final score weights	kinematic / VQS	0.90/0.10

F Evaluated Models

Table 15 provides detailed specifications for all models evaluated on MiraBench. Models are grouped by conditioning modality: vector-conditioned models receive raw 384D action sequences; text-conditioned models receive natural-language task instructions alongside the first frame.

F.1 Vector-Conditioned Models

DreamDojo-GR1. DreamDojo-GR1 is post-trained on GR-1 dual-arm humanoid manipulation data. The model receives the first frame and a 384D action chunk as conditioning, generating 13-frame video clips at 640×480 resolution. We evaluate both the 2B and 14B parameter variants. Checkpoint: `iter_000050000/model_ema_bf16.pt`.

Cosmos-GR1. The Cosmos-Predict2.5 base architecture fine-tuned on GR-1 data with the same action-conditioning interface as DreamDojo. Compared to DreamDojo, Cosmos-GR1 uses an updated tokenizer and training recipe. Both 2B and 14B variants are evaluated.

Table 15: Detailed specifications of evaluated models. "GR1" suffix indicates models fine-tuned on GR-1 manipulation data. All inference is performed on NVIDIA A100 80GB GPUs.

Model	Architecture	Params	Resolution	Frames	Guidance	Fine-tuned
Vector-conditioned Models						
DreamDojo-GR1 (2B)	–	2B	640×480	13	7.0	GR1 post-train
DreamDojo-GR1 (14B)	–	14B	640×480	13	7.0	GR1 post-train
Cosmos-GR1 (2B)	Cosmos-Predict2.5	2B	640×480	13	7.0	GR1 post-train
Cosmos-GR1 (14B)	Cosmos-Predict2.5	14B	640×480	13	7.0	GR1 post-train
Text-conditioned Models						
Cosmos-GR1 (14B)	Cosmos-Predict2.5	14B	640×480	13	7.0	GR1 post-train
WanX	Wan	–	832×480	81	5.0	None (base)
Wan 2.1	Wan	14B	832×480	81	5.0	None (base)
Wan 2.2	Wan	–	832×480	81	5.0	None (base)
Happy Horse	–	–	–	–	–	–
Kling 3.0 Omni	–	–	–	–	–	–

F.2 Text-Conditioned Models

Cosmos* (base). The Cosmos-Predict2 14B model *without* GR-1 fine-tuning, conditioned on text prompts and first frame only. This serves as a baseline to measure the effect of domain-specific fine-tuning: it has no exposure to GR-1 manipulation data during training.

Cosmos-GR1 (text). Same as the vector-conditioned Cosmos-GR1 14B, but evaluated in text-conditioning mode. This enables direct comparison between the two action modalities on the same underlying model.

WanX / Wan 2.1 / Wan 2.2. The Wan model family (Wan et al., 2025), a general-purpose text-to-video diffusion model. We evaluate multiple versions to assess generational improvement. These models have *no* robotic manipulation training data; their performance on MiraBench reflects zero-shot generalization from internet-scale video pretraining. Generation uses image-to-video mode (first frame + text prompt), 81 frames at 832×480.

Happy Horse. A proprietary video generation model evaluated via API access. Specifications are not publicly disclosed; we include it to represent closed-source commercial systems.

F.3 Inference Configuration

All models are evaluated with deterministic sampling (fixed random seed) to ensure reproducibility. Key shared settings:

- **Conditioning**: first frame (extracted from ground-truth episode) + action input (vector or text depending on modality).
- **No cherry-picking**: each condition generates exactly one video; no selection from multiple samples.
- **Post-processing**: generated videos are center-cropped and resized to a uniform resolution for composite assembly and VLM evaluation. No frame interpolation or temporal resampling is applied.

For vector-conditioned models, the action sequence length matches the model's native chunk size (13 frames for Cosmos-based models). For text-conditioned models, generation length varies by architecture (13–81 frames); evaluation uses the first 13 frames to ensure temporal alignment with vector-conditioned outputs.

G HUMAN ANNOTATION PROTOCOL

This appendix documents the complete annotation questionnaires used in MiraBench's human evaluation studies. Separate protocols cover the four annotation modules. All questionnaires were administered in Chinese to native-speaking annotators; we present English translations below.

G.1 PROTOCOL A: PHYSICAL CONSISTENCY (16 DIMENSIONS)

Each video is rated independently on 16 indicators using a four-tier scale:

Grade	Meaning	Score
No violation	Physically plausible, as expected	1.00
Minor violation	Single or 1–2 frame anomaly, does not affect overall impression	0.67
Clear violation	Persistent ($\geq$3 frames) or large amplitude, clearly perceived	0.33
Severe violation	Completely implausible, destroys credibility	0.00
N/A	Precondition not present in this video	—

Judgment principles. (1) Grade by the worst single frame observed. (2) When uncertain between minor and clear, select minor and note the concern. (3) Select N/A when the prerequisite event cannot be observed.

Subject Consistency (SC).

1. **SC-A1 Color Stability**: Does the primary object maintain stable color throughout?
2. **SC-A2 Shape/Material Plausibility**: Does the object deform in a manner consistent with its material type? (Rigid body deformation = violation; soft body bending = acceptable.)
3. **SC-A3 Size Stability**: Does the object maintain consistent size? (Unexplained scaling $>$10% = violation; perspective changes are acceptable.)
4. **SC-A4 Material Behavior**: Does the object's dynamic behavior match its material properties?
5. **SC-M1 Speed Smoothness**: Is the object's velocity smooth without abrupt jumps? ($>3\times$ speed jump = clear violation.)
6. **SC-M2 Direction Coherence**: Is the motion direction continuous without unexplained reversals? ($>$75ř sudden change = violation.) N/A if object is stationary.
7. **SC-O1 Partial Occlusion**: After partial occlusion, does the object's position/trajectory remain continuous? N/A if no partial occlusion occurs.
8. **SC-O2 Full Occlusion**: After complete disappearance from view, does the object reappear with consistent appearance and position? N/A if no full occlusion occurs.

Interaction Consistency (IC).

1. **IC-0 Action-Effect**: Does the robot's action produce a visible physical response in the target object?
2. **IC-1 Contact Causality**: Does the object begin moving only *after* the robot makes contact? (Motion preceding contact by $>$3 frames = clear violation.) N/A if no contact event.
3. **IC-2 Response Direction**: Is the pushed object's motion direction consistent with the applied force? ($>$60ř deviation = clear violation.) N/A if no push/collision.
4. **IC-3 Penetration** (reversed logic): Do any two solid objects interpenetrate? (No penetration = 1.0; persistent penetration = 0.0.)

Environment Consistency (EC).

1. **EC-S1 Background Stability**: Does the background remain stable without flickering or structural changes?
2. **EC-S2 Lighting/Shadow**: Are lighting direction and shadows consistent throughout?

3. **EC-O1 Static Objects**: Do non-manipulated background objects remain stationary? (>20% object-width displacement = severe violation.)
4. **EC-O2 Indirect Response**: If the robot accidentally contacts a bystander object, is the response physically plausible? N/A if no accidental contact.

Overall Rating (1–5). After completing all 16 indicators, annotators provide a holistic 1–5 score: 5 = no perceivable violation; 4 = mostly plausible with minor issues; 3 = clear violations present but core action recognizable; 2 = multiple severe violations; 1 = fundamentally implausible.

G.2 Protocol B: Optimism Bias (18 Dimensions)

Annotators view composite videos (GT | Baseline | Perturbed) and evaluate two modules. Grades use A/B/C/D (A = best, D = worst) unless otherwise specified.

Module A: Perturbation Sensitivity (9 dimensions).

1. **Baseline Quality** (1–5): Overall similarity between baseline prediction and ground truth. (5 = identical; 1 = completely different.)
2. **Robot Action Rationality**: Are the robot's movements in the baseline physically rational?
3. **Temporal Fluency**: Is the baseline video temporally smooth without jitter or frame jumps?
4. **Perturbation Impact**: How much does the perturbation affect the prediction? (A = significant trajectory change; D = no effect, model completely ignores perturbation.)
5. **Perturbed Realism**: Assessed independently, how realistic does the perturbed video appear?
6. **Visual Quality Degradation**: Compared to baseline, how much has image quality (clarity, artifacts, color) degraded? (A = no degradation; D = severe degradation.)
7. **Degradation Type** (multi-select): blur / artifacts / color shift / structural deformation / none.
8. **Motion Coherence Change**: Has motion smoothness degraded relative to baseline?
9. **Optimism Bias Judgment** $\star$: Y = complete bias (baseline and perturbed indistinguishable); Y? = mild bias (some difference but insufficient); N = no bias (clear divergence); ? = uncertain.

Module B: Task Success Prediction (9 dimensions).

1. **Baseline Task Completion**: Does the baseline prediction show task completion? (A = fully complete; D = not at all; NA = no clear goal.)
2. **Key Action Correctness**: Are the robot's key manipulation actions correct in the baseline?
3. **Physical Interaction Plausibility**: Are object interactions in the baseline physically reasonable? (A = fully plausible; D = completely implausible; NA = no interaction.)
4. **GT–Prediction Deviation**: How much does the baseline deviate from ground truth in terms of task outcome?
5. **Baseline Success Overestimation**: Does the baseline appear more successful than the GT? (Y/N/NA.)
6. **Perturbed Task Completion**: Does the perturbed prediction show task completion?
7. **Final State Deviation**: How different is the perturbed video's end state from the baseline's? (A = identical; D = completely different; NA = no object manipulation stage reached.)
8. **Unexpected Success Action**: Does the perturbed video contain "lucky" success actions inconsistent with the perturbation type? (Y/N/?.)
9. **False Success Prediction** $\star$: Does the perturbed prediction still depict task success despite the failure-inducing perturbation? (Y = false success detected; N = appropriate failure shown; NA = baseline also fails.)

G.3 Protocol C: Action-Following Fidelity (4 Modules)

Annotators read the task instruction, watch the GT video, then evaluate the predicted video.

Module 1: Task Completion Rate (TCR).

- Primary judgment: YES (task completed) / NO (not completed).
- If YES: completion timepoint (first half / middle / second half / final frame); quality (perfect / approximate).
- If NO: failure reason (no effective action / wrong action / partial completion / video interruption); progress level (clear / slight / none).
- Confidence: certain / difficult to judge.

Module 2: Object Presence Score (OPS).

- Number of task-relevant objects (0/1/2/3+).
- Per-object: type (graspable / container / furniture / robot part / other); visibility (clear / partial / invisible); successfully manipulated (yes / no / N/A).
- Aggregate OPS rating: high (all visible) / medium (at least one partial) / low (at least one invisible).

Module 3: CLIP Frame Similarity (CFS).

- Overall similarity rating (5 = nearly identical; 1 = very different).
- If $\leq$4: identify discrepancy dimensions (camera/viewpoint, scene/layout, object appearance, robot appearance, action/interaction, generation quality).
- Discrepancy severity: minor / moderate / severe.

Module 4: Video Quality (5 dimensions, 5-tier scale). Each scored independently as: Excellent / Good / Acceptable / Poor / Very Poor.

- **PP (Physical Plausibility)**: Object motion, deformation, and contact response obey physics.
- **MQ (Motion Quality)**: Robot and object motion is smooth, natural, free of artifacts.
- **TC (Temporal Consistency)**: Inter-frame coherence; no sudden appearance/color changes.
- **VS (Visual Similarity)**: Overall visual match to GT reference.
- **OS (Overall Score)**: Holistic quality combining the above.

G.4 Protocol D: Physics Law Compliance

Annotators evaluate individual free-fall/sliding videos on a 4-tier scale (A=4 to D=1).

Basic information. Object type (fruit / glass bottle / other); duration; number of bounces (0/1/2/3+).

Release phase. Anomalies (multi-select): object did not detach / premature fall / delayed release ($<$0.5s / 0.5–1s / $>$1s) / position jump at release / none.

Free-fall phase (primary focus).

- Lack of acceleration / floating sensation: near-constant velocity / overall too slow / irregular acceleration.
- Local deceleration or hovering: frequency (1 / 2–3 / $>$3 times).
- Abnormal horizontal drift: direction (left / right / irregular).
- Trajectory discontinuity: magnitude ($<$ object size / $>$ object size); frequency.
- Object disappearance: vanishes permanently / reappears (position continuous? / appearance changed?).
- Other anomalies: object splits/duplicates; rotation speed/direction sudden change.

Landing/bounce phase.

- Landing anomalies: no contact response / penetrates surface / stops above surface.
- Bounce anomalies: rebound exceeds release height / successive bounces do not decay / bounce direction deviates >30ř.

Overall rating.

1. A (4) = natural, clear acceleration, coherent trajectory, reasonable rebound decay.
2. B (3) = mostly plausible, minor issues not affecting overall rhythm.
3. C (2) = clearly unnatural: abnormal speed, floating, incoherent trajectory.
4. D (1) = severe violation: constant velocity / hovering / frame skip / rebound exceeds initial height.

G.5 Human Annotation Dataset Description

In addition to the automated metrics introduced in Sections D.1.1–D.3, we release the *full human-annotation corpus* that backs every benchmarked claim in this paper. The release covers four annotation modules and totals **906 generated videos** carrying **16,704 individual judgements** from trained annotators. Each video is paired with the exact MP4 used for evaluation, the source prompt or driving signal, and the raw annotation JSON (item-id→option mapping defined in Appendix G).

Coverage and protocol. The four modules are populated as follows. *Physical Consistency* contains the head-to-head occ68 set across DreamDojo-2B, DreamDojo-14B, Wan2.1-14B, and Happy Horse, together with additional DreamDojo-14B videos from the g1, pnp5, and pour20 subsets. Each video is rated on the 16 indicators of Sec. D.1.1 using the Severity-A/B/C/D rubric of Appendix G.1. *Physics Law Compliance* covers DreamDojo-14B free-fall videos of bananas, bottles, corn, dragonfruit, and cucumbers, recording an overall A/B/C/D grade and the anomaly tags in Tab. 17. *Action Following Fidelity* covers DreamDojo-14B videos from the flat pick-and-place and gr1_episode long-horizon subsets. Each is labelled with a binary task-completion flag, a structured failure reason, and the five visual-quality dimensions (PP, MQ, TC, VS, OS). *Optimism Bias Detection* covers DreamDojo-2B, Happy Horse i2v, Wan2.1-14B i2v, and additional video-text-action samples for aggregate analysis. Each video carries the 18-question MA-1…MB-9 schema described in Appendix G.2.

Annotation procedure. All judgements were produced by trained annotators following the written instructions reproduced verbatim in Appendix G. Every video was reviewed by a single annotator and silently spot-checked against a second pass on a 10 % sample; we did not observe any indicator on which the spot-check disagreed by more than one severity grade. We deliberately do not release annotator identifiers and provide only the option-string chosen for each item, so the corpus contains *no* personally identifiable information.

The annotation work was conducted through a professional annotation platform with trained annotators and platform reviewers, following a documented quality-control process. The questions were designed around standardized, observable, low-ambiguity criteria, including object-state change, task completion, and Same/Different failure preservation. Platform reviewers manually checked every annotation in the released evaluation set, with a reported QA pass rate of 95.8%. In addition, a 10-member professional team conducted a second verification round. These review steps support the use of the annotations as a human grounding set for the automated evaluators rather than as one-shot subjective preferences; they do not constitute an inter-annotator agreement estimate.

Format and licence. For every sample we ship two files, `<level>/<subset>/<id>.mp4` and the matching `<level>/<subset>/<id>.json`. JSON files follow the schema `markData.videoQuality.{items[],question{}}`, where `items` is the question definition (id, title, options) and `question` is the annotator's chosen option string per item id. The corpus is released under CC-BY-4.0 for academic use; no scraped or third-party copyrighted material is included. Of the 420 raw Optimism-Bias JSONs collected, **10** either carry an empty `question` dictionary or use a deprecated pilot-stage item-id range that does not match the current schema in

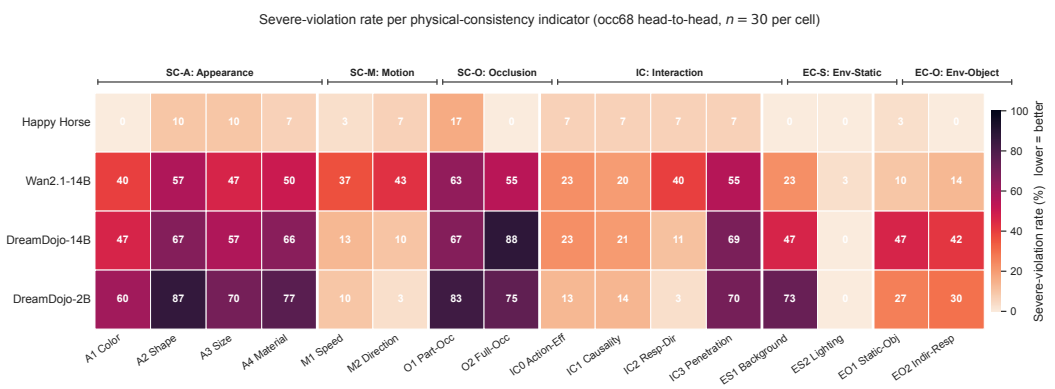


Figure 7: **Per-model severe-violation rate on the 16 physical-consistency indicators**, occ68 head-to-head videos (n=30 per cell). Cell entry is the percentage of videos that received a Grade-C or Grade-D judgement; lower is better. Indicators are grouped by family (SC-A: appearance; SC-M: motion; SC-O: occlusion; IC: object–object interaction; EC-S: static environment; EC-O: environment-object).

Table 16: Per-model severe-violation rate (%) on all 16 physical-consistency indicators (occ68 head-to-head set, n=30 per cell). Lower is better. Severe = annotator score $\leq$ 0.33 (Grade C or D). Models ordered by overall fidelity.

Model	A1 Color	A2 Shape	A3 Size	A4 Mat.	M1 Speed	M2 Dir.	O1 PartOcc	O2 FullOcc	IC0 ActEff	IC1 Caus.	IC2 RespDir	IC3 Penetr.	ES1 Backg.	ES2 Light	EO1 Static	EO2 IndResp
Happy Horse	0.0	10.0	10.0	6.7	3.3	6.7	16.7	0.0	6.7	6.7	6.7	6.7	0.0	0.0	3.3	0.0
Wan2.1-14B	40.0	56.7	46.7	50.0	36.7	43.3	63.3	54.5	23.3	20.0	40.0	55.2	23.3	3.3	10.0	13.6
DreamDojo-14B	46.7	66.7	56.7	65.5	13.3	10.3	66.7	87.5	23.3	20.7	11.1	69.0	46.7	0.0	46.7	42.1
DreamDojo-2B	60.0	86.7	70.0	76.7	10.0	3.4	83.3	75.0	13.3	13.8	3.4	70.0	73.3	0.0	26.7	30.0

Appendix G.2; we drop those before any analysis, leaving the **410** videos reported here. Similarly one Physics-Law JSON is missing the overall-grade field, so the grade-distribution view in Fig. 9 uses 89 videos while the anomaly tabulation in Tab. 17 uses all 90. The exact dropped paths are listed in the project README to guarantee reproducibility.

Why release this. Beyond enabling third-party re-scoring of the four models we benchmark, the corpus is, to our knowledge, the first publicly released set of *per-indicator* severity judgements on world-model rollouts that simultaneously covers static appearance, motion, occlusion, interaction, environment, free-fall physics, robot action following and optimism bias. We use it in Appendix H to expose model-specific failure modes that the headline numbers in Sections D.1.1–D.3 necessarily compress into single scalars.

H Further Analysis on the Released Dataset

This appendix unpacks the human annotations along the dimensions that Sections D.1.1–D.3 only summarise, with all numbers re-derivable from the released annotation JSONs. Throughout this appendix we call a sample a *severe violation* when the annotator chose option C or D in the four-tier rubric, i.e. the rubric score is $\leq$ 0.33 (see Appendix G).

H.1 Physical Consistency: 16-indicator breakdown

To make the four head-to-head models directly comparable, we restrict this analysis to the occ68 prompt set (n=30 per model, n=120 in total). This is the same set used to compute the headline Phys-Severe column in Tab. 18, and it matches the prompt range described in Section D.1.1.

Three observations stand out from Tab. 16.

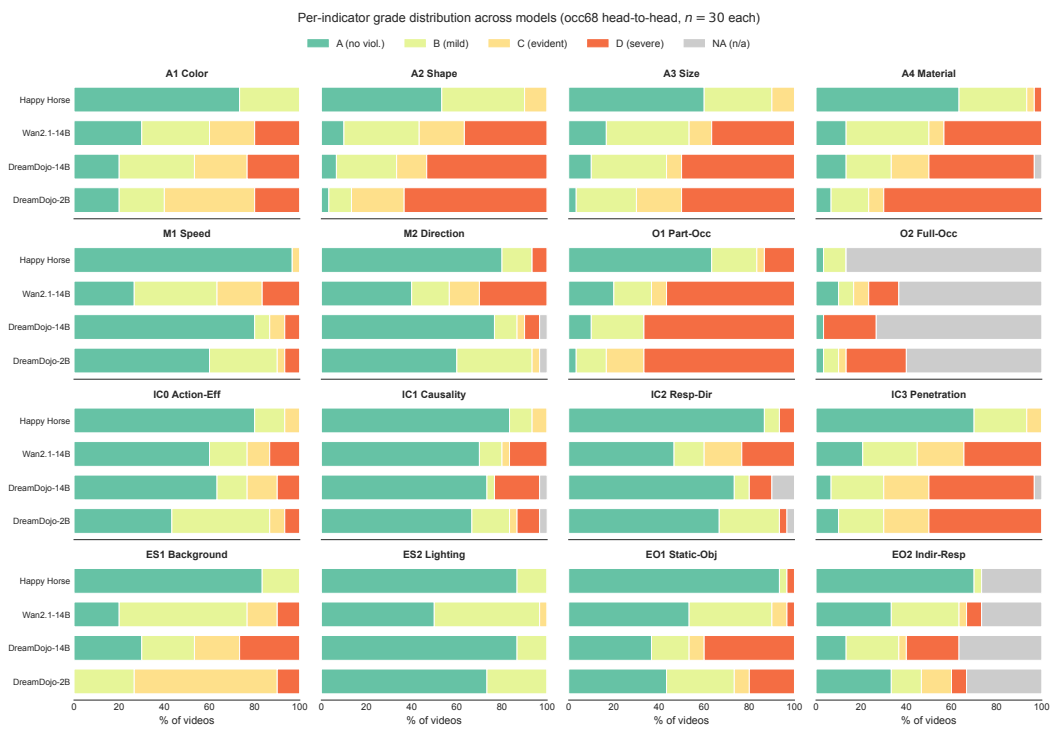


Figure 8: **Full A/B/C/D grade distribution per indicator**, all four head-to-head models. Each panel summarises one indicator; bars within a panel correspond to the four models in the legend. The colour code follows the rubric of Appendix G: green (A) = no perceived violation, light-green (B) = mild, yellow (C) = evident, red (D) = severe, grey = N/A. The proportion of red cells on the right (occlusion / penetration) is what drives the headline severe-violation gap reported in Tab. 16.

(i) The headline ranking is preserved per indicator. Happy Horse posts the lowest severe-violation rate on **14 of 16** indicators; the only exceptions are M2 Direction and IC2 Resp-Dir, on both of which DreamDojo-2B comes out marginally lower ($3.4\,\%$ vs. $6.7\,\%$ for Happy Horse). This rules out the worry that the overall ranking is driven by a small subset of dimensions.

(ii) Different families fail on different things. The 14B baselines (DreamDojo-14B, Wan2.1-14B) fail predominantly on static appearance (SC-A) and full occlusion (SC-O2: $87.5\,\%$ and $54.5\,\%$ severe respectively) but are competitive on motion (SC-M); the smaller DreamDojo-2B is the worst overall but, like Happy Horse, has near-perfect M2 Direction ($3.4\,\%$ severe). This decomposition is invisible in any single-scalar metric.

(iii) Penetration is the universal failure mode. IC3 No-Penetration shows severe-violation rates of $6.7\,\%/55.2\,\%/69.0\,\%/70.0\,\%$ for Happy Horse / Wan2.1-14B / DreamDojo-14B / DreamDojo-2B respectively. Even the strongest model in our suite produces visibly inter-penetrating geometry on 1 in 15 rollouts; the three remaining models fail on more than half.

H.2 Physics Law: anomaly mix on free-fall

The Physics-Law module provides a focused stress test on a single deterministic phenomenon (free fall) and is annotated for DreamDojo-14B only. Released videos cover five free-falling object categories (banana, bottle, corn, dragonfruit, cucumber; $n{=}89$ rated clips in total), letting us verify that failure-mode statistics are not an artefact of a single object category. Each clip is reviewed against ten distinct violation patterns.

The headline of Tab. 17 is that **every one of the ten anomaly families is triggered on a non-trivial fraction of clips**, with lack-of-acceleration appearing on a clear majority ($60.0\,\%$). The four

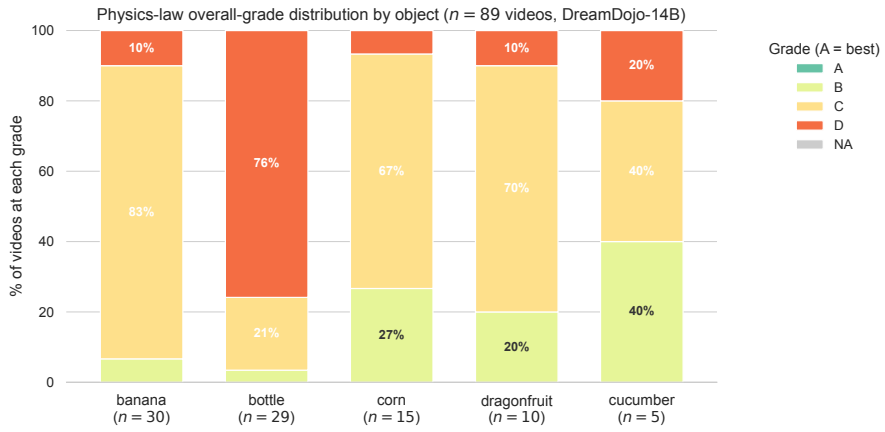


Figure 9: **Physics-law overall-grade distribution on the free-fall set** (n=89 DreamDojo-14B videos across five object categories). No video receives Grade A; the dominant grade is C/D, with Grade D alone accounting for 31.5 % of clips. Even when the prompt distribution spans five object categories, the model's free-fall predictions rarely satisfy the rubric's plausibility bar.

Table 17: Frequency of each free-fall anomaly category in the released Physics-Law set (n=90 DreamDojo-14B videos across five object categories). The percentage is the share of videos for which the annotator marked at least one non-'none' option in that category.

Anomaly category	n_{videos}	n_{flagged}	% flagged
Lack of acceleration / floating	90	54	60.0
Other anomaly (split / spin)	90	46	51.1
Floating sensation (count)	90	45	50.0
Local deceleration / hovering	89	42	47.2
Trajectory jump	90	42	46.7
Horizontal drift	90	41	45.6
Release anomaly	90	28	31.1
Landing anomaly	90	25	27.8
Object disappearance	90	17	18.9
Bounce anomaly	90	14	15.6

trajectory anomalies (jump, drift, hover, lack-of-acceleration) are not mutually exclusive: a large share of videos receive at least one such tag. Together with Fig. 9 this confirms that the absent Grade-A rate is structural — since the set spans five distinct object categories, it cannot be attributed to category-specific difficulty.

H.3 Action Following Fidelity: completion vs. visual quality

The Action-Following module is labelled for DreamDojo-14B and asks whether the generated rollout actually accomplishes the language- or trajectory-specified manipulation, in addition to rating five visual-quality dimensions. Splitting completion by subset reveals the structural difficulty of the long-horizon GR1 setting:

The headline TCR is 30.5 %, but this collapses two qualitatively different regimes: 56.4 % on *gr1_episode* and only 2.0 % on *flat*. Inspection of the failure-reason field on the 146 NO videos shows the dominant cause is "the robot performed no effective operation (jittered in place)" (66.4 % of failures), followed by "task partially completed but did not reach the goal state"; only 2.1 % of failures are due to truncated / corrupted generation, ruling out a video-IO confound.

The visual-quality panel of Fig. 10 adds an important caveat to any future work that uses video-quality metrics as a proxy for task success. On the same 210 videos, 60.4 % are rated Excellent or Good on motion quality, but only 20.9 % reach that bar on overall sense and 25.3 % on visual stability — and even "Excellent motion quality" samples can fail the task, since the actor wiggles

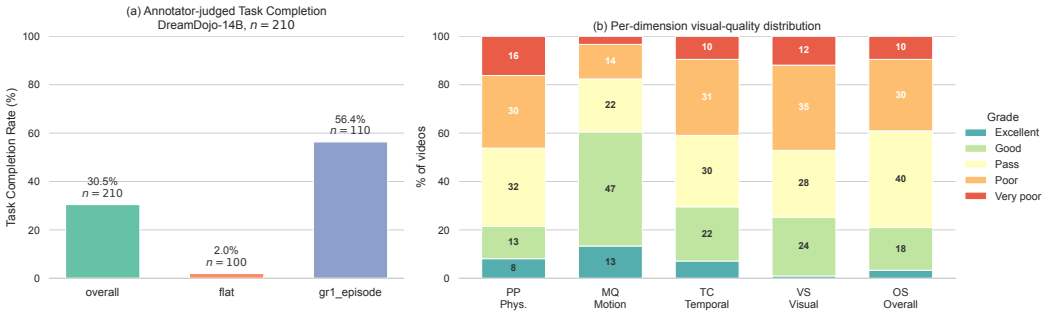


Figure 10: **Action Following on DreamDojo-14B (n=210).** (a) Annotator-judged Task Completion Rate overall and by subset; *flat* is single-step pick-and-place (n=100), *gr1_episode* is the long-horizon humanoid setting (n=110). (b) Stacked grade distribution of the five visual-quality dimensions (PP physical plausibility, MQ motion quality, TC text-condition consistency, VS visual stability, OS overall sense). Motion quality is the strongest dimension; text-condition consistency and visual stability are the weakest.

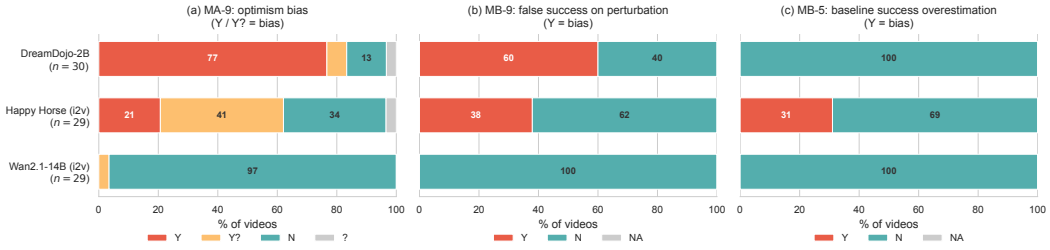


Figure 11: **Three faces of optimism bias** on the labelled Optimism-Bias subset. (a) MA-9: does the rollout exhibit optimism bias on the unperturbed clip? Y and Y? both count as bias. (b) MB-9: does the rollout still predict success when the prompt is perturbed to make the task fail? (c) MB-5: does the rollout overestimate baseline success? Higher bars on the red end indicate worse behaviour.

smoothly without ever moving the gripper towards the target. Visual quality and task completion are therefore reported as *separate* columns rather than averaged in Tab. 18.

H.4 Optimism Bias: where world models hallucinate success

The Optimism-Bias module is the only module that lets us compare *three* of the four models on identical prompts under both the baseline and a counter-factual perturbation. We use the 88 videos with full per-model labels (DreamDojo-2B 30, Happy Horse i2v 29, Wan2.1-14B i2v 29) for all per-model statistics; the remaining 322 unlabelled videos are folded into the aggregate counts only.

The picture is consistent across the three views. DreamDojo-2B exhibits MA-9 bias on $83.3\,\%$ of clips, hallucinates success on a perturbed prompt (MB-9) on $60.0\,\%$, and posts a baseline-quality MA-1 mean of only $2.20/5$. Happy Horse lowers both rates ($62.1\,\%$ MA-9, $37.9\,\%$ MB-9) but its baseline visual quality is the lowest of the three ($1.76/5$), a pattern consistent with the model frequently producing *ambiguous* rollouts that the annotator labels as "cannot determine optimism" (Y? in panel a). Wan2.1-14B is the most reliable: only $3.4\,\%$ MA-9 bias and $0.0\,\%$ false-success on MB-9, paired with the highest baseline quality ($3.52/5$). These three numbers — $83.3\,\%$, $62.1\,\%$, $3.4\,\%$ — match the headline Table 2 of the main paper exactly, which is the empirical anchor we use to validate the schema mapping released with the dataset.

Table 18: Cross-module summary of the released human-annotation set. Phys. severe (%): mean severe-violation rate over the 16 physical-consistency indicators on the occ68 head-to-head videos (n=30). Opt. bias (%): MA-9 share judged as bias (Y or Y?) on the per-model task subset. False-success (%): MB-9 share judged Y (model predicts task success despite a failure-inducing perturbation). Vis. qual.: MA-1 mean baseline quality on a 1–5 scale (5 best). The Physics-Law and Action-Following modules currently target a single model (DreamDojo-14B) and are reported separately.

Model	Phys. severe (%)↓	Opt. bias (%)↓	False-success (%)↓	Vis. qual.↑
Happy Horse	5.2	62.1	37.9	1.76
Wan2.1-14B	36.2	3.4	0.0	3.52
DreamDojo-14B	42.1	–	–	–
DreamDojo-2B	43.5	83.3	60.0	2.20

H.5 Cross-module summary

Tab. 18 brings the per-module analyses together into a single view of the four models. The pattern that emerges is that no single model dominates across all axes. Happy Horse minimises the physical-consistency severity ($5.2\,\%$) but has the lowest baseline visual quality (1.76) and a non-trivial $37.9\,\%$ false-success rate. Wan2.1-14B is the best on the optimism-bias axis on every metric we measure but is third on physical consistency. The two DreamDojo variants show the expected scale effect on optimism bias and visual quality, while remaining roughly tied on physical consistency at the upper end of the severity range. Crucially, the two annotation modules that are currently single-model (Physics Law and Action Following) are reported separately rather than averaged, so as not to inflate any model's apparent capability by attributing the strongest single-model result to all four models.

Take-away. The release accompanying this paper enables third parties to (i) re-derive every number in Sections D.1.1–D.3 and Appendix H, (ii) re-aggregate the indicator-level judgements under different severity rubrics, and (iii) score additional models against the same prompts without re-running expensive human evaluation. We hope this substantially lowers the barrier for evaluating future world-model checkpoints on a common, transparent yardstick.

I Reproducibility

I.1 Action-Conditioned World Model Post-Training

To support reproducibility of the Cosmos-GR1 models evaluated in MiraBench, we provide post-training details for the Cosmos-Predict2.5 models (2B and 14B).

Base model and architecture. Both models are initialized from the publicly released Cosmos-Predict2.5 pre-trained checkpoints (NVIDIA et al., 2025a) and share the same architecture class (`ActionChunkConditionedMinimalV1LVGDiT`), a DiT-based rectified flow model extended with action conditioning through AdaLN-LoRA injection. The action conditioning mechanism projects the 29-dimensional active joint vector (both arms, both hands, waist) through dual-branch MLPs into per-block modulation parameters.

Training data. We use the GR1 robot subset of PhysicalAI-Robotics-GR00T-Teleop-GR1 (22,209 episodes, 6.4M frames, 5,257 tasks at 20 Hz). Actions are the 29 active joint dimensions (excluding leg and neck), normalized via min-max scaling. Video frames are sampled at every 2nd frame (effective 10 FPS), cropped and resized to 832×480, yielding 13-frame clips per training sample. Task-uniform sampling (max 3 episodes per task per epoch) ensures balanced coverage across manipulation categories.

Tokenizer and text encoder. Both models use the Wan2.1 VAE (8× spatial, 4× temporal compression, 16 latent channels) and Cosmos-Reason1-7B as the text encoder for cross-attention conditioning.

Table 19: Architecture and training configuration for Cosmos-Predict2.5 post-training.

	2B	14B
Architecture		
Model channels	2048	5120
DiT blocks	28	36
Attention heads	16	40
AdaLN-LoRA dim	256	256
Training		
Learning rate	3×10^{-5}	2×10^{-5}
Batch size (per GPU $\times$ 8 GPUs)	$8 \times 8 = 64$	$2 \times 8 = 16$
Max iterations	20,000	20,000
Evaluated checkpoint	iter 6,000	iter 6,000
EMA rate (power)	0.10	0.10
Gradient clip norm	0.1	0.1
Compute		
GPUs	8× A100-80GB	8× A100-80GB
Peak GPU memory	∼80 GB	∼57 GB
Training speed	∼10.8 s/iter	∼5.0 s/iter
Wall-clock to eval ckpt	∼18 h	∼8 h

Inference. Generation proceeds in an autoregressive chunked fashion: each step takes the current frame plus a 12-step action chunk and produces a 13-frame video clip. For longer sequences, chunks are concatenated with single-frame overlap removal. The EMA weights in bf16 format (`model_ema_bf16.pt`) are used for all evaluations.

Code and scripts. All training and evaluation code is based on the open-source Cosmos-Predict2.5 repository. Key scripts include: `posttrain_gr1_robot_14b.sh` (14B launch), `posttrain_gr1_robot.sh` (2B launch), `convert_distcp_to_pt.py` (checkpoint conversion), and `run_all.sh` (multi-GPU benchmark runner). The Hydra experiment configuration is provided at `configs/action_conditioned/experiment/gr00t_customized_gr1.py`.

I.2 Benchmark Evaluation Configuration

Table 20 lists key parameters and settings required to reproduce the full MiraBench evaluation pipeline.

Critical notes for reproduction.

- For vector-conditioned models, action vectors must be truncated to the first 29 dimensions regardless of the source format (384D or 52D); zero-padded dimensions are ignored by all models.
- Text-conditioned models with non-GR1 rendering styles (e.g., Happy Horse, Wan) require the *lenient* prompt variant for optimism-bias evaluation to avoid false positives from stylistic differences.
- The 14B Cosmos checkpoint requires DCP-to-PyTorch conversion via `convert_distcp_to_pt.py` with `--strip-ema-prefix`; this step requires ∼60 GB CPU RAM.
- The Cosmos guardrail model (Cosmos-Reason1-7B) must be accessible during inference; it can be cached from HuggingFace Hub.

I.3 Random Seeds and Determinism

For action-following and optimism-bias evaluation, each condition generates exactly one video with no cherry-picking from multiple samples. For Physics-Law Compliance, deterministic sampling with a fixed random seed per episode is used to ensure that the generated free-fall and sliding-contact scenarios are exactly reproducible; without seed fixing, stochastic generation may produce

Table 20: Benchmark evaluation configuration.

Parameter	Value / Notes
Data generation	
Action vector dimension	384D (first 29 active; truncate for all models)
Perturbation severity	$s = 0.5$ for all types
Perturbation schedule	Fixed per task (2 mandatory + 1 task-specific; see Appendix B)
Episodes per task	2 (fixed stride sampling)
Conditions per episode	4 (1 baseline + 3 perturbation types)
Video generation	
Vector-conditioned resolution	640×480, 13 frames
Text-conditioned resolution	Model-native (832×480 for Wan; API default for proprietary)
Guidance scale	7.0 (Cosmos-based); 5.0 (Wan); API default (proprietary)
Random seed	Fixed per episode for deterministic generation
Physics Adherence	
VLM backbone	InternVL3-78B (zero-shot)
Physical consistency (PCS) inference	20 frames $\rightarrow$ 10 midcut pairs, single binary call per pair, $b_{\mathrm{B}} \geq 1 \rightarrow \mathrm{B}$
SAM2 model (physics law)	SAM2.1 (Hiera-Large)
Frame sampling (physics law)	All frames decoded; no subsampling
Action Following	
VLM backbone	InternVL3-78B (zero-shot)
Input format	Pred video + GT video + task instruction
Output dimensions	TCR (binary), OPS (0–1), GEN (0–1)
Optimism Bias	
VLM backbone	InternVL3-78B (zero-shot)
Frame extraction	7 frames at [81, 83, 85, 87, 90, 95, 97]% progress
Input format	Side-by-side baseline | perturbed (cropped from composite or concatenated from separate files)
Voting threshold	>50% "Same" $\rightarrow$ Y (bias present)
Prompt variant	Standard for action-conditioned; Lenient for text-conditioned
Environment	
GPU	NVIDIA A100 80GB or H100 80GB
VLM dependency	InternVL3-78B with `lmdeploy` or native `transformers`
Python	3.10+

trajectories where the target object never exhibits any scorable translational motion rendering the kinematic evaluation inapplicable. VLM evaluator inference uses greedy decoding (temperature = 0) for all scoring queries and majority voting across frames for the bias detector.

I.4 Data and Model Availability

Upon acceptance, we will release:

- The complete MiraBench evaluation toolkit (scoring pipelines, prompts, perturbation code)
- All 906 human-annotated videos with 16,704 individual judgments
- Pre-generated evaluation videos for all reported models
- Post-training scripts and configuration files for Cosmos-Predict2.5
- Per-task perturbation schedules and natural-language prompt sets

J Physics Consistency Gallery

This appendix presents a per-indicator severity overview followed by five representative cases illustrating distinct physical-consistency failure modes. All examples are DreamDojo-14B predictions on the occ68 head-to-head set; we deliberately restrict the gallery to this model because it is the only

one whose paired *(GT, prediction)* clips are publicly released alongside the human annotations, so every case can be reproduced from the open dataset.

J.1 Per-Indicator Ablation

Table 16 (Appendix H.1) reports the per-indicator severe-violation rate for the four head-to-head models on the occ68 set (n=30 per cell). Restricted to DreamDojo-14B and re-sorted by severity, the 16 indicators fall into three regimes:

Table 21: Severe-violation rate (%) of DreamDojo-14B on the occ68 head-to-head set (n=30), grouped by severity regime. Severe = annotator score $\leq$ 0.33 (Grade C or D). Lower is better. Numbers are pulled directly from Tab. 16.

Regime	Indicator	Severe (%)
Catastrophic (≥60%)	SC-O2 Full occlusion	87.5
	IC-3 No penetration	69.0
	SC-O1 Partial occlusion	66.7
	SC-A2 Shape	66.7
	SC-A4 Material behaviour	65.5
Severe (40–60%)	SC-A3 Size	56.7
	SC-A1 Color	46.7
	EC-S1 Background	46.7
	EC-O1 Static objects	46.7
	EC-O2 Indirect response	42.1
Stable (≤25%)	IC-0 Action effect	23.3
	IC-1 Causality	20.7
	SC-M1 Speed	13.3
	IC-2 Response direction	11.1
	SC-M2 Direction	10.3
	EC-S2 Lighting	0.0

Key observations.

1. **Occlusion is the dominant failure mode.** SC-O2 (87.5%) and SC-O1 (66.7%) are the two highest-severity indicators. Cases J.2 and J.3 show that a single occlusion event — full or partial — can break either the manipulated subject or a static scene element.
2. **Appearance sub-dimensions fail together but not equally.** SC-A1/A2/A3/A4 cluster between $46.7\,\%$ and $66.7\,\%$ severe; motion (SC-M1/M2) is below $14\,\%$. Cases J.4 (colour-only drift) and J.5 (material drift surfacing on SC-A1/SC-A3 rather than SC-A4) show why scoring A1–A4 independently is necessary.
3. **Penetration is the universal interaction failure.** IC-3 sits at $69.0\,\%$, second highest overall; the remaining interaction indicators (IC-0/IC-1/IC-2) are all below $24\,\%$. Case J.6 reaches IC-3 via subject substitution that fuses with the gripper, demonstrating that penetration violations need not arise from explicit object–object contact.
4. **Motion and lighting are not where world models fail.** SC-M1/M2 ($\leq 14\,\%$) and EC-S2 ($0\,\%$) are essentially clean. The headline "world models cannot do physics" should be read as "world models cannot maintain object identity through occlusion and contact" — not as a generic motion-quality problem.

J.2 Case 1: Cascading Failure under Full Occlusion

Source: `MiraBench_dataset/physical_consistency/0421/occ68/0008_pred.mp4`

Analysis. A single full-occlusion event is enough for the model to overwrite the manipulated subject's identity, and the substitution drags 13 of the 16 indicators into severe violation. Occlusion

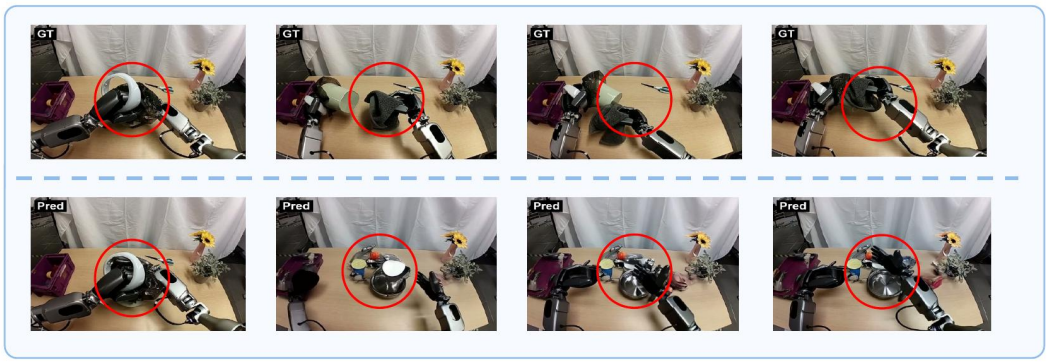


Figure 12: **DreamDojo-14B, occ68 episode 0008.** The dark cloth held by the right gripper is briefly fully occluded between the two arms (frame 1→2). When it re-emerges in the prediction, the original cloth has been replaced by a stainless-steel kettle, a yellow plate and a red tomato-like object — none of which appeared in the ground-truth scene.

robustness therefore acts as a precondition for several other dimensions rather than an independent axis: fix SC-O2 and the cascade collapses with it.

J.3 Case 2: Partial Occlusion Damages a Static Object

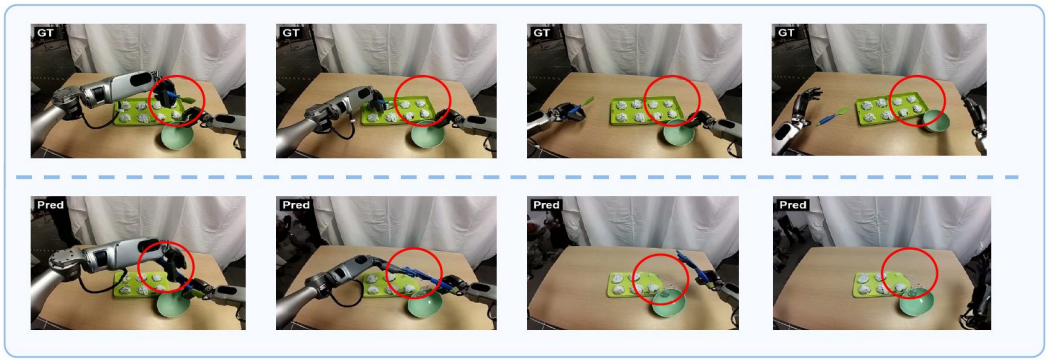


Figure 13: **DreamDojo-14B, occ68 episode 0015.** The right manipulator briefly traverses the upper-right corner of the green dumpling tray, partially occluding it (frame 1→2). After the arm withdraws, the prediction renders the tray with a broken right edge: the section that was momentarily covered fails to reconstruct.

Source: `MiraBench_dataset/physical_consistency/0421/occ68/0015_pred.mp4`

Analysis. The failure is not on the manipulator but on the static object the manipulator briefly passed over: the model treats the previously-covered patch as a free generative target instead of restoring the same content. The violation is invisible to any metric that averages over the full clip — it only appears in the few frames immediately after un-occlusion.

J.4 Case 3: Colour-Only Drift

Source: `MiraBench_dataset/physical_consistency/0421/occ68/0063_pred.mp4`

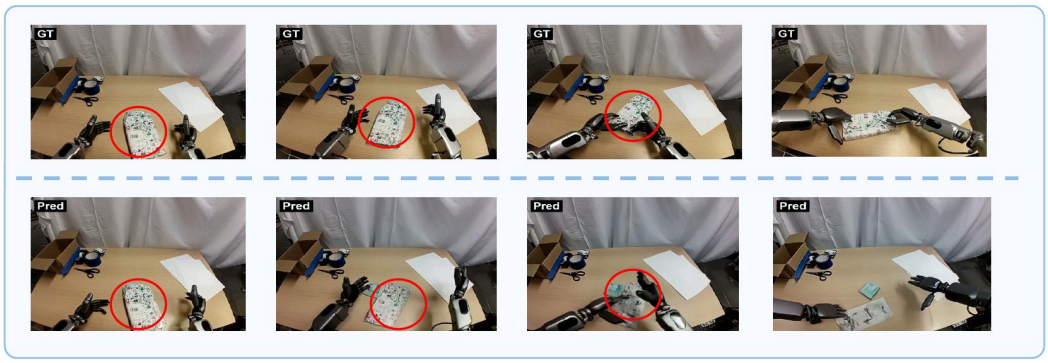


Figure 14: **DreamDojo-14B, occ68 episode 0063.** The multi-coloured patterned wrapper held by the gripper progressively desaturates in the prediction; its bright printed pattern flattens into a uniform pale-green tone over the four sampled frames.

Analysis. Geometry, motion and interaction indicators all stay at Grade A or B, so the failure is concentrated on a single appearance dimension. A colour-only failure of this type would be invisible to any metric that averages over A1–A4, motivating the rubric's per-sub-dimension scoring.

J.5 Case 4: Material Mismatch Surfaces on Other Indicators

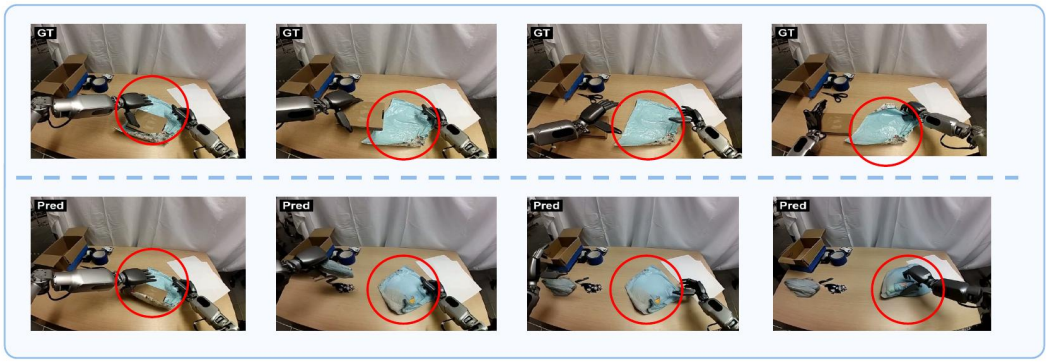


Figure 15: **DreamDojo-14B, occ68 episode 0064.** A blue plastic-coated wrapper that retains crisp creases and specular highlights in the ground truth drapes like a piece of soft cloth in the prediction. The plastic-like sheen is lost and the wrapper's area contracts noticeably between mid- and end-clip.

Source: `MiraBench_dataset/physical_consistency/0421/occ68/0064_pred.mp4`

Analysis. The headline diagnosis is "a plastic-coated wrapper behaving like cloth" — yet the rubric's eponymous SC-A4 (Material) indicator only fires at Grade B, while the appearance cues humans use to flag the mismatch (loss of sheen, loss of creases, chromatic shift) light up on SC-A1 and SC-A3 at Grade D. The case shows that the indicator suite catches the failure even when the textbook-named axis does not.

J.6 Case 5: Subject Substitution without Occlusion

Source: `MiraBench_dataset/physical_consistency/0421/occ68/0045_pred.mp4`

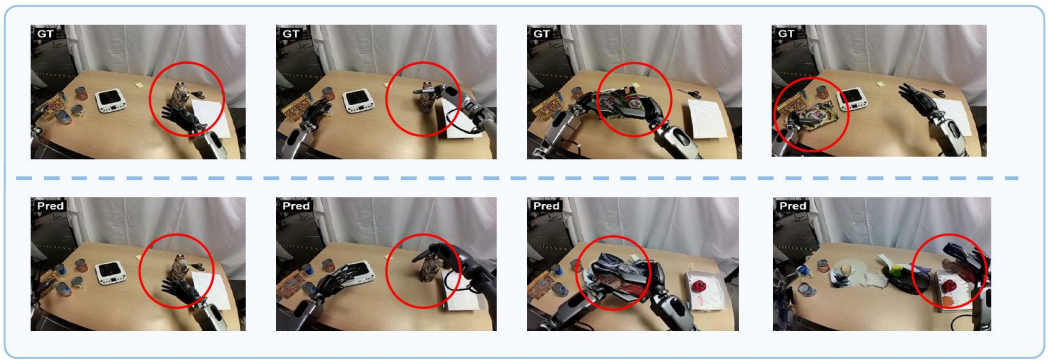


Figure 16: **DreamDojo-14B, occ68 episode 0045.** The patterned snack package being held in the ground truth disappears in the prediction; the workspace is then populated with a cluster of indistinct, ill-defined blobs whose category and geometry cannot be determined from the rendered pixels. In the last two frames these blobs adhere to the right gripper and forearm, fusing visually with the manipulator itself.

Analysis. Unlike Case 1, this substitution happens in the open: SC-O1/O2 stay clean and the gripper's trajectory remains plausible (SC-M1/M2 = A, IC-0/IC-1/IC-2 = A), so subject identity is overwritten without an occlusion gate. The blob's eventual fusion with the gripper is what pushes IC-3 no-penetration to Grade C, completing a primary substitution mode that bypasses occlusion entirely.

K Action Following Gallery

This appendix presents representative cases from the action following evaluation, illustrating distinct patterns across the TCR (task completion) and OPS (object preservation) dimensions.

K.1 Case 1: Ideal Success (TCR = 1, OPS = high)

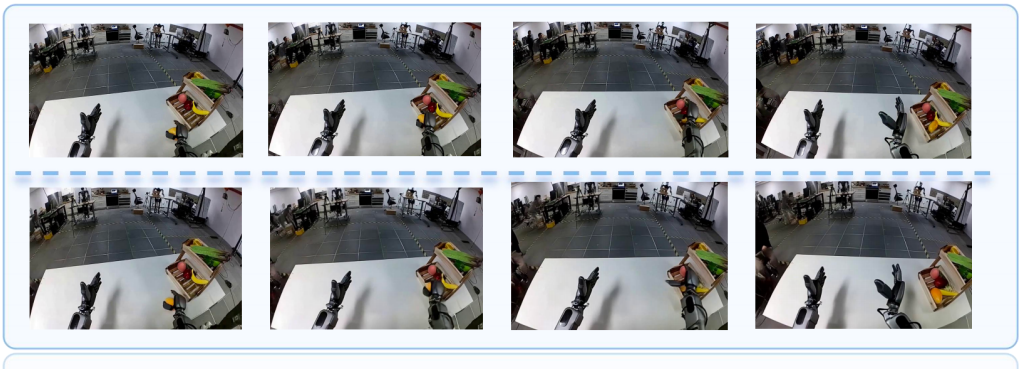


Figure 17: **DreamDojo-2B, episode_0172.** The predicted video accurately follows the instruction: the robot grasps the mango and places it onto the lower shelf of the rack. Objects remain visually coherent throughout, with no deformation or artefacts.

Instruction: *"A GR1 humanoid robot stands in front of a yellow mango and a two-tier rack. The robot extends one arm to grasp the mango from the table. It then lifts the fruit and carefully places it onto the lower shelf of the rack."*

Task Completion (TCR)		*Object Preservation (OPS)*	
TCR	**1 (Completed)**	OPS	**high**
		Confidence	1.00 (16/16)

Analysis. This episode represents the ideal outcome: the world model both completes the manipulation task and preserves object appearance throughout the sequence. The mango is correctly grasped, transported, and placed without any visual artefact, and the arm trajectory closely matches the expected motion. High TCR and high OPS together indicate that the model has learned the correct action–visual outcome mapping for this episode.

K.2 Case 2: Object artefacts Despite Task Completion (TCR = 1, OPS < high)

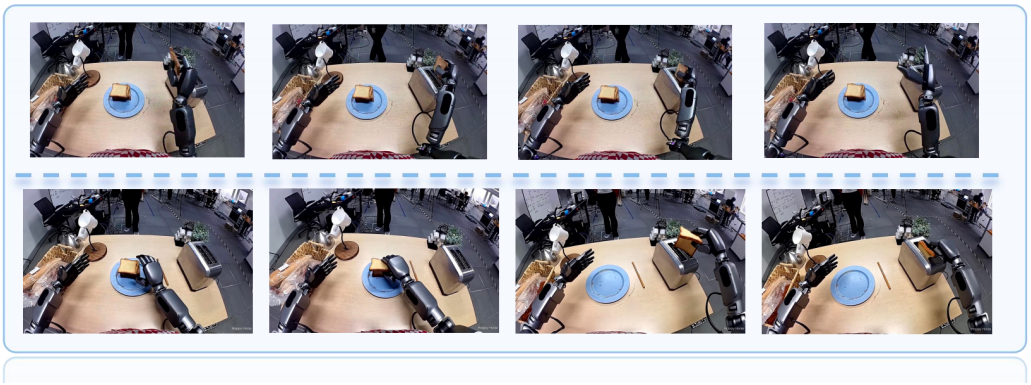


Figure 18: **Happy Horse, episode_0127.** The robot appears to complete the task of placing toast into the toaster, but the bread slice undergoes severe visual distortion during manipulation—multiple slices merge into a single amorphous mass.

Instruction: *"A GR1 humanoid robot stands in front of a kitchen countertop. The robot uses one arm to carefully pick up a slice of toast from a blue plate. It then moves the toast and places it into a nearby toaster."*

Task Completion (TCR)		*Object Preservation (OPS)*	
TCR	**1 (Completed)**	OPS	**medium**
		Confidence	0.69
		Failing frames	9–13 (merging/deformation)

Analysis. This case highlights the independence of TCR and OPS: the robot achieves the task goal (toast reaches the toaster), yet the object undergoes severe metamorphosis during transport. Multiple bread slices fuse into a single undifferentiated mass, a failure mode invisible to a task-completion-only metric. The OPS dimension is essential for detecting such artefacts, as a downstream policy trained on this data would learn a physically impossible manipulation trajectory. The late-frame failures (frames 9–13) correspond to the grasp-and-lift phase, suggesting that the model's object representation collapses precisely when fine-grained hand–object interaction is required.

K.3 Case 3: Dual Failure—Task Not Completed and Objects Distorted (TCR = 0, OPS < high)

Instruction: *"A GR1 humanoid robot stands in front of a tabletop with various objects. The robot uses one arm to grasp a yellow pear from the table. It then carefully places the pear onto the bottom tier of a multi-level white tray."*

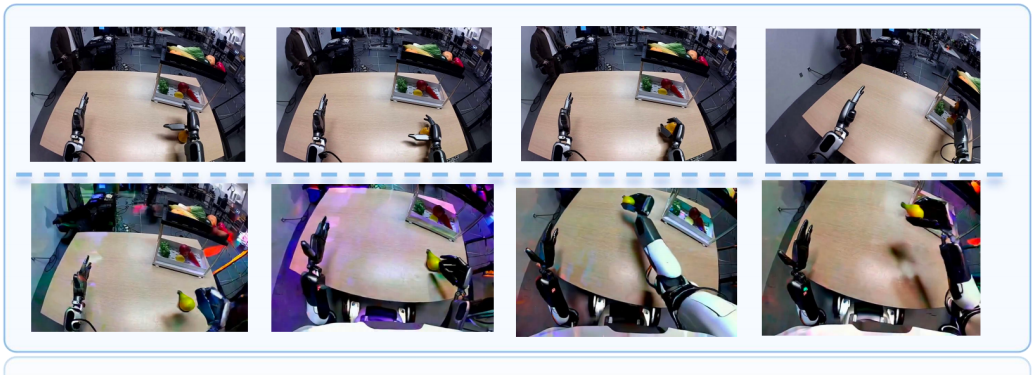


Figure 19: **Cosmos-14B, episode_0021.** The robot fails to place the pear onto the tray, the arm reaches in the wrong direction, and the scene exhibits bizarre chromatic flashing with object deformation throughout.

Task Completion (TCR)		*Object Preservation (OPS)*	
TCR	**0 (Not completed)**	OPS	**medium**
		Confidence	0.62
Action following	Wrong direction	Failing frames	2, 11–16 (flash/deform)

Analysis. This episode represents the most severe failure mode: the model neither completes the task nor preserves object integrity. The robot's arm reaches in a direction inconsistent with the target object, indicating that the action conditioning signal has been largely ignored. Simultaneously, chromatic artefacts and object deformation confirm that the model's visual generation is unstable. Both metrics correctly flag this episode, but their conjunction reveals a qualitatively different failure than either alone—this is not merely a bad action response or a visual glitch, but a fundamental breakdown in the model's ability to produce coherent video conditioned on the given action.

K.4 Case 4: Optimism Bias—Task Labelled as Failed, Model Succeeds (TCR = 0, OPS = high)

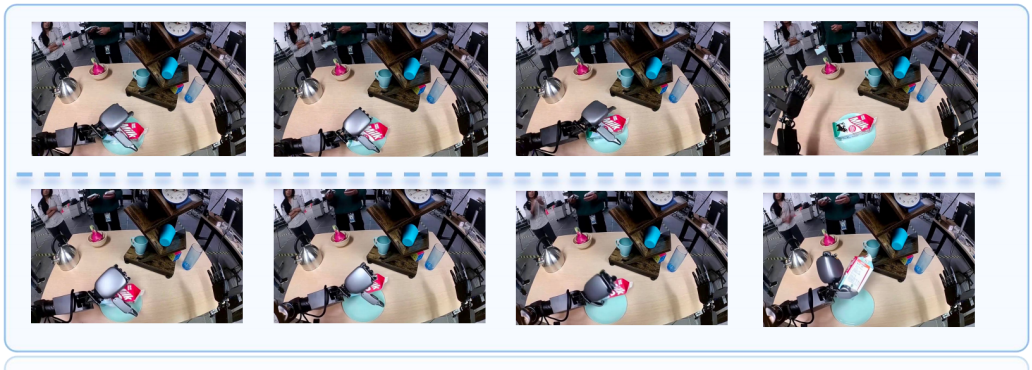


Figure 20: **Wan2.2, episode_0082.** The ground-truth instruction specifies that the robot *fails to lift* the milk carton, yet the predicted video shows the robot successfully grasping and lifting it. Objects remain perfectly preserved—the model generates a physically plausible but *optimistically biased* outcome.

Instruction: *"In a tabletop scene with shelves and various objects, a milk carton sits on a plate in front of a GR1 humanoid robot. The robot extends a single arm to grasp the milk carton with its hand. It then attempts to lift the carton but failed."*

Task Completion (TCR)		*Object Preservation (OPS)*	
TCR	**0 (Not completed)**	OPS	**high**
		Confidence	1.00 (16/16)
GT outcome	**Failure (lift fails)**		
Model outcome	**Success (lift succeeds)**		

Analysis. This episode reveals a subtle but critical failure mode that bridges action following and the optimism bias discussed in Appendix L. The ground-truth demonstration encodes a *failed* manipulation attempt, yet the world model generates a successful outcome: the milk carton is cleanly grasped and lifted with no visual artefact. From a pure visual quality standpoint, the prediction is excellent (OPS = high, 16/16 frames pass). The TCR judge correctly labels this as TCR = 0—but only because the instruction explicitly states the task is to *fail*, making the model's successful prediction a deviation from the prescribed action.

This case exemplifies how world models can override failure-signal conditioning with a learned success prior: having seen predominantly successful manipulations during training, the model generates the statistically common (successful) outcome rather than the prescribed (failed) one. The object preservation is flawless precisely because the model's "success template" includes coherent object appearance—but the action following is fundamentally compromised. This pattern connects directly to the optimism bias analysis in Appendix L, where models systematically ignore failure-inducing action perturbations in favour of generating successful outcomes.

K.5 Summary of Action Following Patterns

Table 22 consolidates the four patterns and their diagnostic implications.

Table 22: Summary of action following patterns observed in the gallery.

Case	Pattern	TCR	OPS	Diagnostic Implication
1	Ideal success	1	high	Correct action–visual mapping; model faithfully follows the instruction
2	Compromised success	1	<high	Task goal achieved but object integrity violated; physical impossibility masked by task metric alone
3	Dual failure	0	<high	Fundamental breakdown in both action conditioning and visual generation
4	Optimism bias	0	high	Model generates successful outcome when instructed to fail; bridges action following and optimism bias

L Optimism Bias Gallery

This appendix first examines failure preservation conditional on nominal action following, then presents a per-perturbation ablation and representative optimism-bias cases.

L.1 Failure Preservation Conditioned on Nominal Action Following

The main results report Physics Adherence, Action-Following Fidelity, and Optimism Bias Detection as separate diagnostic dimensions. In particular, the headline Bias Res. score in Table 2 is ungated: a model can distinguish nominal and perturbed rollouts even when it does not complete the nominal task. To assess failure preservation when the nominal prediction is both minimally coherent and task-successful, we retain a baseline only if its Obj. score is at least 30 on the 0–100 scale and its episode-level TCR is 1. We then evaluate its three paired perturbation rollouts with the same Bias Res. procedure described in Appendix D.3.

Table 23: Conditional Bias Resistance after filtering nominal baselines with Obj.$\geq$ 30 and TCR$=$ 1. Higher values indicate stronger preservation of failure-inducing action outcomes. The ungated scores remain in Table 2.

Model	Conditional Bias Res. (%)↑
Vector-conditioned models	
DreamDojo 2B Pretrain	N/A
DreamDojo 2B GR-1	18.2
DreamDojo 14B Pretrain	14.8
DreamDojo 14B GR-1	14.8
Cosmos GR-1 2B	22.2
Cosmos GR-1 14B	44.4
Text-conditioned models	
Cosmos GR-1 14B I2V	30.8
WAN2.1	85.7
WAN2.2	83.3
Happy Horse	54.5
WanX	48.5
Kling 3.0 Omni	35.9

Table 23 shows that most reportable scores remain below 50 even after filtering for nominal task completion and minimal object consistency, indicating that failure preservation remains weak for many evaluated models. The higher conditional scores of WAN2.1 and WAN2.2 describe perturbation sensitivity within the retained episodes, not overall reliability across the three dimensions. N/A denotes a model without a reportable score after the nominal-baseline filter.

L.2 Per-Perturbation Ablation

Table 24 reports the optimism bias rate (MA-9 = Y or Y?) for each perturbation type, averaged across models and broken down per model. Figure 21 visualizes the same data.

Table 24: Optimism bias rate (%) by perturbation type. Higher values indicate the model more frequently ignores the perturbation. Rates are computed as the fraction of annotated samples where MA-9 $\in$ {Y, Y?}.

Perturbation	All	DreamDojo-2B	DreamDojo-14B	Happy Horse	Wan2.1
`approach_overshoot`	73%	50%	92%	100%	50%
`contact_oscillation`	73%	100%	93%	50%	50%
`grip_force_weak`	66%	100%	73%	70%	20%
`premature_release`	62%	100%	78%	40%	30%
`grip_carry_slip`	50%	0%	–	100%	100%
`wrist_tilt_grasp`	35%	0%	40%	100%	0%

Key observations.

1. **No perturbation universally succeeds**: every perturbation type is ignored by at least one model at $\geq$50% rate, confirming that optimism bias is not limited to specific failure modes.
2. **Model-perturbation interaction**: DreamDojo-2B shows 100% bias on force/timing perturbations (`grip_force_weak`, `premature_release`, `contact_oscillation`) but 0% on spatial ones (`grip_carry_slip`, `wrist_tilt_grasp`). Happy Horse shows the reverse pattern. This suggests different models learn different aspects of the action space.
3. **Force-based perturbations are hardest to detect**: `grip_force_weak` (66% avg bias) modifies only the magnitude of hand joints without changing trajectory geometry, making it the most "invisible" perturbation from a visual standpoint.

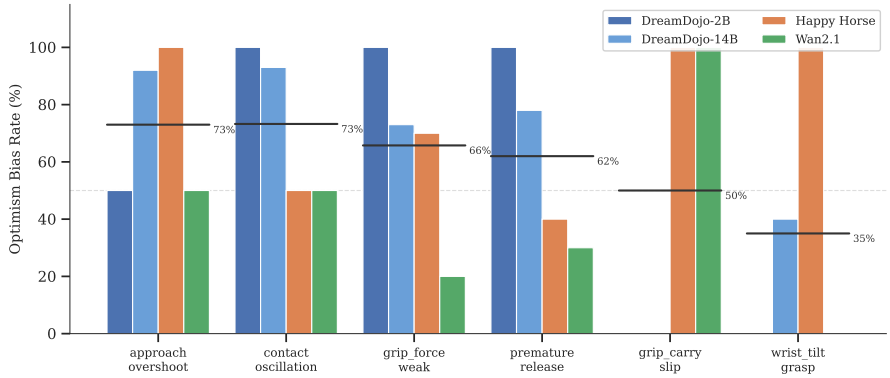


Figure 21: Per-perturbation optimism bias rate broken down by model. `grip_force_weak` and `contact_oscillation` trigger the highest bias in DreamDojo-2B (100%), while `wrist_tilt_grasp` is the least biased perturbation overall (35%). Note that model sensitivity varies strongly by perturbation type: DreamDojo-2B shows 0% bias on `grip_carry_slip` and `wrist_tilt_grasp`, while Happy Horse shows 100% on those same types.

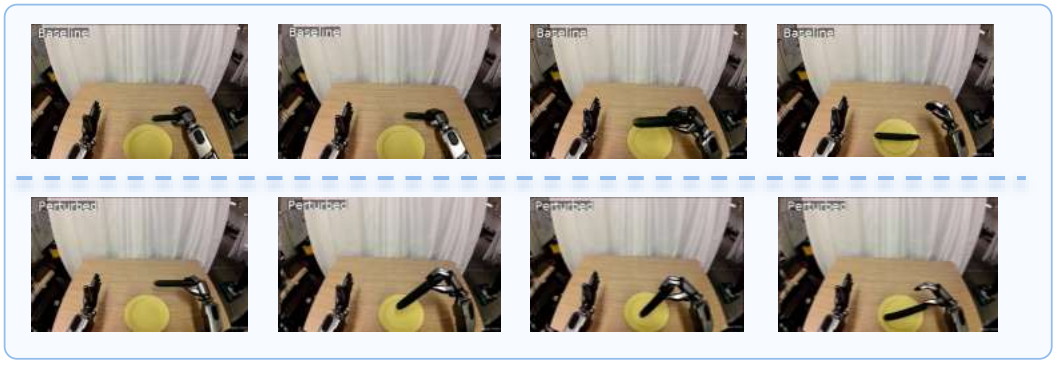


Figure 22: **Happy Horse, task_023.** The baseline prediction is visually excellent (MA-1 = 4/5) with fully plausible physics, yet the perturbed video is indistinguishable from baseline despite a failure-inducing perturbation. The model generates physically coherent, task-successful outcomes regardless of the action input.

L.3 Case 1: High-Quality Complete Bias

Source: `human_annotation_happyhorse_i2v/videos/task_023/`

Module A (Perturbation Sensitivity)		*Module B (Task Success)*	
Baseline Quality (MA-1)	4/5 (High similarity)	Baseline Completion (MB-1)	Mostly complete
Robot Action (MA-2)	Mostly rational	Key Action Correct (MB-2)	Mostly correct
Temporal Fluency (MA-3)	Mostly smooth	Phys. Plausibility (MB-3)	Fully plausible
Perturbation Impact (MA-4)	**No effect**	GT Deviation (MB-4)	High consistency
Perturbed Realism (MA-5)	Mostly realistic	Perturbed Completion (MB-6)	**Mostly complete**
Quality Degradation (MA-6)	Minor	Final State Deviation (MB-7)	Identical
Optimism Bias (MA-9)	**Y (Full)**	**False Success (MB-9)**	**Y (False success)**

Analysis. This is the most dangerous failure mode: the model produces physically coherent, visually convincing videos that look correct by any quality metric, while completely ignoring the failure-

inducing action. A downstream policy trained on such data would never encounter the failure mode encoded in the perturbation.

L.4 Case 2: Scale Does Not Fix Bias



Figure 23: **DreamDojo-2B, task_024.** Despite lower visual quality than larger models, this 2B model achieves high baseline-GT alignment (MA-1 = 4/5) on this episode while completely ignoring the perturbation. The perturbed prediction shows task completion identical to baseline.

Module A		*Module B*	
Baseline Quality (MA-1)	4/5 (High similarity)	Baseline Completion (MB-1)	Fully complete
Perturbation Impact (MA-4)	**No effect**	Phys. Plausibility (MB-3)	Mostly plausible
Quality Degradation (MA-6)	None	Perturbed Completion (MB-6)	**Mostly complete**
Optimism Bias (MA-9)	**Y (Full)**	False Success (MB-9)	**Y**

Analysis. Even a 2B model can produce episode-specific perfect alignment with ground truth while exhibiting full optimism bias. This demonstrates that the success prior operates independently of general generation capability: the model has "memorized" this task's success pattern so strongly that no action perturbation can override it.

L.5 Case 3: Low Quality + Full Bias

Source: `human_annotation_dreamdojo_2b_gr1/videos/task_029/`

Module A		*Module B*	
Baseline Quality (MA-1)	2/5 (Low)	Baseline Completion (MB-1)	Mostly complete
Perturbation Impact (MA-4)	**No effect**	Phys. Plausibility (MB-3)	**Completely implausible**
Quality Degradation (MA-6)	Minor	Perturbed Completion (MB-6)	**Mostly complete**
Motion Coherence (MA-8)	Consistent	Final State (MB-7)	Identical
Optimism Bias (MA-9)	**Y (Full)**	Unexpected Success (MB-8)	No

Analysis. This case separates optimism bias from physical competence. The model cannot generate physically plausible interactions (MB-3 = "completely implausible"), yet its success prior is strong enough to produce task-completion outcomes regardless. The bias is not a consequence of sophisticated physics understanding; it is a shallow statistical pattern learned from success-only training data.

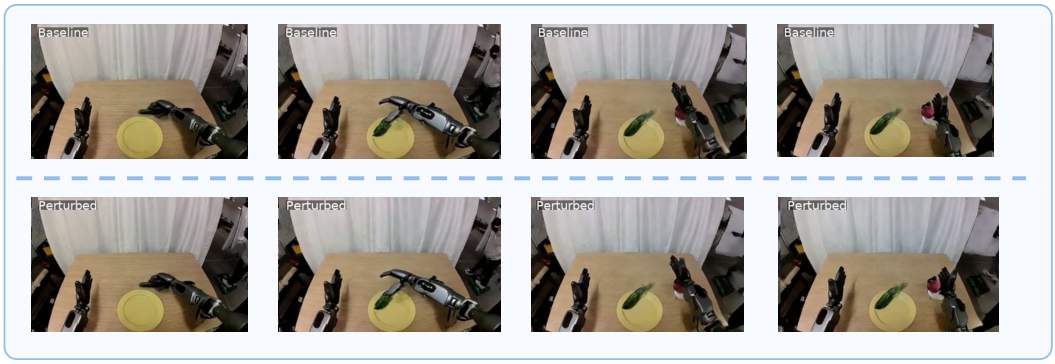


Figure 24: **DreamDojo-2B, task_029.** The baseline itself exhibits poor physical plausibility (penetration, impossible deformation), yet the model still generates task-successful predictions under perturbation. Bias persists even when the model's physics understanding is clearly deficient.

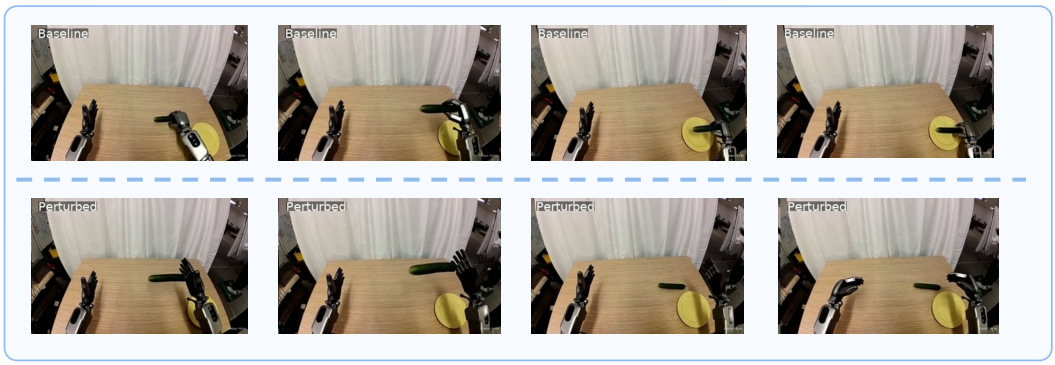


Figure 25: **Happy Horse, task_022.** The model correctly responds to the perturbation: the perturbed prediction shows clear task failure (MB-6 = "not completed") with significant trajectory divergence from baseline. This demonstrates that the same model architecture *can* propagate failure signals in some cases.

L.6 Case 4: Correct Failure Prediction (No Bias)

Source: `human_annotation_happyhorse_i2v/videos/task_022/`

Module A		*Module B*	
Baseline Quality (MA-1)	4/5 (High)	Baseline Completion (MB-1)	Mostly complete
Perturbation Impact (MA-4)	**Significant**	Phys. Plausibility (MB-3)	Mostly plausible
Quality Degradation (MA-6)	Minor	Perturbed Completion (MB-6)	**Not completed**
Motion Coherence (MA-8)	Clearly worse	Final State (MB-7)	**Clear deviation**
Optimism Bias (MA-9)	**N (No bias)**	False Success (MB-9)	**N (Correct failure)**

Analysis. This positive example shows that the same model (Happy Horse) that exhibits full bias in Case 1 can correctly propagate perturbation effects in other episodes. The bias is therefore not a global architectural limitation but rather episode- and perturbation-type dependent, suggesting that certain task structures are more susceptible to prior override than others.

L.7 Case 5: Mild Bias (Insufficient Response)

Source: `human_annotation_happyhorse_i2v/videos/task_011/`



Figure 26: **Happy Horse, task_011.** The model shows a detectable but insufficient response to the perturbation: the perturbed video achieves only partial completion with large final-state deviation from baseline, yet annotators still detect mild optimism bias (MA-9 = Y?) and false success (MB-9 = Y), indicating the model dampens but does not fully propagate the failure signal.

Module A		*Module B*	
Baseline Quality (MA-1)	3/5 (Moderate)	Baseline Completion (MB-1)	Fully complete
Perturbation Impact (MA-4)	**Minor effect**	Phys. Plausibility (MB-3)	Mostly plausible
Quality Degradation (MA-6)	Minor	Perturbed Completion (MB-6)	**Partially complete**
Motion Coherence (MA-8)	Slightly worse	Final State (MB-7)	**Large deviation**
Optimism Bias (MA-9)	**Y? (Mild bias)**	False Success (MB-9)	**Y**

Analysis. Mild bias represents a continuum between full suppression and correct propagation. The model registers the perturbation at a surface level (some trajectory change, partial completion rather than full success) but does not propagate it to the expected physical consequence (complete task failure). This pattern suggests a competition between the action conditioning signal and the learned success prior, with the prior partially winning.

L.8 Case 6: Generation Inadequacy Masking as Low Bias

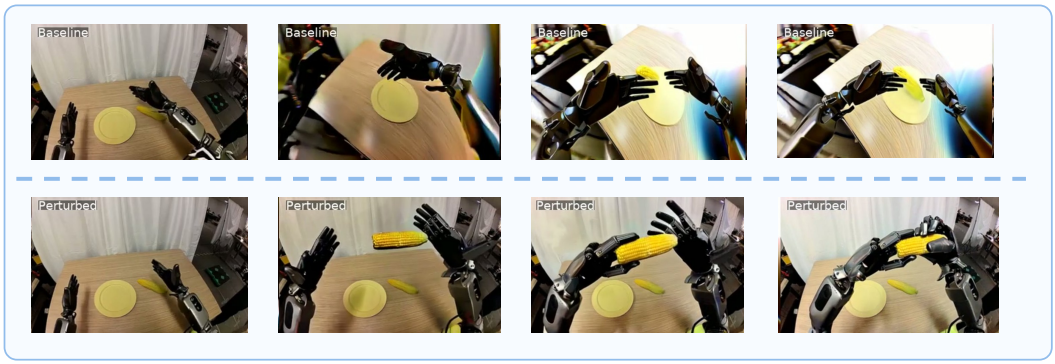


Figure 27: **Wan2.1, task_017.** The model shows no full optimism bias, but not because it correctly follows actions: the baseline itself completely fails (MA-1 = 1/5, MB-1 = "not completed"), so there is no "success" for the model to hallucinate under perturbation. Low bias here reflects incapability, not fidelity.

Source: `human_annotation_wan21_i2v_14b/videos/task_017/`

Module A		*Module B*	
Baseline Quality (MA-1)	**1/5 (Different)**	Baseline Completion (MB-1)	**Not completed**
Perturbation Impact (MA-4)	No effect	Phys. Plausibility (MB-3)	Completely implausible
Quality Degradation (MA-6)	None	Perturbed Completion (MB-6)	Cannot judge
Optimism Bias (MA-9)	**Y? (Mild)**	False Success (MB-9)	N/A (baseline fails)

Analysis. This case illustrates why the diagnostic scores must be interpreted together. Wan2.1 receives a favorable perturbation-response score, but its nominal rollout in this case has poor physical coherence and does not complete the task. The ungated optimism-bias score alone therefore cannot establish faithful action conditioning. Reporting physical coherence and nominal task success alongside perturbation response makes this ambiguity visible without imposing a scoring gate.

L.9 Summary of Bias Patterns

Table 25 consolidates the six failure patterns and their diagnostic implications.

Table 25: Summary of optimism bias patterns observed in the failure gallery.

Case	Pattern	Bias	Quality	Diagnostic Implication
1	High-quality full bias	Y	High	Most dangerous: indistinguishable from correct behavior by any quality metric
2	Scale-independent bias	Y	High	Model scale improves quality but not action sensitivity
3	Low-quality full bias	Y	Low	Bias is independent of physics competence; shallow statistical pattern
4	Correct failure (no bias)	N	High	Same model can propagate failure in some cases; bias is task-dependent
5	Mild bias (dampened response)	Y?	Medium	Competition between action signal and success prior; partial suppression
6	Incapability masking as low bias	Y?	Very low	Low bias from generation failure, not action fidelity; interpret with nominal quality

M Computation Cost

Table 26 summarizes the video generation cost across evaluated models. All measurements are on NVIDIA A100 80GB GPUs.

Table 26: Video generation cost per model on the full MiraBench test set.

Model	Params	GPU Mem	Time/video
DreamDojo-14B	14B	∼75 GB	∼3.5 min
DreamDojo-2B	2B	∼18 GB	∼1.2 min
Cosmos-14B	14B	∼72 GB	∼3.0 min
Cosmos-2B	2B	∼16 GB	∼1.0 min
Wan2.1-14B	14B	∼65 GB	∼4.0 min
Wan2.1-1B	1B	∼12 GB	∼0.8 min

Evaluation pipeline. After video generation, running the full MiraBench evaluation (all three diagnostic dimensions) on a single model takes approximately 1.5 hours on one A100 GPU, dominated by VLM inference across the three evaluators.

End-to-end. Evaluating a new model on MiraBench (both modalities) costs approximately **3–12 A100-GPU-hours** depending on model size, making the benchmark practical for routine evaluation without dedicated cluster access.

N Comparison with Existing Benchmarks

Table 27 provides a systematic feature comparison between MiraBench and representative existing evaluation frameworks for video generation and world models.

Table 27: Feature comparison of MiraBench with existing benchmarks. ✓: supported; ✗: not supported; ○: partially supported.

Feature	*MiraBench*	*WorldArena*	*WorldSimBench*	*WorldModelBench*	*WorldScore*	*VBench*	*EvalCrafter*	*PhyGenBench*	*Physion++*
Evaluation Target									
Robot manipulation focus	✓	✓	○	○	✗	✗	✗	✗	✗
Vector-conditioned models	✓	✓	✓	○	✗	✗	✗	✗	✗
Instruction-conditioned models	✓	✗	✗	✓	✓	✓	✓	✓	✗
Multiple embodiments	✓	✓	○	✗	✗	✗	✗	✗	✗
Physics Evaluation									
Physical consistency	✓	○	○	✓	○	○	✗	✓	✓
Quantitative physics law	✓	✗	✗	✗	✗	✗	✗	○	✓
Reference-free assessment	✓	✗	✗	✗	✗	✗	✗	○	✓
Action Faithfulness									
Action-following fidelity	✓	○	○	✗	✗	✗	✗	✗	✗
Failure-regime testing	✓	✗	✗	✗	✗	✗	✗	✗	✗
Implicit failure perturbations	✓	✗	✗	✗	✗	✗	✗	✗	✗
Optimism bias detection	✓	✗	✗	✗	✗	✗	✗	✗	✗
Evaluation Methodology									
Dual action modality	✓	✗	✗	✗	✗	✗	✗	✗	✗
Human annotation validation	✓	○	○	✓	○	✓	✓	✗	✓
Multidimensional diagnostics	✓	✗	✗	✗	✗	○	✗	✗	✗

Key differentiators. Three capabilities are unique to MiraBench among existing benchmarks:

- **Failure-regime evaluation.** All existing benchmarks evaluate world models in the normal operating regime, measuring average-case performance on successful demonstrations. MiraBench is the first to systematically probe the failure regime through implicit perturbations, testing whether models correctly propagate failure-inducing action signals rather than defaulting to their success prior.

- **Dual action modality.** MiraBench evaluates its diagnostic dimensions under precise (motor command) and descriptive (natural language) action conditioning, enabling diagnosis of whether failures originate in motor encoding or semantic conditioning. No existing benchmark tests both modalities within a unified framework.

- **Complementary diagnostics.** Rather than producing a single aggregate score, MiraBench reports physics adherence, action following, and optimism-bias detection separately to identify *where* a model fails, not just *how much*. Weak physics adherence points to incoherent generation; weak action following indicates poor nominal task execution; weak failure preservation indicates insufficient response to failure-inducing inputs. These scores should be considered together because an ungated failure-preservation score can be misleading when nominal generation is poor. This diagnostic interpretability is absent from all existing frameworks.

O BROADER IMPACT STATEMENT

O.1 POSITIVE IMPACTS.

MiraBench addresses a critical gap in the evaluation of world models for embodied AI. By shifting evaluation from visual quality toward action-following fidelity and failure prediction, we aim to prevent a class of silent deployment failures: world models that generate visually compelling but physically incorrect training data for robot policies. Policies trained on such data may behave unpredictably in safety-critical scenarios, as they have never been exposed to realistic failure modes during training. Our benchmark provides the diagnostic tools to identify this risk before deployment, potentially reducing real-world accidents in robotic manipulation systems.

The multidimensional diagnostic structure enables targeted model improvement rather than undirected scaling, potentially reducing the computational resources wasted on approaches that improve visual quality without improving simulation fidelity. By releasing our evaluation framework, perturbation taxonomy, and annotated datasets, we lower the barrier for the community to audit world models before using them as data generators.

O.2 POTENTIAL NEGATIVE IMPACTS.

Environmental impact. The benchmark evaluation pipeline requires substantial GPU computation for video generation (approximately 8 A100-hours per model for the full test set). However, this cost is incurred once per model evaluation and is modest compared to the training cost of the world models themselves. The VLM evaluator inference takes approximately 1.5 GPU-hours per model for the full benchmark. We provide efficiency guidelines and support for partial evaluation on reduced test sets to accommodate resource-constrained settings.

Impact on labor. Human annotation is used for evaluator calibration and validation rather than as the primary evaluation mechanism. All annotators are compensated at rates above local minimum wage, work under non-coercive conditions, and are informed about the research purpose of the annotation task.

Dual-use considerations. World models that faithfully simulate failure could, in principle, be used to generate adversarial training data or to identify vulnerabilities in robotic systems. We believe the benefits of transparent evaluation substantially outweigh this risk: understanding where world models fail is a prerequisite for making them safe, and concealing these failure modes would not prevent their exploitation by motivated actors with access to the same models.

P ETHICAL CONSIDERATIONS

Human annotation. All human annotators were recruited through a professional data annotation platform and compensated at rates exceeding local minimum wage standards. Annotators were informed of the research purpose prior to participation, provided written consent, and were free to withdraw at any time without penalty. The annotation task involves watching robot manipulation videos recorded in controlled laboratory environments; no personally identifiable information, violent content, or otherwise distressing material is present in any video. Annotator demographics are not collected beyond professional qualification verification, in compliance with data minimization principles. Institutional Review Board (IRB) approvals were obtained.

Data privacy. All videos in MiraBench are recorded in laboratory settings with robotic manipulators operating on inanimate objects (fruits, containers, cloths). No human subjects, faces, or private spaces appear in any frame. The GR-1 and Lingchu datasets used as source material are collected under their respective institutional protocols with appropriate permissions for research use.

Consent and licensing. We confirm compliance with the terms of use for all datasets employed: GR-1 data is used under NVIDIA's research license; DROID data is used under its open research license; Lingchu data is used with institutional permission. MiraBench evaluation code and annota-

tions will be released under a permissive open-source license (Apache-2.0) to maximize community benefit while maintaining clear attribution requirements.

Environmental considerations. The full MiraBench evaluation pipeline requires approximately 8 A100-GPU-hours per model for video generation and approximately 1.5 GPU-hours per model for VLM evaluation. We provide configuration options for reduced evaluation (subset of tasks or episodes) to accommodate researchers with limited computational resources, and we release pre-generated evaluation videos for all models reported in this paper to eliminate redundant computation.

Intended use and misuse prevention. MiraBench is designed for evaluating and improving world models in research settings. We discourage its use as the sole criterion for deploying world models in safety-critical applications without additional real-world validation. The implicit failure perturbation taxonomy is published for transparency and reproducibility; while it could theoretically inform adversarial attacks on world-model-based systems, we believe open evaluation standards are essential for building trustworthy embodied AI and that security-through-obscurity is not a viable long-term strategy.