PAWBENCH: HOW FAR ARE WE FROM PROBABILISTICALLY ALIGNED WORLD MODELING?

Anonymous authors
Paper under double-blind review

ABSTRACT

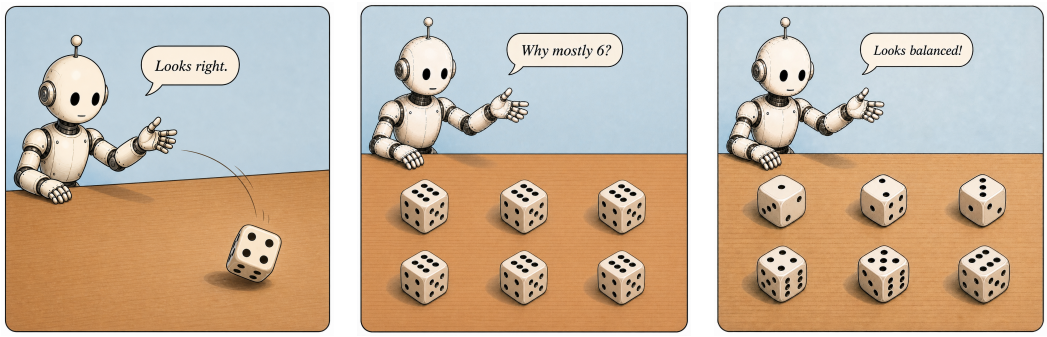
Recent video generation models are increasingly framed as world models. Many physical processes can unfold in more than one valid way. Therefore, a world model should reproduce not only a plausible trajectory, but also the distribution of possible behaviors under the same initial observation and action. We call this distribution-level requirement **probabilistic alignment**. However, existing evaluations largely assess individual-video plausibility and do not test whether repeated generations recover the correct distribution. This raises a central question: **how far are current video generators from probabilistically aligned world modeling?** To answer it, we formalize probabilistic alignment as a distributional criterion for world models and introduce **PAWBench**, a benchmark for evaluating video generators as stochastic samplers of world dynamics. We further introduce **PAWEval**, an outcome-level protocol that converts repeated video rollouts into empirical distributions over possible physical behaviors. Across 50 scenarios and twelve current systems, no model consistently matches the reference probabilities while recovering the range of valid behaviors. Models also respond inconsistently to physical changes while shifting their outcome distributions in response to irrelevant visual or textual cues. Having established this gap, we test whether language prompts, initial noise sampling, or model training can reshape the model's predictive distribution. Prompt engineering is limited by errors in both predicting outcome distributions and realizing requested outcomes, while initial noise sampling broadens exploration at a fixed rollout budget. Fine-tuning shifts generated outcome frequencies toward the training proportions, but does not reliably adapt them to changes in physical state. We believe our work can serve as a foundation for future efforts to move towards probabilistically aligned world modeling.

1 INTRODUCTION

Recent advances in video generation have substantially improved instruction following, visual quality, and temporal coherence (Google, 2025; Kling AI, 2026; ByteDance Seed, 2026b; Lightricks, 2026a; MiniMax, 2026). Given an observation and a prompt or action-like control signal, current systems can generate plausible continuations of a scene. Together, these advances (Brooks et al., 2024; Ball et al., 2025; NVIDIA, 2026) have strengthened the case for viewing video generators not only as content creation tools, but also as visual world models.

Yet a world model must do more than render visually plausible rollouts. Because an observation reveals only part of the world state and an action does not specify all of its consequences, the generated rollouts must remain consistent with the scene's underlying geometry and dynamics (World Labs Team, 2026). When the underlying process is stochastic, the same initial observation and action can admit multiple physically valid futures rather than a single correct continuation (Babaeizadeh et al., 2018; Denton & Fergus, 2018). A model should therefore capture the conditional distribution over these futures, not merely render one plausible sample. We call this requirement **probabilistic alignment**: under a fixed initial observation and action, the induced distribution should reflect both which futures are physically possible and how likely each one is. Probabilistic alignment matters for interaction and planning (Ravi et al., 2025), because decisions depend on both the possible consequences of an action and their relative likelihoods.

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.



(a) One plausible future (b) Probabilistically unaligned world (c) Probabilistically aligned world

Figure 1: **One plausible future is not enough.** While individual rollouts may appear physically plausible, repeated generations under identical inputs often reveal probabilistic misalignment by collapsing into a narrow subset of outcomes. Rather than judging models on isolated videos, PAWBench evaluates the induced distribution and outcome support across repeated rollouts.

Evaluation, however, still predominantly treats generated videos as isolated outputs. Existing benchmarks score individual rollouts on visual quality, temporal coherence, text alignment, and basic physical plausibility (Huang et al., 2024; 2025; Bansal et al., 2025; Li et al., 2025). Success on these standard criteria often obscures whether a model collapses to a highly narrow subset of valid outcomes or assigns probability mass in the wrong proportions under identical conditioning inputs. As illustrated in Fig. 1, single sample evaluation cannot distinguish a truly probabilistically aligned world model from one that merely memorizes a few visually plausible endpoints. This leaves a central question: *how far are we from probabilistically aligned world modeling?*

To answer this question, we introduce **PAWBench** (Probabilistically Aligned World Bench), a diagnostic benchmark for evaluating whether video generators model the distribution of possible futures. PAWBench contains 50 scenarios spanning eight mechanism groups, organized into two complementary suites. *PAW-Calibration* covers 25 scenarios with analytically specified or symmetry-derived reference distributions, such as tossing a coin or spinning a wheel. *PAW-Coverage* covers 25 scenarios whose valid terminal outcomes can be enumerated but whose relative probabilities cannot be reliably specified, such as rolling a bowling ball or flipping a bottle. For each scenario, the source image and action prompt are held fixed across repeated rollouts. **PAWEval** maps readable, in-schema rollouts to terminal outcomes and records the remaining cases as outcome-readout failures. In PAW-Calibration, PAWBench compares the resulting empirical distribution with the reference probabilities; in PAW-Coverage, it measures recovery of the valid support.

We benchmark twelve current video generation models on PAWBench. No model achieves accurate outcome probabilities, broad coverage of valid futures, and reliable performance across scenes simultaneously. Scaling the rollout budget does not resolve these distributional gaps, and our automated metrics closely align with human judgments. Through controlled interventions, we further reveal that model distributions fail to respond adequately to causal physical changes, yet incorrectly shift when non-causal visual cues are altered. These findings demonstrate that despite their visual realism, current systems fail to internalize the underlying probabilistic structure of physical events, operating as biased visual samplers rather than probabilistically aligned world models.

We next move from measuring probabilistic alignment to asking whether it can be improved. We examine three points of intervention: prompt engineering, initial noise sampling, and parameter-level fine-tuning. Language uses text prompts or VLM predictions to explicitly specify target futures. Sampling leaves the prompt fixed and varies initial noise to broaden outcome exploration within the existing generator. Fine-tuning directly updates model weights, testing whether training can reshape the learned predictive distribution and adapt outcome probabilities across physical states.

Our main contributions are as follows.

- We formalize **probabilistic alignment** as a distributional criterion for world models and operationalize it through **PAWBench**, a 50-scenario diagnostic benchmark spanning eight physical mechanism groups under fixed initial observations and actions. **PAWEval** converts

repeated video rollouts into empirical outcome distributions, enabling PAW-Calibration and PAW-Coverage to test probability-mass alignment and valid-support recovery, respectively.

- We benchmark current video generation models and reveal a failure hidden by current single-sample evaluation: plausible rollouts do not imply the correct distribution over possible futures. This gap persists beyond finite sampling and automated outcome readout.
- We probe probabilistic alignment from external guidance to model learning: we use language to predict and steer outcomes, initial noise sampling to broaden finite-budget exploration, and fine-tuning to reshape the learned distribution. These results distinguish inference-time steering and exploration from changes to the model's learned predictive distribution.

2 Probabilistically Aligned World Modeling

World models represent distributions over possible futures. Let x denote an initial observation and a an action. When the underlying process is inherently stochastic, the same initial observation and action may lead to different physically valid outcomes. We therefore describe a video world model M through the conditional distribution $P_M(\tau \mid x, a)$ over the future trajectories τ it generates (World Labs Team, 2026). Probabilistic alignment concerns both which futures this distribution supports and the relative probability it assigns to each of these possible trajectories.

From a plausible future to an aligned distribution. Producing one plausible continuation meets only the weakest requirement: it shows that the model can realize one future. ***Support alignment*** asks whether it can realize the distinct outcomes available under the same condition, rather than collapsing to a subset. ***Probability-mass alignment*** is stronger: it asks whether those outcomes occur in the right proportions. A model can therefore be diverse without being aligned; covering every outcome does not guarantee assigning the correct probability to each.

To define these two levels of alignment, we map each generated trajectory to the terminal outcome it realizes. Let $\mathcal{Y}$ be the finite set of terminal outcomes, and let $g(\tau) \in \mathcal{Y}$ denote the outcome realized by trajectory τ. The model's trajectory distribution then induces

$$p_M(y \mid x, a) = \Pr_{\tau \sim P_M(\cdot \mid x, a)}[g(\tau) = y] .$$

This outcome distribution captures how much probability the model assigns to each possible result, while abstracting away differences among trajectories that reach the same result. When a reference distribution q over $\mathcal{Y}$ is available, probability-mass alignment requires $p_M(\cdot \mid x, a) = q$. When only the set of possible outcomes is known, support alignment requires $\mathrm{supp}\big(p_M(\cdot \mid x, a)\big) = \mathcal{Y}$.

3 PAWBench

3.1 Benchmark Design

PAWBench evaluates a video generation model through repeated rollouts conditioned on the same initial observation and action. Each benchmark item provides a source image representing the initial observation x, an action prompt specifying the action a, a finite set of valid terminal outcomes $\mathcal{Y}$, and, when available, a reference distribution q over $\mathcal{Y}$. The model is queried K times with the same (x, a) pair, and the resulting videos are mapped to terminal outcomes and aggregated into an empirical outcome distribution $\hat{p}_M$. Keeping the source image and action prompt fixed ensures that variation across rollouts reflects the model-induced future distribution rather than changes in the model inputs.

As shown in Fig. 2, PAWBench contains 50 scenarios spanning eight mechanism groups, divided into **PAW-Calibration** and **PAW-Coverage**. **PAW-Calibration** includes 25 scenarios whose valid terminal outcomes have analytically specified or symmetry-derived reference distributions, such as tossing, rotation, routing, and draw-style randomizers. It tests whether $\hat{p}_M$ approaches the reference distribution q, or instead concentrates probability mass on a biased subset of valid futures. **PAW-Coverage** includes 25 scenarios whose valid outcomes can be enumerated but whose probabilities cannot be reliably specified, including collision, stability, agent interaction, and material transition scenarios. It tests whether repeated rollouts recover the qualitatively distinct valid futures in $\mathcal{Y}$. Detailed benchmark composition and outcome-set cardinalities are summarized in Fig. 8.



Figure 2: **PAWBench scenario taxonomy.** PAWBench spans eight mechanism groups across two subsuites: *PAW-Calibration*, featuring analytically specified reference distributions, and *PAW-Coverage*, featuring complex stochastic interactions evaluated via valid-support coverage. Fixing the initial image and action prompt, each scenario scores repeated model rollouts by their terminal outcomes.

This division separates two failures that single-sample evaluation cannot distinguish. PAW-Calibration measures probability misallocation: every rollout may show a plausible die roll or spinner motion, yet one outcome may occur far too often. PAW-Coverage measures missing outcomes: individual rollouts may look plausible even though some valid futures never appear. We therefore report calibration and coverage separately rather than combine them into a single score.

3.2 Scenario Curation

PAWBench relies on repeated-rollout statistics, so each scenario must make the induced future distribution well defined. We therefore curate scenarios around three requirements. First, stochasticity must arise from a visible physical mechanism, rather than from ambiguity in the action prompt or hidden initial conditions. Second, the action prompt must specify one atomic intervention whose completion can be judged from the generated video. Third, the terminal outcomes must form a finite, visually distinguishable set so that rollouts can be consistently mapped into $\mathcal{Y}$. We manually curate and review each of the 50 scenarios against these requirements.

For each scenario, we generate one source image (Google DeepMind, 2025; OpenAI, 2026b) to represent the initial observation and use one action prompt to specify the action across all rollouts. We list the valid terminal outcomes and specify when a rollout fails to complete the intended physical process. For PAW-Calibration, we also derive a reference distribution from analytic reasoning or physical symmetry. We review and finalize the source image, action prompt, outcome set, failure criteria, and reference distribution before evaluating any model. Appendix C.2 describes the construction process and the quality-control checks used to finalize each scenario.

3.3 Evaluating Distributions over Possible Futures

To estimate the distribution induced by a video generator, we introduce **PAWEval**, an outcome-level evaluation protocol for repeated rollouts. As illustrated in Fig. 9, we generate K video rollouts under the same (x, a) pair. Each scenario has an outcome rubric that specifies which terminal physical result to identify, how valid results map to $\mathcal{Y}$, and when to return an outcome-readout failure. Gemini 3.5 Flash (Google DeepMind, 2026) applies this rubric to each rollout and returns either a terminal-outcome label in $\mathcal{Y}$ or an outcome-readout failure. We normalize the assigned outcome labels to obtain the conditional empirical distribution $\hat{p}_M$; readout failures are reported separately because they do not represent physical outcomes in $\mathcal{Y}$ and cannot be counted as additional possible futures.

PAW-Calibration compares this conditional distribution with the reference distribution q using total variation distance (TVD), half the ℓ_1 distance between the two categorical distributions (Gibbs & Su, 2002). PAW-Coverage instead measures valid-support recovery as the fraction of valid outcomes

Table 1: **Main PAWBench results.** PAW-Calibration reports conditional TVD ($\times 100$, $\downarrow$): a 70/30 model distribution against a 50/50 reference scores 20, while an exact match scores 0. PAW-Coverage reports valid-support recovery (%, $\uparrow$). Avg. is computed over passing scenes; SPR is the percentage of all 25 scenes that pass. and indicate the best and second-best conditional scores; NA indicates that no scene in the corresponding mechanism group passes the gate.

Model	PAW-Calibration: TVD $\times 100$ ($\downarrow$)						PAW-Coverage: Coverage (%) ($\uparrow$)					
	Avg.	SPR	Toss	Rot.	Rout.	Draw	Avg.	SPR	Coll.	Stab.	Agent	Mat.
HappyHorse	43.1	92.0%	40.1	27.5	41.9	59.5	47.1	100.0%	45.0	36.8	49.0	58.3
Veo3.1 Fast	35.4	88.0%	37.1	12.2	32.4	55.0	41.8	100.0%	39.8	21.1	53.3	44.2
Kling 3 Std.	34.9	92.0%	29.9	20.3	38.7	46.0	52.8	88.0%	52.7	40.0	44.3	77.5
Seedance 2	30.5	100.0%	32.7	15.1	31.1	40.5	50.9	84.0%	48.6	50.0	44.3	67.5
Wan2.7	26.3	92.0%	26.0	12.2	26.8	36.5	50.0	96.0%	60.3	50.5	37.9	53.3
Wan2.2	26.3	64.0%	22.4	15.8	23.2	45.9	63.4	92.0%	76.8	56.7	55.0	58.3
LTX-2.3 Dev	30.1	24.0%	29.3	11.0	NA	41.0	**71.7**	72.0%	**82.7**	50.0	59.5	**90.0**
LTX-2.5 Dev	27.1	36.0%	25.3	9.5	34.9	NA	61.5	76.0%	69.1	37.5	55.7	75.8
Cosmos 3 Super I2V	**20.5**	80.0%	**19.4**	**3.5**	**17.7**	35.5	55.2	92.0%	56.7	31.7	49.4	81.7
LingBot-Video-MoE	41.8	72.0%	31.8	13.4	44.4	60.8	58.8	100.0%	62.2	**61.8**	46.5	72.5
LingBot-World V2	21.9	36.0%	23.6	27.0	NA	**12.3**	54.9	56.0%	46.7	25.0	**62.4**	66.7
MiniMax H3	24.2	68.0%	22.0	14.0	28.9	26.5	48.7	92.0%	59.5	39.5	36.4	58.3

observed across repeated rollouts. Both metrics use readable, in-schema outcomes; the outcome-readout gate separately determines whether a scene is scoreable. Appendix D provides the full metric definitions, gate, and aggregation rules used to compute the reported results.

4 Evaluation on PAWBench

4.1 Evaluation Setup

We evaluate twelve video generators on PAW Calibration and PAW Coverage (Tab. 1). These span proprietary and open systems: HappyHorse (Alibaba Cloud Community, 2026), Veo3.1 Fast (Google, 2025), Kling 3 Std. (Kling AI, 2026), Seedance 2 (ByteDance Seed, 2026a), Wan2.7 (Alibaba Cloud, 2026), Wan2.2 (Wan et al., 2025), LTX 2.3 Dev (Lightricks, 2026b), LTX 2.5 Dev (HaCohen et al., 2026; Lightricks, 2026a), Cosmos 3 Super I2V (Agarwal et al., 2025; NVIDIA, 2026), LingBot Video MoE (Ma et al., 2026), LingBot-World V2 (Gao et al., 2026), and MiniMax H3 (MiniMax, 2026). For each PAWBench scenario, we sample $K = 50$ independent rollouts per model using a fixed source image, action prompt, and default inference configuration. A scene passes the outcome readout gate if at least 20 of its 50 rollouts yield readable, in schema outcomes. Scene Pass Rate (SPR) measures the percentage of the 25 scenes per track successfully passing this strict evaluation gate to ensure robust analysis, with TVD and Coverage computed only over scenes that pass the gate.

4.2 Benchmark Results

Current video generators remain far from probabilistically aligned world modeling. As shown in Tab. 1, no model combines well-aligned probability mass, broad support recovery, and reliable performance across scenes. Cosmos 3 Super I2V achieves the lowest Calibration TVD, but only 80.0% of its Calibration scenes pass the outcome-readout gate. LTX-2.3 Dev attains the highest Coverage average, yet that average is computed over the 72.0% of its Coverage scenes that pass. Conversely, Seedance 2 and LingBot-Video-MoE pass every scene on Calibration and Coverage, respectively, but neither leads the corresponding conditional metric. Conditional alignment and scene-level reliability therefore capture distinct limitations of current models and must be read together. We provide qualitative results and visual examples in Appendix G.1.

Recovering valid futures is not the same as assigning them the right probability. The models that recover broad support are not those that best match the reference probabilities, and this separation persists across physical mechanisms. A generator may expose many plausible outcomes while allocating probability mass incorrectly, or closely match the frequencies of observed outcomes while leaving valid alternatives unseen. PAW-Coverage and PAW-Calibration therefore expose complementary failures: recovering the valid support is necessary, but it does not establish that probability mass is correctly allocated among the outcomes within that support.

Table 2: **Impact of post-training on probabilistic alignment.** Comparisons across checkpoints offer exploratory observations on how post-training adaptations affect alignment. PT denotes pre-trained.

Model	PAW-Calibration: TVD ×100 (↓)						PAW-Coverage: Coverage (%) (↑)					
	Avg.	SPR	Toss	Rot.	Rout.	Draw	Avg.	SPR	Coll.	Stab.	Agent	Mat.
LTX-2.5 PT	**26.4**	40.0%	**22.0**	**3.7**	**31.5**	48.1	**69.3**	48.0%	61.5	**65.0**	**76.7**	71.7
LTX-2.5 Dev	27.1	36.0%	25.3	9.5	34.9	NA	61.5	76.0%	**69.1**	37.5	55.7	**75.8**
LTX-2.5 Distilled	30.2	60.0%	30.2	17.1	39.9	**27.1**	57.4	88.0%	66.7	30.0	59.4	57.5

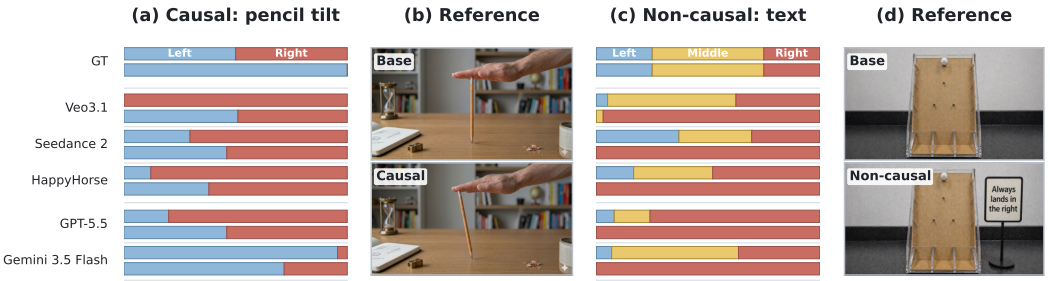


Figure 3: **Models underreact to physically causal interventions and overreact to non-causal cues.** Upper and lower bars show outcome distributions before and after intervention; panels (b) and (d) show the paired scenes. The pencil tilt is causal because it changes the physical transition, whereas the Galton-board text is non-causal because it leaves the transition unchanged.

Prompt engineering and post-training affect probabilistic alignment. Although prompt engineering (PE) and post-training are widely used to enhance video quality and controllability, higher perceptual quality does not guarantee accurate probability distributions over physical outcomes. We investigate both factors as potential contributors to this shift. For PE, Tab. 4 and Tab. 13 demonstrate that automated prompt enhancers systematically worsen probabilistic alignment, with effects that vary across enhancers and base generators (Sec. 5.1). For post-training, while constrained by the limited availability of multi-stage open checkpoints, Tab. 2[1] offers a preliminary observation that sequential post-training adaptations may affect probabilistic alignment, warranting further study.

Models Do Not Consistently Distinguish Causal from Non-Causal Changes. A probabilistic world model should change its future distribution only when the physical transition changes. We test this criterion with paired interventions: physically causal interventions alter the transition and its reference distribution, whereas non-causal interventions change only an irrelevant visual or textual signal and should therefore leave both the physical process and its future distribution unchanged.

As shown in Fig. 3, model distributions shift incompletely or in the wrong direction under physically causal interventions. Under non-causal interventions, distractor text redirects probability mass despite an unchanged reference distribution. Appendix D.6 reports results across all paired controls for the twelve video generators and direct VLM future samplers. Models respond to altered inputs, but their responses do not reliably track whether the underlying physical transition has changed.

Evaluation with Real World Inputs. To test generalization beyond synthetic source images, we further evaluate four video generators using the initial frames of three real world videos depicting coin spinning, die tossing, and bottle flipping. As shown in Fig. 4, models exhibit substantial differences in both outcome frequencies and overall reliability. These findings clearly demonstrate that even with real video inputs, models frequently collapse onto repetitive futures or entirely fail to generate readable physical endpoints. Appendix G.2 provides additional visual details and generated examples.

[1]Official LTX-2.5 model collection: `https://huggingface.co/collections/Lightricks/ltx-25`.

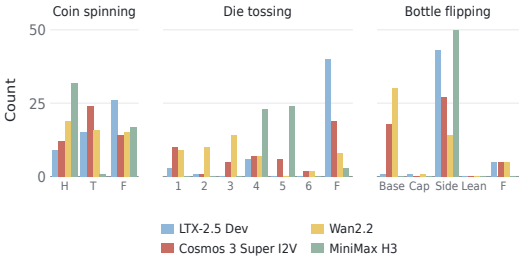


Figure 4: **Results with real-world inputs.** Outcome counts from 50 rollouts for each of four generators on coin spinning, die tossing, and bottle flipping show persistent misalignment with real-world inputs. F denotes failed outcome readout.

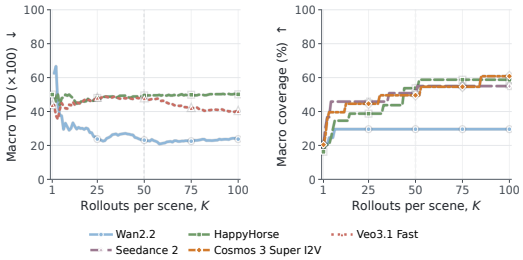


Figure 5: **Larger budgets increase coverage but leave calibration largely unchanged.** Conditional TVD (left) and valid-support coverage (right) as the budget increases from $K = 1$ to 100 rollouts per scene under fixed model inputs.

Table 3: **VLM distributions over possible futures.** Repeated VLM responses are mapped to PAWBench outcomes and aggregated into empirical distributions without video generation.

Model	PAW-Calibration: TVD ×100 (↓)						PAW-Coverage: Coverage (%) (↑)					
	Avg.	SPR	Toss	Rot.	Rout.	Draw	Avg.	SPR	Coll.	Stab.	Agent	Mat.
Qwen3.5 Plus	40.9	84.0%	36.4	**15.1**	44.9	59.0	35.9	96.0%	19.8	49.5	48.6	29.2
GPT-5.5	42.3	100.0%	35.6	42.9	41.9	56.0	34.3	100.0%	24.9	29.6	44.7	39.2
GLM-5V Turbo	**34.8**	88.0%	**24.1**	36.9	**33.9**	**50.8**	39.9	100.0%	30.9	**56.6**	42.3	38.3
Kimi K2.6	38.2	96.0%	32.6	32.1	36.8	58.5	45.1	100.0%	44.1	55.7	46.5	34.2
Gemini 3.5 Flash	39.4	100.0%	37.0	29.0	40.6	52.0	**46.6**	96.0%	**49.5**	39.5	**48.9**	**43.3**

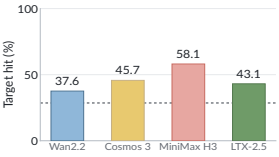


Figure 6: **Oracle PE.** Fraction of rollouts matching the target.

4.3 Robustness of the Evaluation Protocol

The main results show that current models do not reproduce the relative frequencies of valid futures and recover only part of the valid outcome support. We test whether either finding reflects a limitation of the evaluation protocol: too few rollouts or disagreement between PAWEval and human judgments.

Rollout-Budget Sensitivity. Figure 5 shows that doubling the rollout budget leaves calibration largely unchanged, although coverage continues to rise for three of the four models. Additional samples can uncover more valid outcomes without correcting their observed relative frequencies. We therefore use $K = 50$ as a shared evaluation budget without treating it as a convergence point. The Monte Carlo analysis in Appendix D.5 tests whether finite sampling could still account for the calibration gap. Across all twelve video generators, observed TVD averages 30.2. For comparison, we draw matched samples from the reference distributions while preserving each model's passing scenes and readable sample counts. Their mean TVD is 8.26; in 99% of these simulations, average TVD remains below 9.16, far below the calibration error observed across the evaluated video generators.

Human Validation of Outcome Evaluation. Our main evaluation uses the closed-source Gemini 3.5 Flash. To validate outcome readout beyond a single evaluator, we also evaluate videos with the open-weight Qwen3.8-27B (Qwen Team, 2026b). On 1,459 videos with a clear human majority and valid outcome labels from both evaluators, Gemini and Qwen agree with human judgments on 81.8% and 79.8%, respectively. These results support the reliability of PAWEval's automatic outcome readout. Appendix E provides the annotation protocol and detailed results.

5 Toward Probabilistically Aligned World Modeling

5.1 Prompt Engineering

Language can steer a video generator by explicitly naming a future in its prompt. If a model can already render several plausible futures, varying these requests across rollouts could reshape its overall output distribution. We therefore evaluate two core capabilities essential to this approach: (1) whether the language controller selects futures with the correct frequencies or support, and (2) whether the video generator successfully realizes each selected future.

Table 4: **Results of prompt engineering.** We evaluate GPT-5.5's predicted outcomes and compare Base, PE, and Oracle PE for two video generators. PE increases conditional TVD, whereas Oracle PE lowers TVD and improves Coverage for both generators.

Model	PAW-Calibration: TVD ×100 (↓)						PAW-Coverage: Coverage (%) (↑)					
	Avg.	SPR	Toss	Rot.	Rout.	Draw	Avg.	SPR	Coll.	Stab.	Agent	Mat.
GPT-5.5 PE	44.3	100.0%	40.1	49.5	43.6	49.0	35.0	100.0%	35.4	34.6	35.5	33.3
Wan2.2	26.3	64.0%	22.4	15.8	23.2	45.9	63.4	92.0%	**76.8**	56.7	55.0	58.3
+PE	27.1	92.0%	23.5	15.4	28.8	39.5	61.9	100.0%	68.1	50.7	62.9	57.5
+Oracle PE	**15.3**	84.0%	**10.5**	**13.9**	**14.0**	**27.3**	**76.2**	96.0%	71.1	**72.7**	**82.8**	**76.7**
Cosmos 3 Super I2V	20.5	80.0%	19.4	**3.5**	17.7	35.5	55.2	92.0%	56.7	31.7	49.4	**81.7**
+PE	31.6	92.0%	27.0	15.0	34.9	47.0	64.9	100.0%	71.1	62.0	53.5	76.7
+Oracle PE	**12.8**	88.0%	**17.6**	9.2	**13.2**	**5.1**	**87.3**	100.0%	**84.8**	**91.4**	**91.3**	80.8

We test three settings. First, prior work finds that language models often understand or describe a target distribution more accurately than they reproduce it through repeated sampling, although distribution-aware prompting and training can narrow this gap (Meister et al., 2024; Gu et al., 2024; Misaki & Akiba, 2026; Sorensen et al., 2026). We therefore ask whether VLMs can act as samplers over possible futures without being given the target probabilities. We repeatedly query five VLMs (Qwen Team, 2026a; OpenAI, 2026a; Z.AI, 2026; Moonshot AI, 2026; Google DeepMind, 2026) under the same initial observation and action, map each prediction to the PAWBench outcome space, and aggregate the predictions into an empirical distribution. Second, GPT-5.5 performs prompt engineering (PE) for each rollout: without access to the reference distribution, it predicts a possible outcome and writes a generator prompt requesting that outcome. Third, for Oracle PE, we directly specify one target outcome in each generator prompt, arranging the targets across rollouts to follow the reference distribution in PAW-Calibration and to balance the valid outcomes in PAW-Coverage. Together, these settings separate two sources of error: predicting the wrong distribution of outcomes and failing to produce a requested outcome in video. Direct VLM future sampling tests the first, PE evaluates the full chain, and Oracle PE supplies the target outcomes directly to isolate the second.

The distributions produced by direct VLM sampling are inherently misaligned: Tab. 3 shows that GLM-5V Turbo achieves the lowest Calibration TVD at 34.8, while Gemini 3.5 Flash achieves the highest Coverage at only 46.6%. Because the controller itself selects the wrong distribution of futures, using VLM predictions to prompt video generators (PE) fails to align the final outputs. As shown in Tab. 4 and Tab. 14, while PE raises SPR across all four tested models, it inherits the controller's bias, strictly increasing Calibration TVD among passing scenes compared to the Base setting. Oracle PE successfully lowers Calibration TVD and raises Coverage for every generator. Even if we isolate and eliminate this controller error via manually specified targets, the visual generators themselves often miss the supplied target. Fig. 6 shows that generators realize only 37.6–58.1% of the perfectly requested outcomes. Therefore, current language-based control is severely limited by both errors: it can change individual rollouts, but cannot reliably align their overall distribution.

5.2 Initial Noise Sampling

While language interventions force specific outcomes, they obscure the model's intrinsic physical diversity under a fixed prompt. We therefore shift our focus from language to initial noise sampling. Since standard random seeds often get trapped in frequent modes, we keep the prompt fixed and apply a targeted sampling strategy to force broader exploration of valid outcomes.

To test whether enhancing initial-noise diversity can improve output coverage beyond standard random seeds, we adopt Couple to Control (C2C) (Jia et al., 2026), a repulsive Gaussian coupling scheme that introduces negative dependence among the $K = 50$ initial-noise samples while preserving each sample's standard Gaussian marginal. The action prompt, generator, and rollout budget remain fixed across C2C and independent sampling; Appendix F.2 details the coupled-noise construction and the matched evaluation conditions used to isolate the effect of noise coupling.

Tab. 5 reports results across all 25 scenes in each PAWBench track. Across the scenes that pass in each condition, C2C lowers mean Calibration TVD and raises mean Coverage for all three generators. The gains vary across mechanisms and do not consistently raise SPR. C2C therefore helps the 50

Table 5: **Coupled noise broadens finite-gallery exploration.** C2C (Jia et al., 2026) couples $K = 50$ rollout noises while preserving the standard Gaussian marginal of each noise.

Model	PAW-Calibration: TVD ×100 (↓)						PAW-Coverage: Coverage (%) (↑)					
	Avg.	SPR	Toss	Rot.	Rout.	Draw	Avg.	SPR	Coll.	Stab.	Agent	Mat.
Wan2.2	26.3	64.0%	22.4	15.8	23.2	45.9	63.4	92.0%	76.8	56.7	55.0	58.3
+C2C	25.7	84.0%	**17.4**	22.3	21.6	48.7	69.2	88.0%	82.4	57.5	59.4	68.3
LTX-2.3	30.1	24.0%	29.3	11.0	NA	41.0	71.7	72.0%	82.7	50.0	**59.5**	**90.0**
+C2C	19.9	32.0%	18.6	7.2	34.0	27.7	**74.8**	72.0%	**92.8**	**87.5**	57.1	76.7
Cosmos 3 Super I2V	20.5	80.0%	19.4	**3.5**	17.7	35.5	55.2	92.0%	56.7	31.7	49.4	81.7
+C2C	**19.4**	72.0%	25.8	6.1	**14.3**	**24.2**	63.9	92.0%	74.9	56.7	49.1	76.7

rollouts explore the model's existing possibilities more broadly, rather than changing the distribution learned by the model. We therefore turn next to the model itself.

5.3 Updating model parameters

Unlike language and noise interventions, changing the training distribution can alter the generator's learned allocation of probability mass. We train LoRA (Hu et al., 2021) adaptations of Wan2.2, MiniMax H3, and LTX-2.5 on three binary-outcome scenarios, varying the proportion of one outcome in the training data from 0% to 100%. Within each experimental series, we hold the training budget and recipe fixed and evaluate each adaptation on two physical states using a neutral action prompt and $K = 50$ rollouts. Because the interior mixtures contain both outcomes, differences among them probe relative mass rather than support. Appendix F.3 provides the details.

Fig. 7 shows that increasing an outcome's training proportion generally raises its generated frequency across all three scenarios and all three models, with a nonlinear response. Even balanced training does not guarantee balanced outputs. This control does not reliably transfer to related physical states with different reference distributions. Training on outcome frequencies therefore controls generation probabilities, but does not reliably adapt them to the physical distribution of related scenes in these experiments.

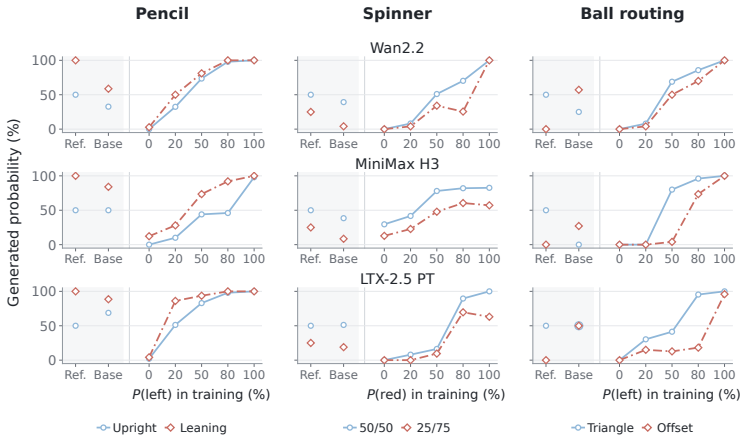


Figure 7: **Training control is nonlinear and state-dependent.** Outcome frequencies respond to training proportions but do not reliably match physical references across related states.

6 Conclusion

World modeling requires capturing the true distribution of possible futures given an initial observation and action, rather than merely producing plausible continuations. We introduce PAWBench to measure this capability via repeated video rollouts. Across twelve video generators, no model consistently aligns with reference probabilities while covering the range of valid futures, a systematic shortfall that persists beyond finite sampling limits and is robustly measured by metrics aligned with human evaluation. Controlled interventions reveal that model distributions fail to reliably track physical causal changes; while language prompting, coupled noise, and fine-tuning can alter outcome frequencies, none restores the true scene-conditioned distribution. PAWBench demonstrates that plausible, diverse, or controllable rollouts alone do not constitute probabilistically faithful world modeling, highlighting the need for probabilistic alignment.

AI USE STATEMENT

We used generative AI tools to generate synthetic source images for PAWBench and synthetic videos for controlled fine-tuning, as described in the data-construction sections. We also used large language models to edit the manuscript for clarity and readability. The authors reviewed the generated data and AI-assisted revisions and take responsibility for all of the paper's final content, claims, and artifacts.

ETHICS STATEMENT

PAWBench evaluates controlled physical scenarios. We used Rapidata[2] to collect human judgments of visible terminal outcomes (Appendix E). Judgments concern model outputs, not annotators' characteristics. Benchmark scores do not certify the safety or reliability of real-world deployment.

REPRODUCIBILITY STATEMENT

Scenario construction and outcome definitions appear in Appendix C; the evaluation protocol, readout criteria, and metric aggregation appear in Appendix D. Appendix E documents human evaluation, and Appendix F details the prompting, noise-sampling, and fine-tuning settings used in our experiments.

REFERENCES

Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, et al. Cosmos World Foundation Model Platform for Physical AI. *arXiv preprint arXiv:2501.03575*, 2025.

Alibaba Cloud. Wan2.7, 2026. URL https://www.alibabacloud.com/help/en/model-studio/video-generate-edit-model. Model Studio documentation.

Alibaba Cloud Community. Alibaba rolls out HappyHorse 1.0 in limited beta, 2026. URL https://www.alibabacloud.com/blog/alibaba-rolls-out-happyhorse-1-0-in-limited-beta_603068.

Mohammad Babaeizadeh, Chelsea Finn, Dumitru Erhan, Roy H Campbell, and Sergey Levine. Stochastic variational video prediction. *arXiv preprint arXiv:1710.11252*, 2018.

Philip J. Ball, Jakob Bauer, Frank Belletti, Bethanie Brownfield, Ariel Ephrat, Shlomi Fruchter, Agrim Gupta, Kristian Holsheimer, Aleksander Holynski, Jiri Hron, Christos Kaplanis, Marjorie Limont, Matt McGill, Yanko Oliveira, Jack Parker-Holder, Frank Perbet, Guy Scully, Jeremy Shar, Stephen Spencer, Omer Tov, Ruben Villegas, Emma Wang, Jessica Yung, Cip Baetu, Jordi Berbel, David Bridson, Jake Bruce, Gavin Buttimore, Sarah Chakera, Bilva Chandra, Paul Collins, Alex Cullum, Bogdan Damoc, Vibha Dasagi, Maxime Gazeau, Charles Gbadamosi, Woohyun Han, Ed Hirst, Ashyana Kachra, Lucie Kerley, Kristian Kjems, Eva Knoepfel, Vika Koriakin, Jessica Lo, Cong Lu, Zeb Mehring, Alex Moufarek, Henna Nandwani, Valeria Oliveira, Fabio Pardo, Jane Park, Andrew Pierson, Ben Poole, Helen Ran, Tim Salimans, Manuel Sanchez, Igor Saprykin, Amy Shen, Sailesh Sidhwani, Duncan Smith, Joe Stanton, Hamish Tomlinson, Dimple Vijaykumar, Luyu Wang, Piers Wingfield, Nat Wong, Keyang Xu, Christopher Yew, Nick Young, Vadim Zubov, Douglas Eck, Dumitru Erhan, Koray Kavukcuoglu, Demis Hassabis, Zoubin Gharamani, Raia Hadsell, Aäron van den Oord, Inbar Mosseri, Adrian Bolton, Satinder Singh, and Tim Rocktäschel. Genie 3: A new frontier for world models. *Google DeepMind Blog*, 2025. URL https://deepmind.google/blog/genie-3-a-new-frontier-for-world-models/.

Hritik Bansal, Zongyu Lin, Tianyi Xie, Zeshun Zong, Michal Yarom, Yonatan Bitton, Chenfanfu Jiang, Yizhou Sun, Kai-Wei Chang, and Aditya Grover. VideoPhy: Evaluating physical commonsense for video generation. In *International Conference on Learning Representations*, volume 2025, pp. 102075–102121, 2025.

Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video

[2] https://www.rapidata.ai/

generation models as world simulators. *OpenAI Blog*, 2024. URL https://openai.com/research/video-generation-models-as-world-simulators.

Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. In *Forty-first international conference on machine learning*, 2024.

ByteDance Seed. Seedance 2.0, 2026a. URL https://seed.bytedance.com/en/seedance2_0.

ByteDance Seed. One-take creation, flexible referencing: Introducing Seedance 2.5, 2026b. URL https://seed.bytedance.com/en/blog/one-take-creation-flexible-referencing-introducing-seedance-2-5.

Emily Denton and Rob Fergus. Stochastic video generation with a learned prior. In *International conference on machine learning*, pp. 1174–1183. PMLR, 2018.

Zelin Gao, Qiuyu Wang, Jiapeng Zhu, Jingye Chen, Zichen Liu, Qingyan Bai, Jiahao Wang, Yufeng Yuan, Hanlin Wang, Yichong Lu, Ka Leong Cheng, Haojie Zhang, Jian Gao, Tianrui Feng, Yuzheng Liu, Yao Yao, Yinghao Xu, Xing Zhu, Yujun Shen, and Hao Ouyang. Infinite worlds with versatile interactions. *arXiv preprint arXiv:2607.07534*, 2026. URL https://arxiv.org/abs/2607.07534.

Alison L. Gibbs and Francis Edward Su. On choosing and bounding probability metrics. *International Statistical Review*, 70(3):419–435, 2002. doi: 10.1111/j.1751-5823.2002.tb00178.x.

Google. Veo 3.1 Fast, 2025. URL https://ai.google.dev/gemini-api/docs/veo. Gemini API documentation.

Google DeepMind. Nano Banana Pro, 2025. URL https://deepmind.google/models/gemini-image/pro/.

Google DeepMind. Gemini 3.5 Flash, 2026. URL https://deepmind.google/models/model-cards/gemini-3-5-flash/.

Jia Gu, Liang Pang, Huawei Shen, and Xueqi Cheng. Do LLMs play dice? exploring probability distribution sampling in large language models for behavioral simulation, 2024. URL https://arxiv.org/abs/2404.09043.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pp. 1321–1330. PMLR, 2017. URL https://arxiv.org/abs/1706.04599.

Xuyang Guo, Jiayan Huo, Zhenmei Shi, Zhao Song, Jiahao Zhang, and Jiale Zhao. T2VPhysBench: A first-principles benchmark for physical consistency in text-to-video generation. *arXiv preprint arXiv:2505.00337*, 2025.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In *Advances in Neural Information Processing Systems 31*, pp. 2451–2463. Curran Associates, Inc., 2018. URL https://papers.nips.cc/paper/7512-recurrent-world-models-facilitate-policy-evolution. https://worldmodels.github.io.

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

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*, 2020.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. *CoRR*, abs/2106.09685, 2021. URL https://arxiv.org/abs/2106.09685.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. VBench: Comprehensive benchmark suite for video generative models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21807–21818. IEEE, 2024.

Ziqi Huang, Fan Zhang, Xiaojie Xu, Yinan He, Jiashuo Yu, Ziyue Dong, Qianli Ma, Nattapol Chanpaisit, Chenyang Si, Yuming Jiang, et al. VBench++: Comprehensive and versatile benchmark suite for video generative models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

Jing Jia, Liyue Shen, and Guanyang Wang. Couple to control: Joint initial noise design in diffusion models, 2026. URL https://arxiv.org/abs/2605.11311.

Bingyi Kang, Yang Yue, Rui Lu, Zhijie Lin, Yang Zhao, Kaixin Wang, Gao Huang, and Jiashi Feng. How far is video generation from world model: A physical law perspective, 2025. URL https://arxiv.org/abs/2411.02385.

Kling AI. Kling Video 3.0 Model User Guide, 2026. URL https://kling.ai/quickstart/klingai-video-3-model-user-guide.

Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models, 2019. URL https://arxiv.org/abs/1904.06991.

Dacheng Li, Yunhao Fang, Yukang Chen, Shuo Yang, Shiyi Cao, Justin Wong, Michael Luo, Xiaolong Wang, Hongxu Yin, Joseph Gonzalez, Ion Stoica, Song Han, and Yao Lu. WorldModelBench: Judging video generation models as world models. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference. Curran Associates, Inc., 2025. doi: 10.52202/085713-1834. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/4ec03ed08a3fcb59e1c815b5598beff1-Paper-Datasets_and_Benchmarks_Track.pdf.

Shiyang Li, Yuewen Cao, Yihao Liu, Yuandong Pu, Baochang Zhang, Xiaofei Li, and Changqing Zou. AcoustiTrace: When plausible sound violates physics. *arXiv preprint arXiv:2608.02035*, 2026.

Lightricks. LTX-2.5, 2026a. URL https://ltx.io/model/ltx-2-5.

Lightricks. LTX-2.3 Video Engine, 2026b. URL https://ltx.io/model/ltx-2-3.

Yaofang Liu, Xiaodong Cun, Xuebo Liu, Xintao Wang, Yong Zhang, Haoxin Chen, Yang Liu, Tieyong Zeng, Raymond Chan, and Ying Shan. EvalCrafter: Benchmarking and evaluating large video generation models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22139–22149. IEEE, 2024.

Shuailei Ma, Jiaqi Liao, Xinyang Wang, Jingjing Wang, Chaoran Feng, Zijing Hu, Chong Bao, Zichen Xi, Yuqi Gan, Weisen Wang, et al. Scaling mixture-of-experts video pretraining for embodied intelligence. *arXiv preprint arXiv:2607.07675*, 2026.

Zhiting Mei, Ola Shorinwa, and Anirudha Majumdar. How confident are video models? empowering video models to express their uncertainty, 2025. URL https://arxiv.org/abs/2510.02571.

Zhiting Mei, Tenny Yin, Micah Baker, Ola Shorinwa, and Anirudha Majumdar. World models that know when they don't know - controllable video generation with calibrated uncertainty, 2026. URL https://arxiv.org/abs/2512.05927.

Nicole Meister, Carlos Guestrin, and Tatsunori Hashimoto. Benchmarking distributional alignment of large language models, 2024. URL https://arxiv.org/abs/2411.05403.

Fanqing Meng, Jiaqi Liao, Xinyu Tan, Wenqi Shao, Quanfeng Lu, Kaipeng Zhang, Yu Cheng, Dianqi Li, Yu Qiao, and Ping Luo. Towards world simulator: Crafting physical commonsense-based benchmark for video generation. *arXiv preprint arXiv:2410.05363*, 2024.

MiniMax. MiniMax H3: An open model breaking the boundaries between tasks and modalities, July 2026. URL https://www.minimax.io/blog/minimax-h3.

Kou Misaki and Takuya Akiba. String seed of thought: Prompting LLMs for distribution-faithful and diverse generation, 2026. URL https://arxiv.org/abs/2510.21150.

Moonshot AI. Kimi-K2.6, 2026. URL https://huggingface.co/moonshotai/Kimi-K2.6.

Muhammad Ferjad Naeem, Seong Joon Oh, Youngjung Uh, Yunjey Choi, and Jaejun Yoo. Reliable fidelity and diversity metrics for generative models. *CoRR*, abs/2002.09797, 2020. URL https://arxiv.org/abs/2002.09797.

Eric Nalisnick, Akihiro Matsukawa, Yee Whye Teh, Dilan Gorur, and Balaji Lakshminarayanan. Do deep generative models know what they don't know? In *International Conference on Learning Representations*, 2019. URL https://arxiv.org/abs/1810.09136.

NVIDIA. Cosmos 3: Omnimodal world models for physical AI. *arXiv preprint arXiv:2606.02800*, 2026. URL https://research.nvidia.com/labs/cosmos-lab/cosmos3/technical-report.pdf.

OpenAI. GPT-5.5 model, 2026a. URL https://developers.openai.com/api/docs/models/gpt-5.5.

OpenAI. GPT Image 2, April 2026b. URL https://developers.openai.com/api/docs/models/gpt-image-2.

Yuandong Pu, Le Zhuo, Songhao Han, Jinbo Xing, Kaiwen Zhu, Shuo Cao, Bin Fu, Si Liu, Hongsheng Li, Yu Qiao, Wenlong Zhang, Xi Chen, and Yihao Liu. PICABench: How far are we from physical realistic image editing? In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 106564–106593, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/ae0509bfd3e0c2414164209949708e4e-Paper-Conference.pdf.

Yiran Qin, Zhelun Shi, Jiwen Yu, Xijun Wang, Enshen Zhou, Lijun Li, Zhenfei Yin, Xihui Liu, Lu Sheng, Jing Shao, et al. WorldSimBench: Towards video generation models as world simulators. *arXiv preprint arXiv:2410.18072*, 2024.

Qwen Team. Qwen3.5: Towards native multimodal agents, 2026a. URL https://qwen.ai/blog?id=qwen3.5.

Qwen Team. Qwen3.8-Max: A new bar for coding and cowork, August 2026b. URL https://qwen.ai/blog?id=qwen3.8.

Sahithya Ravi, Aditya Chinchure, Pushkar Shukla, Vered Shwartz, and Leonid Sigal. Position: World models must live in parallel worlds. In *NeurIPS 2025 Workshop on Bridging Language, Agent, and World Models for Reasoning and Planning*, 2025. URL https://openreview.net/forum?id=sj9Nyke43S.

Jonathan Sadeghi, Jenny Seidenschwarz, Jesse Allardice, Sirish Srinivasan, Benjamin Graham, and Jeffrey Hawke. CaliBench: Are the stochastic dynamics of video world models physically calibrated? *arXiv preprint arXiv:2608.16829*, 2026.

Mehdi S. M. Sajjadi, Olivier Bachem, Mario Lucic, Olivier Bousquet, and Sylvain Gelly. Assessing generative models via precision and recall, 2018. URL https://arxiv.org/abs/1806.00035.

Enes Sanli, Baris Sarper Tezcan, Aykut Erdem, and Erkut Erdem. Can your model separate yolks with a water bottle? benchmarking physical commonsense understanding in video generation models. *arXiv preprint arXiv:2507.15824*, 2026.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, et al. Mastering Atari, Go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, 2020.

Taylor Sorensen, Benjamin Newman, Jared Moore, Chan Park, Jillian Fisher, Niloofar Mireshghallah, Liwei Jiang, and Yejin Choi. Spectrum tuning: Post-training for distributional coverage and in-context steerability, 2026. URL https://arxiv.org/abs/2510.06084.

Richard S Sutton. Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. In *Machine learning proceedings 1990*, pp. 216–224. Elsevier, 1990.

Stephen Tian, Chelsea Finn, and Jiajun Wu. A control-centric benchmark for video prediction, 2023. URL https://arxiv.org/abs/2304.13723.

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. *arXiv preprint arXiv:2503.20314*, 2025.

World Labs Team. A functional taxonomy of world models: Renderers, simulators, planners, and the loop that connects them, June 2026. URL https://www.worldlabs.ai/blog/taxonomy-of-world-models.

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.

Z.AI. GLM-5V-Turbo overview, 2026. URL https://docs.z.ai/guides/vlm/glm-5v-turbo.

A Related Work

Video Generators as World Models. World models have long been used to support planning by predicting the consequences of actions, from classical model-based reinforcement learning to latent dynamics and decision-centric models (Sutton, 1990; Ha & Schmidhuber, 2018; Hafner et al., 2020; Schrittwieser et al., 2020). Recent video systems increasingly pursue this role through controllable visual rollouts, including large-scale video generators and interactive world models (Brooks et al., 2024; Bruce et al., 2024; Ball et al., 2025; Agarwal et al., 2025; Yang et al., 2024). This shift makes repeated rollouts from the same initial observation and action natural. PAWBench focuses on the distributional question left open by these systems: not only whether a rollout is coherent, controllable, or visually realistic, but whether the induced samples place probability mass on the right futures.

Video-generation and world-model benchmarks. Video-generation benchmarks have substantially improved evaluation of visual quality, temporal consistency, and text-video alignment (Liu et al., 2024; Huang et al., 2024; 2025). Physics-focused benchmarks further probe physical plausibility and action following (Bansal et al., 2025; Meng et al., 2024; Guo et al., 2025; Sanli et al., 2026; Pu et al., 2026; Li et al., 2026). Recent world-model benchmarks evaluate action-conditioned prediction, physical-law adherence, embodied consistency, or downstream utility for planning and control (Li et al., 2025; Qin et al., 2024; Kang et al., 2025; Tian et al., 2023). These evaluations are complementary to PAWBench, but their unit of analysis is usually an individual generation, a deterministic violation, or task success under a prompt. PAWBench instead holds the initial observation and action fixed, samples repeated rollouts, maps them to terminal outcome labels, and evaluates whether the empirical outcome distribution or support matches the stochastic structure of the scene.

Calibration and coverage under fixed actions. Calibration and distributional evaluation separate accuracy, fidelity, probability assignment, and support recovery (Guo et al., 2017; Nalisnick et al., 2019; Sajjadi et al., 2018; Kynkäänniemi et al., 2019; Naeem et al., 2020). Recent work on generative-video uncertainty also shows that confidence estimation is feasible and important (Mei et al., 2025; 2026). Closest to our setting, CaliBench evaluates nine stochastic scenes with known reference distributions, reporting scorability separately from conditional TVD and testing for miscalibration (Sadeghi et al., 2026). PAWBench shares the repeated-rollout, discrete-outcome perspective but separates two regimes: PAW-Calibration when defensible reference probabilities are available, and PAW-Coverage when only the valid support can be specified. It pairs these measurements with controlled probes of causal state, language, sampling noise, and training distribution.

B Limitations

PAWBench is designed as a diagnostic test of probabilistic alignment, and its current form has three main limitations. First, it evaluates stochastic futures through terminal outcomes, which makes distributional comparison tractable but does not fully capture trajectory-level dynamics or intermediate physical processes. Second, its estimates are based on a finite number of rollouts: larger sampling budgets can reveal the induced distribution more reliably, but they increase evaluation cost and do not by themselves correct biased model distributions. Third, PAWBench uses controlled, visually parseable scenarios to isolate stochastic future modeling, leaving longer-horizon, interactive, and embodied environments for future study. Future work should extend probabilistic alignment from terminal labels to richer state trajectories, study more efficient and reliable rollout-based estimators, scale the benchmark to interactive settings, and develop training objectives for models that explicitly learn calibrated distributions over possible futures across different physical states.

C Benchmark Details

C.1 Scenario Definition and Task Taxonomy

Each PAWBench scenario fixes the inputs and outcomes of one repeated-rollout test, as shown in Tab. 6. A source image represents the initial observation, an action prompt specifies one atomic intervention, and a finite outcome set $\mathcal{Y}$ lists the valid terminal outcomes of the physical process. Each scenario also includes outcome-readout criteria and, when it can be derived independently of

Table 6: **PAWBench task families and outcome readouts.** The eight families cover PAW-Calibration and PAW-Coverage. Audit examples include outcome-readout failures and auxiliary trustworthiness concerns. Only outcome readout determines the labels and scene gate used for scoring; trustworthiness checks remain diagnostic (Appendix D.1).

Task family	Task definition	Terminal readout	Audit examples
PAW-Calibration			
Tossing	An object is thrown, flicked, or released into one of several settled states.	Face, orientation, or resting region.	No throw or release; continuity break; settled state unreadable.
Rotation	An object spins relative to a fixed partition, pointer, or orientation frame.	Final sector or orientation.	No rotation; reference frame or endpoint unreadable; out-of-schema orientation.
Routing	A released object traverses a visible branching structure.	Terminal branch, lane, bin, pocket, or location.	No traversal; continuity break; terminal location unreadable or out of schema.
Draw	One item is selected or released from a visible finite collection.	Selected identity or category.	No selection; item continuity break; selected identity unreadable or out of schema.
PAW-Coverage			
Collision	An object undergoes contact or a target-directed interaction.	Contact, deflection, scoring, miss, or settling state.	Intervention absent; impossible transition; continuity or readability failure.
Stability	A near-threshold arrangement is subjected to a perturbation.	Stable, shifted, toppled, or collapsed state.	No perturbation; continuity break; terminal state unreadable or out of schema.
Agent interaction	A living agent receives a fixed visible stimulus.	Observable response category.	Stimulus not delivered as specified; agent continuity break; response unreadable.
Material transition	A material or deformable object undergoes a fixed intervention.	Resulting physical or material state.	Intervention absent; material continuity break; final state unreadable or out of schema.

model outputs, a reference distribution q over $\mathcal{Y}$. We finalize these elements before evaluating any model.

C.1.1 PAW-Calibration Tasks

PAW-Calibration comprises scenarios whose reference distribution q is fixed before model evaluation. We derive q from sector proportions, combinatorial outcome counts, physical symmetry, or visible causal state, not from model outputs or a default uniform assumption. When the initial observation and action do not support a defensible reference distribution, the scenario is assigned to PAW-Coverage or discarded. PAW-Calibration tests whether repeated rollouts allocate probability mass in accordance with q.

C.1.2 PAW-Coverage Tasks

PAW-Coverage comprises scenarios whose valid terminal outcomes can be enumerated but whose relative probabilities cannot be justified from the initial observation and action. It tests support recovery across repeated rollouts without assuming equal outcome probabilities.

C.2 Scenario Construction and Quality Review

We construct each scenario so that variation across repeated rollouts reflects the model's future distribution rather than ambiguity in the benchmark item. Before evaluation, we finalize the source

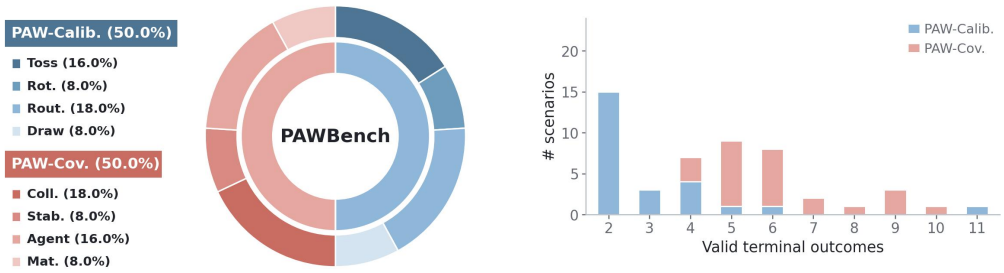


Figure 8: **Benchmark statistics of PAWBench.** PAWBench contains 50 manually curated scenarios, evenly divided between PAW-Calibration and PAW-Coverage, across eight stochastic mechanism groups. Each scenario fixes one source image and action prompt and uses a compact terminal-outcome set, allowing repeated rollouts to be aggregated into an empirical outcome distribution.

image, action prompt, valid terminal outcomes, reference distribution when applicable, and outcome-readout criteria. The physical mechanism must be visible, the action must specify one intervention, and the terminal outcomes must be distinguishable.

Source-image selection. For each scenario, we generate candidate source images with image generation models (Google DeepMind, 2025; OpenAI, 2026b) and manually select one image. The selected image must clearly show the relevant physical mechanism and action target, avoid hidden initial conditions, and support terminal outcomes that can be distinguished from the final frames. We reject candidates with an obscured mechanism, an ambiguous initial state, or competing action targets.

Scenario finalization. Given the selected source image, we write an action prompt that specifies one atomic intervention whose completion can be judged from the generated video. We then define the valid terminal outcomes, criteria for readable in-schema labels, and separate trustworthiness checks. For PAW-Calibration, we retain a reference distribution only when analytic reasoning or physical symmetry justifies it independently of model outputs. Outcome categories describe terminal physical states rather than incidental differences in appearance or intermediate trajectories.

Quality review. We review each candidate as a complete scenario before evaluating any model. The review checks mechanism visibility, action specificity, terminal-outcome distinguishability, reference-distribution justification when applicable, and the outcome-readout criteria. Failed checks lead us to select or generate a new source image, revise the action prompt, or refine the outcome categories. We discard a scenario when its ambiguity cannot be removed without changing the physical process being evaluated.

D Evaluation Protocol and Supporting Analyses

D.1 PAWEval Outcome Readout

PAWEval uses Gemini 3.5 Flash (Google DeepMind, 2026) to apply a frozen scene-specific outcome rubric to each generated rollout. The rubric returns a terminal label $y \in \mathcal{Y}$ when the endpoint is readable and belongs to the outcome set; otherwise, it returns the shared readout-failure label $\perp$. When a visible endpoint is more specific than the outcome categories, the rubric maps it to the corresponding category before aggregation. These outcome labels determine conditional TVD and Coverage. Readout failures instead determine whether a scene passes the outcome-readout gate and therefore affect SPR; they never enter the conditional outcome distribution.

We also run an auxiliary trustworthiness audit that records whether the requested action is completed, object continuity is preserved, and the transition follows the specified physical mechanism. This separate diagnostic does not change the terminal label or any PAWBench score. Tab. 7 summarizes the scoring readout and the auxiliary audit.

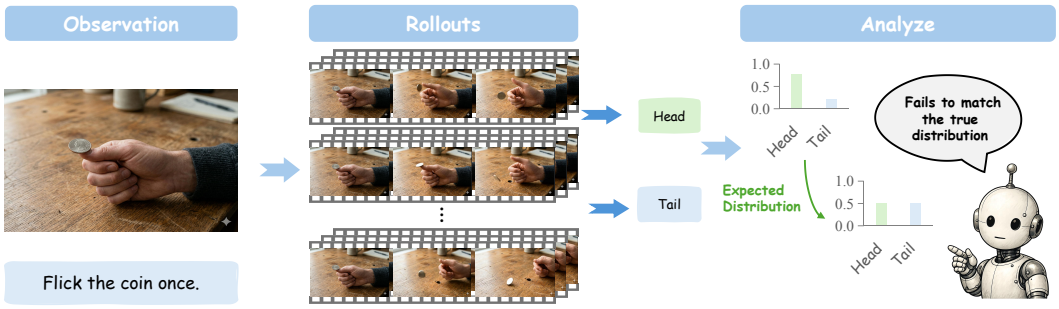


Figure 9: **PAWEval turns repeated rollouts into a distributional test.** In this PAW-Calibration example, a rubric-based judge maps each readable, in-schema coin-toss rollout to Head or Tail. Aggregating these labels yields the empirical outcome distribution, which is compared with the reference probabilities rather than matching generated videos frame by frame.

For example, in a routing scene whose schema contains only named terminal bins, a ball that comes to rest visibly outside those bins has a readable but out-of-schema endpoint and is therefore recorded as an outcome-readout failure. By contrast, a spinner that ends clearly on blue after an implausible or discontinuous rotation still receives the in-schema label blue, while the trajectory is flagged only by the separate trustworthiness audit and the blue label still enters the conditional outcome counts.

Table 7: **PAWEval separates scoring from auxiliary diagnostics.** Outcome readout supplies the labels and scene gate used by PAWBench; trustworthiness criteria only diagnose how readable outcomes are realized.

Criterion	Required condition	Failure condition
Outcome readout		
Endpoint readability	The terminal state can be determined visually.	The endpoint is hidden, ambiguous, or unreadable.
Schema membership	The terminal state maps to the fixed outcome set $\mathcal{Y}$.	A readable endpoint falls outside $\mathcal{Y}$.
Trustworthiness audit		
Action completion	The requested intervention is executed as specified.	The action is absent or the clip is off task.
Object continuity	Key objects remain identifiable throughout the rollout.	Objects disappear, duplicate, or change identity.
Physical process	The transition follows the specified physical mechanism.	The transition violates the specified physical mechanism.

Fig. 10 shows the fixed PAWEval prompt scaffold into which the scene-specific outcome and trustworthiness rubrics are inserted.

D.2 Auxiliary Trustworthiness Diagnostics

The trustworthiness audit asks how a readable endpoint was reached. A clip may reach such an endpoint without completing the intended physical process: the action may not be executed, object identity may break, or the endpoint may be reached through physically implausible dynamics. These flags do not change the outcome label, TVD, Coverage, or SPR.

The audit covers three types of failure: action execution, physical process, and object continuity. Fig. 11 shows that physical-process errors account for the largest share overall, while action-execution and object-continuity errors remain substantial across mechanism groups.

Prompt for PAWEval Rubric Judging

Fixed two-axis judge scaffold

```
System: You are a strict PAWEval JSON judge. Return one valid JSON object and no prose.

Runtime inputs shared by both axes:
- Scene: {scene_id}; Sample: {sample_id}
- Evidence: {evidence_summary}
- Scene rubric: {rubric_text}
- Video evidence: sampled frames attached in initial / action / terminal order
```

Axis 1: Outcome readout

```
# PAWEval Outcome Readout

Use the provided evidence and scene rubric to return one outcome
label.
Do not judge trustworthiness in this axis.
Return a single JSON object only. Do not use Markdown fences,
explanations,
or any text before or after the JSON object. The response must begin
with {
and end with }.

Return JSON with this exact shape and preserve the provided scene_id:

{
  "scene_id": "{scene_id}",
  "outcome_readable": true,
  "outcome_in_schema": "IN_SCHEMA",
  "outcome_label": "<one rubric label or null>",
  "observed_result": "<short evidence-grounded observation>",
  "failure_axes": []
}

If the result is not readable, set outcome_readable=false,
outcome_in_schema="NOT_READABLE", outcome_label=null, and include the
reason
in failure_axes. If the result is readable but outside the rubric, set
outcome_in_schema="OUT_OF_SCHEMA" and outcome_label=null.

Runner addendum: append JSON schema paweval.outcome_readout/v1,
require one
canonical rubric label for IN_SCHEMA, and end with "Return JSON only."
```

Axis 2: Trustworthiness audit

```
# PAWEval Trustworthiness Audit

Audit whether the evidence supports strict reporting. Do not choose the
outcome
label in this axis. Return a single JSON object only.

The top-level status field must be exactly one of:
TRUSTED, QUESTIONABLE, UNTRUSTED, UNCERTAIN.

Do not use PASS or FAIL as the top-level status. PASS, FAIL, UNCLEAR, and
NOT_APPLICABLE are allowed only inside nested audit blocks such as
scene_grounding.status and action_execution.status.

Return JSON with this exact shape and preserve the provided scene_id:

{
  "scene_id": "{scene_id}",
  "status": "TRUSTED",
  "scene_grounding": {"status": "PASS", "notes": ""},
  "action_execution": {
    "status": "PASS",
    "target_hit": "YES",
    "object_acquired": "YES",
    "action_spec_followed": "YES",
    "notes": ""
  },
  "object_continuity": {"status": "PASS", "notes": ""},
  "physical_process": {"status": "PASS", "notes": ""},
  "failure_axes": []
}

Use TRUSTED only when all required evidence supports the report. Use
QUESTIONABLE for incomplete or ambiguous evidence, UNTRUSTED for
contradictory
evidence, and UNCERTAIN when the evidence cannot support a reliable audit.

For this automated runner, keep every notes field as the empty string "".
Name concrete failure axes in failure_axes rather than adding prose outside
the JSON object.

Runner addendum: append JSON schema paweval.trustworthiness_audit/v1,
require
the official trust enums, and end with "Return JSON only."
```

Figure 10: **PAWEval prompt scaffold.** The scene-specific outcome rubric assigns a terminal label or an outcome-readout failure. A separate trustworthiness rubric checks action execution, physical process, and object continuity without changing the PAWBench score. The placeholders `{evidence_summary}` and `{rubric_text}` are filled with sampled-frame evidence and the scene-specific rubric.

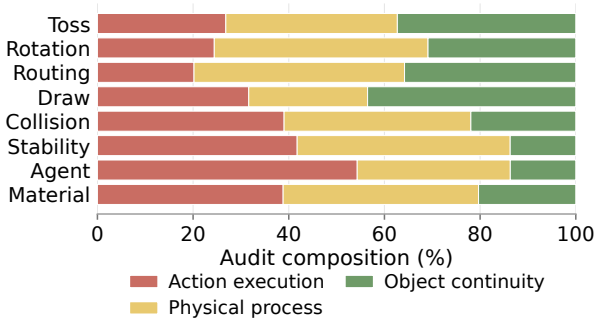


Figure 11: **Failure anatomy.** Incidence composition by mechanism group for eight models: HappyHorse, Veo 3.1 Fast, Kling 3 Std., Seedance 2, Wan2.7, Wan2.2, LTX-2.3, and Cosmos 3 Super I2V. Percentages use classifiable audits and count non-exclusive failures.

D.3 Metric Computation and Aggregation

PAWBench first applies a scene-level outcome-readout gate. For each model–scene pair, we evaluate $K = 50$ rollouts generated under the same initial observation and action. The outcome readout assigns each rollout either a terminal label $y_i \in \mathcal{Y}$ or the readout-failure label $\perp$. Let m denote

the number of rollouts assigned to $\perp$. A scene passes the gate when $m \leq 30$, equivalently when at least 20 rollouts have readable, in-schema outcomes. Otherwise, the scene does not contribute a conditional alignment score.

Conditional metrics are computed only for passing scenes. Let $n_{\text{readout}} = K - m$. The empirical distribution over readable, in-schema outcomes is

$$\hat{p}_{M,\text{readout}}(y) = \frac{1}{n_{\text{readout}}} \sum_{i=1}^{K} \mathbb{1}[y_i = y], \qquad y \in \mathcal{Y}.$$

The readout-failure label $\perp$ is not included in this conditional distribution.

For PAW-Calibration, each scenario specifies a reference distribution q over its valid terminal outcomes. We measure valid-only conditional total variation distance,

$$\text{TVD}_{\text{readout}}(\hat{p}_{M,\text{readout}}, q) = \frac{1}{2} \sum_{y \in \mathcal{Y}} \left| \hat{p}_{M,\text{readout}}(y) - q(y) \right|.$$

Lower TVD indicates closer probability-mass alignment among readable, in-schema outcomes. We report $100 \times$ TVD.

For PAW-Coverage, the valid outcome set is enumerable but its probabilities are not specified. We measure valid-support recovery,

$$\text{Cov}_{\text{readout}}(\hat{p}_{M,\text{readout}}, \mathcal{Y}) = \frac{|\{y \in \mathcal{Y} : \hat{p}_{M,\text{readout}}(y) > 0\}|}{|\mathcal{Y}|}.$$

Higher coverage indicates that repeated rollouts recover a broader set of valid terminal outcomes. We report coverage as a percentage. Because observed support depends on the rollout budget, all primary comparisons fix $K = 50$; Fig. 5 examines sensitivity to increasing the budget to $K = 100$.

Mechanism-group scores average the scene-level conditional metric over passing scenes in that group. The track-level *Avg.* instead averages over all passing scenes in the corresponding 25-scene track; it is not the simple average of the four mechanism-group scores. Scene Pass Rate is $n_{\text{pass}}/25$, where the denominator includes the complete track roster. Conditional alignment scores and Scene Pass Rate should therefore be read together. PAW-Calibration and PAW-Coverage remain separate and are not combined into a single ranking.

D.4 Robustness to the Choice of Outcome Evaluator

To verify that our core findings are robust to the underlying outcome evaluator, we re-execute the benchmark using the open-weight Qwen3.8-27B (Qwen Team, 2026b). All experimental parameters, including the generated videos, rollout budget ($K = 50$), readout thresholds, reference distributions, and aggregation protocols, remain identical to the primary setup. For this evaluation, Qwen processes a three-frame sequence (initial state, action, and terminal state) under the `xhigh` reasoning configuration with no token generation limits.

Tab. 8 details the performance across all twelve generators, under the Qwen evaluator, Calibration TVD spans 23.7 to 44.5, while Coverage ranges from 50.4% to 73.7%. Although the top-performing Calibration model shifts from Cosmos 3 Super I2V (under Gemini) to LTX-2.3 Dev (under Qwen), the generators systematically fail to match reference probabilities and persistently leave valid physical outcomes unexplored. Furthermore, seemingly competitive conditional scores often obscure fundamental generation failures. For instance, while LTX-2.3 Dev achieves the lowest TVD (23.7), this score is conditioned on a mere 40% scene pass rate in Calibration. Similarly, its leading Coverage of 73.7% applies only to the 76% of successfully parsed scenes.

These results demonstrate that the macroscopic trends remain invariant across different vision-language models. Both evaluators independently expose pervasive probability mismatches, restricted outcome coverage, and frequent failures in rendering coherent, in-schema physical states. Consequently, this robustness check confirms that the observed deficit in probabilistic alignment is an intrinsic limitation of current video generators, rather than an artifact of the evaluation model.

Table 8: **PAWBench results with Qwen3.8-27B outcome readout.** Metrics follow Tab. 1. Avg. and mechanism scores are conditional on passing scenes; SPR uses all 25 scenes in each track. Red and blue mark the best and second-best conditional scores among the openly released models; NA denotes no passing scene.

Model	PAW-Calibration: TVD ×100 (↓)						PAW-Coverage: Coverage (%) (↑)					
	Avg.	SPR	Toss	Rot.	Rout.	Draw	Avg.	SPR	Coll.	Stab.	Agent	Mat.
HappyHorse	44.5	92.0%	47.7	10.4	42.7	59.5	53.6	96.0%	56.9	36.8	56.6	57.5
Veo3.1 Fast	35.9	92.0%	43.8	8.2	29.1	54.8	53.2	100.0%	57.9	31.1	58.8	53.3
Kling 3 Std.	35.7	96.0%	34.3	22.1	36.7	46.8	54.2	96.0%	57.0	52.0	46.8	65.8
Seedance 2	31.0	92.0%	34.6	13.5	29.5	39.7	57.8	84.0%	58.5	50.0	51.4	71.7
Wan2.7	28.1	92.0%	33.0	8.6	26.4	36.4	57.6	100.0%	59.4	65.4	46.5	68.3
Wan2.2	27.6	80.0%	35.1	11.2	**20.9**	38.0	66.8	100.0%	73.6	63.2	**61.1**	66.7
LTX-2.3 Dev	**23.7**	40.0%	44.2	3.9	23.3	23.6	**73.7**	76.0%	**91.7**	**76.7**	55.4	90.0
LTX-2.5 Dev	28.6	72.0%	**29.9**	9.4	35.0	27.7	63.8	92.0%	70.1	53.0	57.7	75.8
Cosmos 3 Super I2V	25.7	84.0%	30.4	**2.1**	22.2	34.1	60.0	96.0%	69.8	31.1	49.4	**90.8**
LingBot-Video-MoE	42.4	72.0%	41.2	10.4	41.6	61.0	60.9	92.0%	65.8	54.3	54.5	71.7
LingBot-World V2	25.0	44.0%	32.1	12.1	NA	**12.8**	50.4	84.0%	37.3	36.7	59.7	61.7
MiniMax H3	26.8	72.0%	33.3	14.1	24.7	26.7	55.4	92.0%	62.7	39.5	44.1	75.0

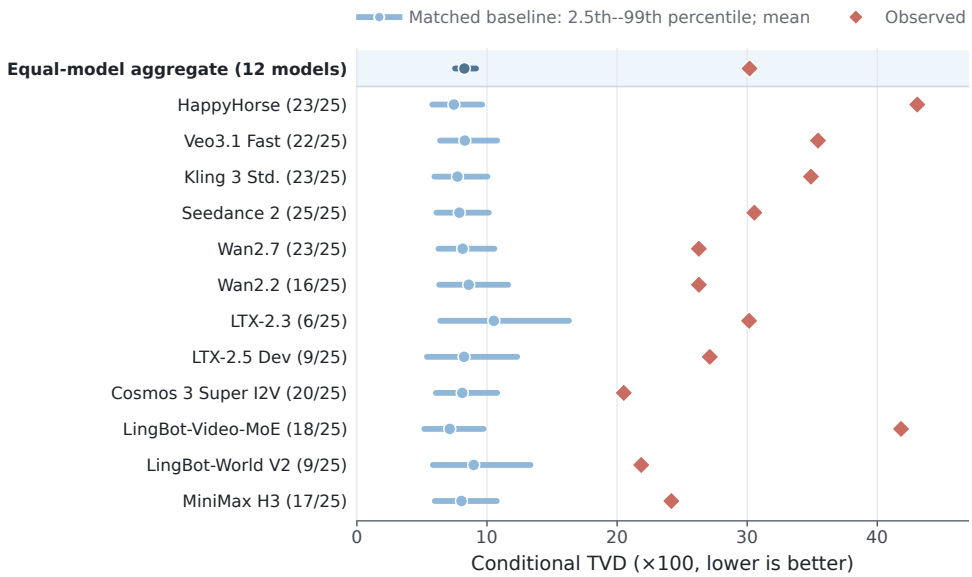


Figure 12: **Observed TVD exceeds the matched finite-sample baseline.** Blue intervals show the 2.5th to 99th percentiles obtained by sampling from the reference distributions with each model's passing scenes and readable sample counts. Circles mark the simulated means, and red diamonds mark the observed model TVDs. The aggregate row first averages passing scenes within each model and then weights all twelve models in Table 1 equally.

D.5 Finite-Sample Effects on Calibration

Because PAW-Calibration estimates an outcome distribution from a finite number of readable rollouts, even samples drawn from the reference distribution q can produce nonzero TVD. We measure this sampling error with matched Monte Carlo simulations. Models pass different sets of scenes and yield different numbers of readable rollouts, so we construct a separate matched baseline for each model. For each passing model–scene pair, we draw the same number of readable outcomes from the scene's reference distribution and compute TVD using the same aggregation as in the main evaluation. We include all twelve models in Table 1, and use 50,000 Monte Carlo replicates.

Figure 12 compares the observed model TVDs with these matched baselines. Across these twelve video generators, the observed TVD averages 30.2. Samples drawn from the reference distributions

Table 9: **Finite-sample null sensitivity.** The primary and stricter analyses weight models equally, while the pooled analysis weights passing model and scene cells equally. All rows use $B = 50000$ replicates and the model set evaluated in Table 1. p_{MC} denotes the plus-one upper-tail probability.

Aggregation	Models	Cells	Observed	Null mean	Null q99	Exceedances	p_{MC}
Equal-model (primary)	12	211	30.2	8.26	9.16	0/50,000	2.0×10^{-5}
Pooled passing cells	12	211	31.1	8.05	8.82	0/50,000	2.0×10^{-5}
Equal-model, $n_{\mathrm{readout}} \geq 30$	12	162	31.4	6.81	7.63	0/50,000	2.0×10^{-5}

average 8.26 TVD, and 99% of the simulated averages remain below 9.16. Each generator's observed TVD also exceeds the 99th percentile of its own matched baseline. Finite sampling therefore contributes to measured TVD, but it is too small to explain the observed calibration gap. Table 9 shows that the same conclusion holds under alternative aggregation and a stricter minimum-readout requirement. The baseline assumes independent samples from fixed references and measures sampling error in conditional Calibration TVD. It does not account for evaluator error, dependent rollouts, or uncertainty in the references.

D.6 Full Causal and Non-Causal Controls

Figs. 13 through 16 expand the paired-control diagnostic in Fig. 3 to eight controlled pairs. The evaluation includes all twelve video generators from the main benchmark alongside five vision-language models that predict future outcomes from identical initial states and actions. For every model and scene, we draw $K = 50$ samples and map readable responses to fixed terminal outcomes. Because video generators undergo PAWEval video readout while VLMs output semantic predictions directly, this design supports a diagnostic analysis of paired response directions rather than direct cross-modal performance comparisons.

By definition, a causal intervention alters the physical mechanics and reference distribution, necessitating a corresponding shift in the model's predictions. Conversely, a non-causal intervention leaves the physical rules intact, dictating that a true world model's distribution should remain stable.

Our extended evaluation, however, reveals a systemic failure in capturing this physical causality. Models generally fail to adjust their outcome distributions adequately in response to genuine causal interventions, while simultaneously exhibiting high vulnerability to non-causal visual perturbations. Additionally, a substantial number of intervention pairs fail the outcome-readout gate in at least one condition, fundamentally precluding reliable paired comparisons across varying physical states.

E Human Study and PAWEval Alignment

We compare Gemini 3.5 Flash (Google DeepMind, 2026) and Qwen3.8-27B (Qwen Team, 2026b) with human judgments on 1,459 videos from twelve generators across all 50 PAWBench scenes. Annotators on Rapidata[3] select the final visible outcome without seeing either evaluator's prediction. We use seven responses per video and require at least four to agree on a physical outcome. The comparison includes videos with a human majority label and readable, in-schema outcome labels from both evaluators.

Gemini agrees with the human majority on 81.8% of these videos, and Qwen on 79.8%. Agreement increases with human consensus, reaching 95.2% and 92.4%, respectively, when all seven responses agree (Table 10 and Fig. 17). These results support automatic outcome readout for videos with clear physical outcomes.

F Intervention Experiments

We vary the generator prompt, sampling noise, and training outcome frequencies in the experiments of Sec. 5.

[3] https://www.rapidata.ai/

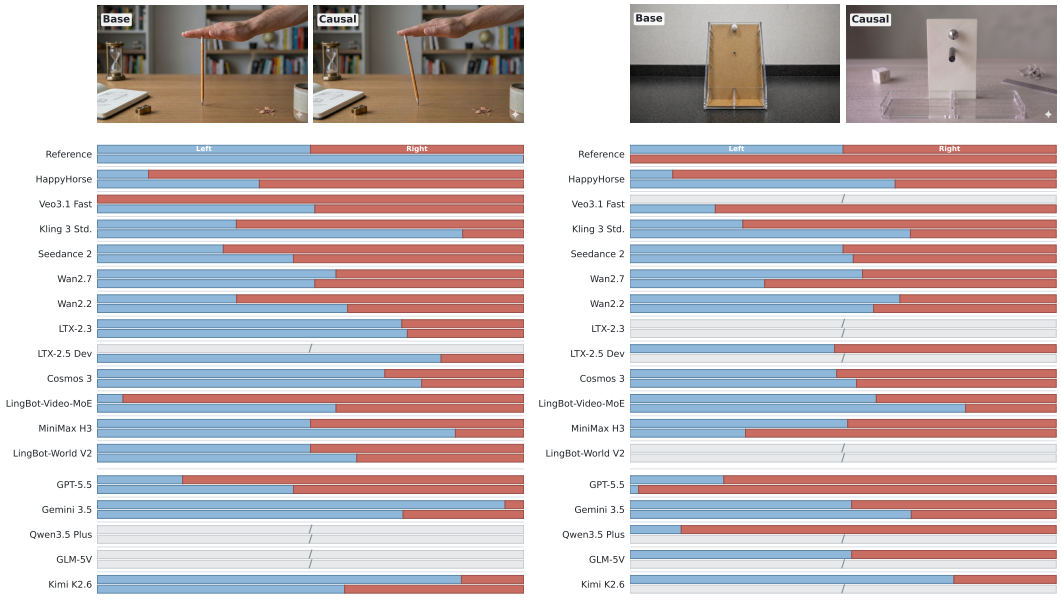


Figure 13: **Causal-state controls across video generators and direct VLM future samplers.** Each panel compares the outcome distribution for a base scene (upper bar) with that for a causal variant (lower bar), whose reference distribution changes. Bars report distributions conditional on readable, in-schema outcomes; gray bars marked / denote scenes that fail the outcome-readout gate.

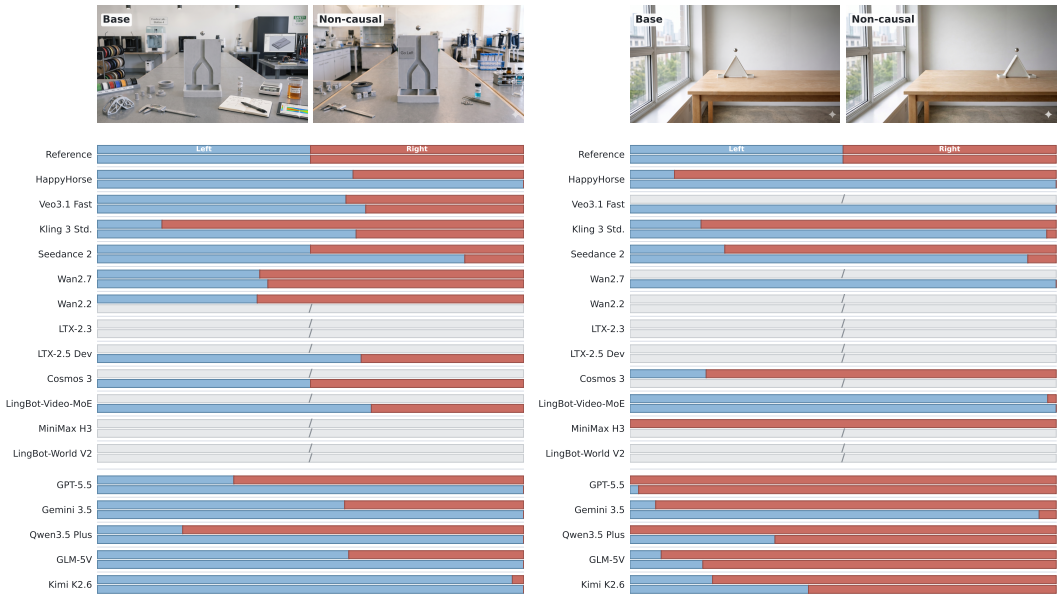


Figure 14: **Non-causal routing controls.** The paired scenes change an irrelevant cue while preserving the physical transition and reference distribution. Upper and lower bars show the base and cue-perturbed outcome distributions, respectively; gray bars marked / denote scenes that fail the outcome-readout gate.

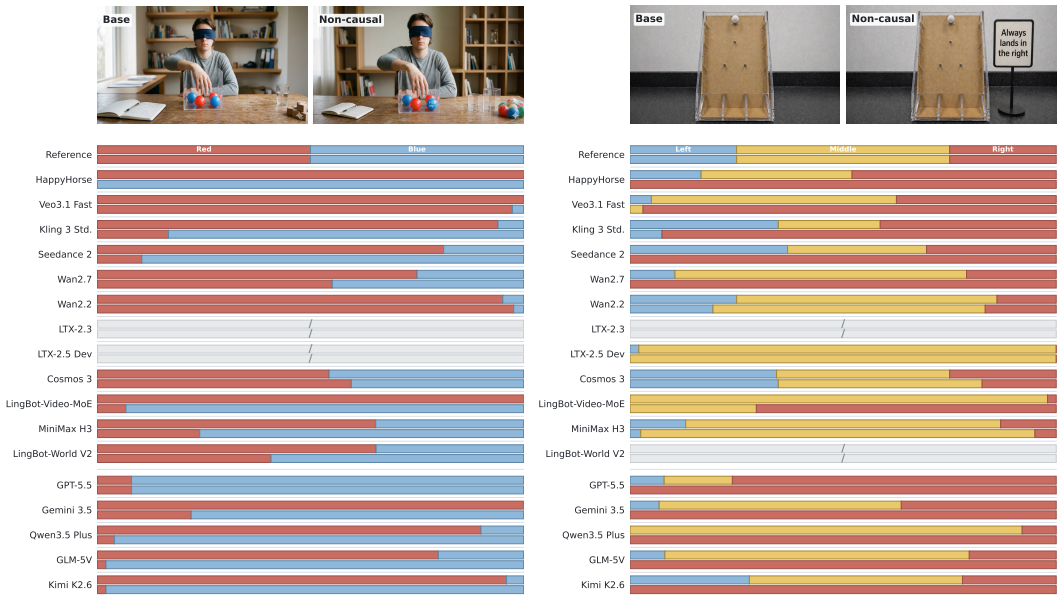


Figure 15: **Non-causal draw and text controls.** Distractor appearance in the blind-draw scene and an outcome-suggestive sign in the Galton-board scene leave the reference distribution unchanged. Several systems nevertheless move substantial probability mass between the paired conditions.

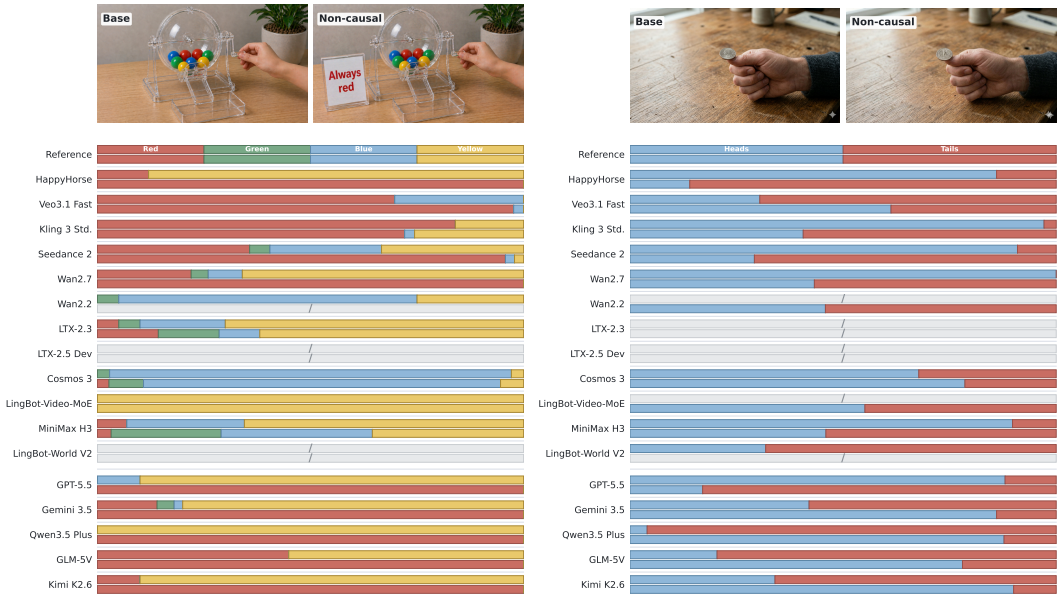


Figure 16: **Non-causal dispenser and coin controls.** Outcome-suggestive text or an irrelevant appearance change leaves the underlying chance process and reference distribution unchanged. Upper and lower bars show the base and cue-perturbed distributions conditional on readable, in-schema outcomes.

Table 10: **Human agreement under stricter consensus thresholds.** Both evaluators agree more often with humans as the required consensus increases.

Human votes	Paired videos	Gemini matches	Qwen matches
$\geq 4/7$	1,459	1,193 (81.8%)	1,164 (79.8%)
$\geq 5/7$	1,097	959 (87.4%)	928 (84.6%)
$\geq 6/7$	795	733 (92.2%)	705 (88.7%)
$\geq 7/7$	419	399 (95.2%)	387 (92.4%)

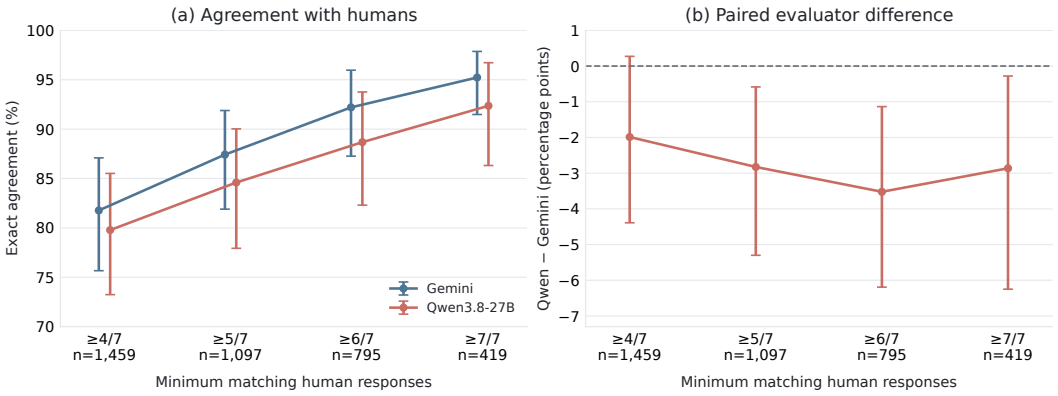


Figure 17: **Evaluator agreement versus human consensus.** Left: exact human-label agreement for Gemini and Qwen on the same study videos. Right: paired Qwen-minus-Gemini agreement in percentage points. Each threshold requires at least the indicated number of matching responses from a fixed seven-response panel; labels report the actual video count. Error bars are pointwise 95% intervals from bootstrapping scenes. The dashed line denotes zero paired difference.

Table 11: **Experimental settings for the Section 5 probes.** Each row records the evaluated systems, prompt interface, sampling budget, and readout; the corresponding constructions follow below.

Probe	Systems	Prompt interface	Sampling	Readout and scope
VLM future sampling	Qwen3.5 Plus, GPT-5.5, GLM-5V Turbo, Kimi K2.6, Gemini 3.5 Flash	Fixed VLM scaffold; source image and action prompt are inserted at runtime	25 scenes per track; $K = 50$ queries; hosted-model randomness	Fixed GPT-5.5 projection; a scene fails above 30 unmapped or invalid responses
PE / Oracle PE	Wan2.2, Cosmos 3 Super I2V, MiniMax H3, LTX-2.5 Distilled	GPT-5.5 predicts an outcome and writes the PE prompt; Oracle PE manually places scheduled targets in the prompts	25 scenes per track; $K = 50$ rollouts under each condition	PAWEval target hit plus Calibration TVD or Coverage; all current model–track pairs
C2C	Wan2.2, LTX-2.3, Cosmos 3 Super I2V	Base prompt unchanged	$G = 10$ groups of $m = 5$; matched $K = 50$ IID gallery	Common PAWEval protocol over all 25 scenes in both tracks
LoRA	MiniMax H3, Wan2.2, and LTX-2.5 PT; five rank-32 adaptations per mechanism	Fixed first frame and neutral action; pencil, spinner, and ball	$K = 50$ attempts per profile and state; Base + five mixtures; 108 conditions	Conditional outcome probability and valid-outcome fraction; Wan2.2 pencil uses an outcome-only evaluator

F.1 Language-Side Diagnostics and Interventions

We evaluate language at two interfaces: predicting possible outcomes without video synthesis and requesting a specific outcome from a video generator. In PE, GPT-5.5 predicts a possible outcome from the initial observation and action without access to the reference distribution, then writes a generator prompt that requests this outcome. In Oracle PE, we instead write the target outcome directly into the generator prompt. These settings separate two errors: predicting the wrong outcome distribution and failing to produce a requested outcome in video.

Prompts for Language-Side Probes

Fixed prompt scaffolds

Inputs filled at runtime

```
Runtime inputs:
- Source image: attached to the request
- Original prompt: {original_prompt}
- Target outcome: {target_outcome_description}  [target-conditioned generation only]
```

Future prediction (VLM)

```
You are given one image showing the initial state of a physical scene.
Action:
{original_prompt}
Your task is to imagine one concrete, physically plausible future after the action.
Rules:
- Do not ask for or enumerate alternative futures.
- Do not output probabilities, frequencies, or outcome labels.
- Describe the system after it has stabilized, focusing on the final readable physical
  result.
- Commit to one concrete, stable, visually readable terminal state that would be visible
  in a generated video.
- Do not describe an in-progress, ambiguous, hidden, or unreadable final state.
- Keep the description grounded in the image and action.
- The video prompt should be a concise instruction that could be passed to an
  image-to-video generator.
Output format:
<future_description>{detailed description of one concrete, stable, visually readable
terminal future}</future_description>
<video_prompt>{concise image-to-video prompt for that future}</video_prompt>
```

Target-conditioned generation (I2V)

```
Using the provided image as the first frame, create one
continuous real-time video. Use a locked-off stable camera.
{original_prompt}
Keep the full interaction area and all key objects visible
throughout. Preserve the visual style, lighting, object
identities, and layout of the input image.
{target_outcome_description}
By the final moment, all final positions, spatial relations,
and object identities are clearly readable.
```

Braced fields are replaced per scene; all remaining text is fixed.

Figure 18: **Prompts for the two language-side probes.** The VLM scaffold requests one plausible future without exposing outcome labels or probabilities. The target-conditioned I2V scaffold instead adds one terminal outcome to the original scene prompt. Braced fields are filled at runtime.

VLM sampling without video. We first test whether repeated VLM predictions produce the required outcome distribution before video generation. For each of the 25 scenes in both tracks, we show a VLM the same source image and action prompt and ask it to describe one possible outcome, repeating the query $K = 50$ times. A fixed GPT-5.5 projector maps these free-form predictions to the scene-specific PAWBench outcome set, yielding an empirical distribution that can be evaluated against the scene's outcome specification.

Only responses mapped to valid outcomes contribute to this conditional distribution. Unmapped or invalid responses instead count toward the scene gate, which fails a scene when they exceed 30 of the 50 queries. Over passing scenes, Calibration reports valid-only TVD and Coverage reports support recovery; SPR separately gives the fraction of the 25 scenes that pass.

Prompt Engineering(PE). For each rollout, GPT-5.5 predicts one possible outcome from the initial observation and action, then writes a generator prompt requesting that outcome. It receives neither the reference distribution nor a target outcome from the benchmark.

Oracle PE. For Oracle PE, we manually specify one PAWBench target outcome in every generator prompt. For PAW-Calibration, we convert the reference probabilities into $K = 50$ target requests using largest-remainder allocation. For PAW-Coverage, we divide the 50 requests as evenly as possible across the valid outcomes. We deterministically permute each target schedule across rollout indices and reuse the same schedule for every generator. Across the 25 scenes in each track, this gives 2,500 target-conditioned rollouts for each generator. PAWEval reads the terminal outcome, and a rollout counts as a target hit only when that outcome matches the request; unreadable, invalid, missing, and different outcomes are all misses. We then ask a separate distributional question by comparing the resulting Calibration TVD and Coverage with matched Base rollouts. Target-hit rate measures whether a generator follows an individual outcome request, whereas the PAWBench metrics measure the aggregate distribution produced across requests.

Native PE. Native PE does not name a benchmark outcome. Instead, we invoke the prompt processors distributed with Cosmos 3 Super I2V and MiniMax H3 and pass their outputs through each model's standard conditioning pathway (NVIDIA, 2026; MiniMax, 2026). These processors produce different forms of conditioning. Tab. 12 summarizes these interfaces: H3 Context-IR and Cosmos paired positive–negative prompts. Native PE is therefore a family of model-specific interfaces, not one shared intervention whose effect can be pooled across generators.

Tab. 14 reports Base, GPT-5.5 PE, and Oracle PE results for Wan2.2, Cosmos 3 Super I2V, MiniMax H3, and LTX-2.5 Distilled. All conditions cover the 25 scenes in each track at $K = 50$.

Table 12: **Native PE across generators.** Each model-specific enhancer transforms the original PAWBench request into the conditioning consumed by its paired generator.

Generator	Native PE	Input to video generator	Link
MiniMax H3	Official H3-Context-IR from the source image and original prompt	Returned Context-IR prompt	Official
Cosmos 3 Super I2V	Official Prompt Upsampling from the source image and original prompt	Upsampled positive and negative prompts	Official

Table 13: **Native prompt processing versus Base.** Native PE increases Calibration TVD for both generators. and indicate the better and worse conditional scores within each Base/Native PE pair; ties share the better-score color.

Model	PAW-Calibration: TVD ×100 (↓)						PAW-Coverage: Coverage (%) (↑)					
	Avg.	SPR	Toss	Rot.	Rout.	Draw	Avg.	SPR	Coll.	Stab.	Agent	Mat.
Cosmos 3 Super I2V	**20.5**	80.0%	**19.4**	**3.5**	**17.7**	**35.5**	**55.2**	92.0%	56.7	31.7	**49.4**	**81.7**
+Native PE	32.3	92.0%	36.0	14.1	28.0	47.1	**55.2**	96.0%	**64.5**	**41.6**	46.5	67.5
MiniMax H3	**24.2**	68.0%	**22.0**	**14.0**	**28.9**	**26.5**	**48.7**	92.0%	**59.5**	**39.5**	36.4	**58.3**
+Native PE	38.3	100.0%	39.5	24.8	33.6	60.0	39.5	96.0%	43.8	21.1	**38.5**	51.7

Native prompt processing increases Calibration TVD. For MiniMax H3 and Cosmos 3 Super I2V, the Base and Native PE conditions retain the source image and action, generator, and $K = 50$ attempts per scene, with the same reevaluation rubric within each pair. Metrics consume only terminal-outcome judgments; process-trustworthiness scores do not filter the outcome distribution. A scene passes when at most 30 of its 50 attempts are unreadable or out of schema. TVD and Coverage are averaged over each condition's passing scenes, and SPR retains the full 25-scene denominator. Thus, the mean differences also reflect changes in which scenes pass.

For MiniMax H3, Native PE raises TVD from 24.2 to 38.3 and lowers Coverage from 48.7% to 39.5%. For Cosmos 3 Super I2V, TVD rises from 20.5 to 32.3 while Coverage remains 55.2% after rounding. Native PE therefore worsens conditional Calibration for both tested generators, without improving their aggregate Coverage.

Table 14: **Predicted outcomes do not substitute for target outcomes.** The first row scores GPT-5.5 PE predictions before video synthesis. Red/blue mark best/second within each generator across Base, PE, and Oracle PE. All conditions use 25 scenes per track and $K = 50$.

Model	PAW-Calibration: TVD ×100 (↓)						PAW-Coverage: Coverage (%) (↑)					
	Avg.	SPR	Toss	Rot.	Rout.	Draw	Avg.	SPR	Coll.	Stab.	Agent	Mat.
GPT-5.5 PE	44.3	100.0%	40.1	49.5	43.6	49.0	35.0	100.0%	35.4	34.6	35.5	33.3
Wan2.2	26.3	64.0%	22.4	15.8	23.2	45.9	63.4	92.0%	**76.8**	56.7	55.0	58.3
+PE	27.1	**92.0%**	23.5	15.4	28.8	39.5	61.9	**100.0%**	68.1	50.7	62.9	57.5
+Oracle PE	**15.3**	84.0%	**10.5**	**13.9**	**14.0**	**27.3**	**76.2**	96.0%	71.1	**72.7**	**82.8**	**76.7**
Cosmos 3 Super I2V	20.5	80.0%	19.4	**3.5**	17.7	35.5	55.2	92.0%	56.7	31.7	49.4	**81.7**
+PE	31.6	**92.0%**	27.0	15.0	34.9	47.0	64.9	**100.0%**	71.1	62.0	53.5	76.7
+Oracle PE	**12.8**	88.0%	**17.6**	9.2	**13.2**	**5.1**	**87.3**	**100.0%**	**84.8**	**91.4**	**91.3**	80.8
MiniMax H3	24.2	68.0%	22.0	14.0	28.9	26.5	48.7	92.0%	59.5	39.5	36.4	58.3
+PE	38.9	**96.0%**	32.5	35.0	42.2	48.6	49.8	**96.0%**	59.3	41.8	40.5	57.5
+Oracle PE	**10.8**	**96.0%**	**21.6**	**6.0**	**2.1**	**12.6**	**82.9**	**96.0%**	**82.4**	**91.4**	**82.1**	**76.7**
LTX-2.5 Distilled	30.2	60.0%	30.2	**17.1**	39.9	27.1	57.4	88.0%	**66.7**	30.0	59.4	57.5
+PE	36.4	**96.0%**	34.3	28.7	36.6	48.0	55.6	**100.0%**	59.8	42.1	51.4	68.3
+Oracle PE	**18.0**	88.0%	**26.2**	21.8	**15.0**	**5.7**	**73.9**	96.0%	56.0	**78.0**	**89.9**	**73.3**

Language steers outcomes but does not determine their distribution. Tab. 3 shows that repeated VLM predictions remain misaligned with the reference distribution. Tab. 14 shows that PE raises

Calibration TVD relative to Base for all four generators and improves Coverage for only two, whereas Oracle PE lowers mean Calibration TVD and raises mean Coverage for all four. Fig. 6 nevertheless shows that only 37.6–58.1% of their rollouts reach the requested outcome. The results expose both errors: the language controller predicts the wrong outcome distribution, and the video generator often fails to produce a requested outcome. Language can change individual rollouts without reliably aligning their aggregate distribution.

F.2 Coupled Noise Sampling

The initial noise of a diffusion generator is usually sampled independently across rollouts. C2C asks a narrower question: can a finite gallery cover the generator's accessible futures more evenly when these noises are coupled, while preserving the standard-Gaussian marginal seen by every rollout?

Marginal-preserving coupling. For group g, let $\epsilon_{g,1}, \ldots, \epsilon_{g,m}$ be independent standard-Gaussian noise tensors. Following the centered construction of Couple to Control (Jia et al., 2026), we form

$$\epsilon_{g,i} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, I), \qquad \widetilde{\epsilon}_{g,i} = \sqrt{\frac{m}{m-1}} \left(\epsilon_{g,i} - \frac{1}{m} \sum_{j=1}^{m} \epsilon_{g,j} \right). \tag{1}$$

Each $\widetilde{\epsilon}_{g,i}$ remains marginally distributed as $\mathcal{N}(0, I)$, while the samples within a group satisfy $\sum_i \widetilde{\epsilon}_{g,i} = 0$ and $\mathrm{Cov}(\widetilde{\epsilon}_{g,i}, \widetilde{\epsilon}_{g,j}) = -I/(m-1)$ for $i \neq j$. Groups are constructed independently. The intervention therefore changes the joint distribution of the gallery without changing the noise distribution of any single rollout.

Experimental construction. We use $G = 10$ independent groups with $m = 5$ samples per group, yielding $K = Gm = 50$ rollouts for every model–scene pair. The coupled tensors are constructed at each generator's native initial-latent shape and supplied at the start of denoising. The matched IID condition instead draws K independent tensors from the same $\mathcal{N}(0, I)$ marginal; the source image, prompt, generator, and remaining generation settings are held fixed. Tab. 5 reports Wan2.2, LTX-2.3, and Cosmos 3 Super I2V on the full 25-scene PAW-Calibration roster and the full 25-scene PAW-Coverage roster. All C2C and IID rollouts follow the common PAWEval outcome-readout and aggregation protocol in Appendix D.3; the intervention changes only the dependence among initial noises, not the validity gate or metric denominator.

Because C2C preserves the marginal distribution of every initial noise, it does not alter the generator's learned one-rollout conditional law. It instead probes whether a finite IID gallery misses futures that the generator can already realize. Improved coverage under coupling is therefore evidence about finite-budget exploration, not about learning a better-calibrated world model.

F.3 Training-Distribution Intervention

To investigate whether parameter-efficient fine-tuning enables video generators to dynamically adjust their outcome distributions in response to physical state changes, we adapt MiniMax H3, Wan2.2, and LTX-2.5 Pre-Trained using LoRA across controlled outcome ratios. As detailed in Table 15 and Table 16, we construct training mixtures with 0%, 20%, 50%, 80%, and 100% positive outcomes while maintaining constant overall training exposure through video repetition. Each mixture configuration is trained once, and the resulting checkpoint is evaluated directly across different physical states without post-hoc selection, using the original training first frame for baseline state evaluations.

For LTX-2.5 PT, we use the official LTX trainer with AdamW, LoRA alpha 32, an effective batch size of one, and training seed 20260910. The learning rate is $10^{-4}/3$ for the first five updates and 10^{-4} thereafter. We uniformly sample 121 frames over each complete source video, including both endpoints, then resize with preserved aspect ratio and center padding to 832×480. Each mixture contains 1,000 video exposures repeated twice; evaluation uses the final 2,000-step checkpoint. Inference uses 1536×1024 resolution and the settings in Table 16.

The fine-tuning dataset comprises 100 synthetic videos per outcome for each scenario, shared across all three models. While synthesis is guided by endpoint target images, both training and evaluation strictly consume only the initial frame alongside a neutral action prompt. To guarantee data integrity, all generated sequences undergo VLM verification and manual frame inspection to enforce action completion, clear outcome readability, and temporal scene consistency.

Table 15: **Training data and physical states.** All three models share the same synthetic video pool for each scenario. q denotes the physical reference probability of the positive outcome.

Scenario	Positive outcome	Video synthesis	Training state (q)	Changed state (q)
Pencil	Left fall	GPT Image 2 + Seedance 2.0	Upright (0.5)	Left-leaning (1)
Spinner	Red	GPT Image 2 + Seedance 2.5	Half red (0.5)	Quarter red (0.25)
Ball	Leftward	GPT Image 2 + Seedance 2.5	Centered (0.5)	Right-shifted (0)

Table 16: **LoRA training and inference settings.** Inference settings refer to the adapted models.

Setting	MiniMax H3	Wan2.2	LTX-2.5 PT
LoRA rank (Hu et al., 2021)	32	32	32
Adapted layers	Attention and MLP	Both noise experts	Video attention and MLP
Learning rate	10^{-4}	10^{-4}	10^{-4}
Training updates	2,000	2,000 per expert	2,000
Training resolution	832×480	832×480	832×480
Training frames	124	49	121
Inference steps / CFG	50 / 1	50 / 5	30 / 3
Output frames / fps	124 / 24	49 / 15	121 / 24
Sigma shift / expert switch	—	5 / 0.875	—

To assess physical responsiveness, trained checkpoints are evaluated on modified physical initial states. In the ball scenario, the idealized reference model assumes release from rest under vertical gravity without horizontal impulse or external forces. In the perturbed state, shifting the initial position enables the ball to clear the apex and contact the right slope, yielding a deterministic reference probability of $q = 0$ for leftward outcomes. This controlled modification establishes a rigorous baseline to determine whether fine-tuned models track physical state changes or simply memorize training distribution priors.

G Qualitative Examples and Failure Cases

G.1 Qualitative Examples on PAWBench

We present successful trials and failure cases from the main PAWBench evaluation in Figs. 19–31. The failures include a ball stopping at a track junction and a pencil changing appearance during its fall. Each caption describes the requested action and the observed behavior.

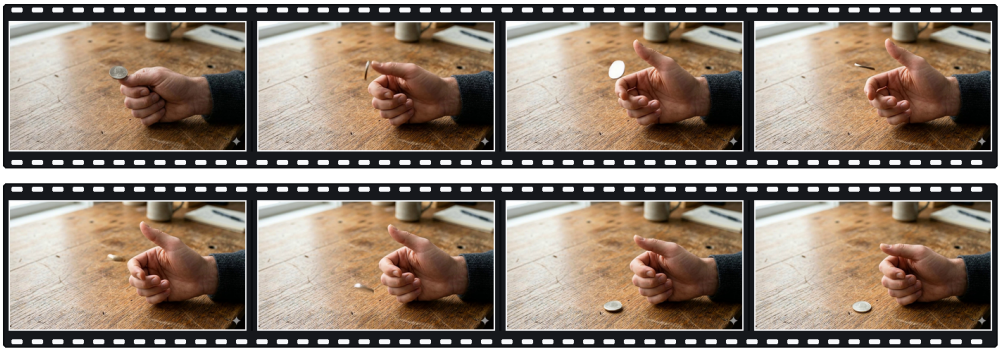


Figure 19: **Example of the scene "Coin flip" generated by HappyHorse.** The model is instructed to flick the coin once. The generated rollout ends with the coin lying heads-up on the table.



Figure 20: **Example of the scene "Die odd/even toss" generated by Kling 3 Std.** The model is instructed to throw the die once onto the table. The generated rollout ends with the die settled on the table, showing five pips on top.



Figure 21: **Example of the scene "Two-color equal-sector spinner" generated by Veo3.1 Fast.** The model is instructed to spin the wheel once under the fixed pointer. The final frame places the blue sector under the pointer, but the spin is visually inconsistent across the rollout.

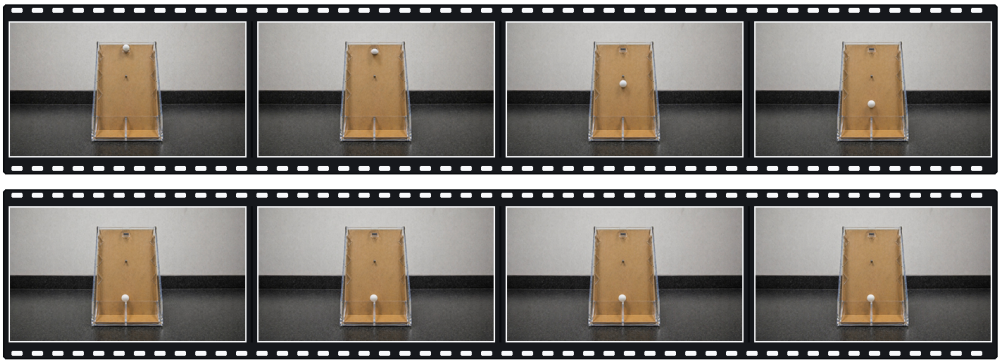


Figure 22: **Example of the scene "One-peg Galton board" generated by Wan2.2.** The model is instructed to release the ball once. The ball settles on the divider instead of entering either bin.

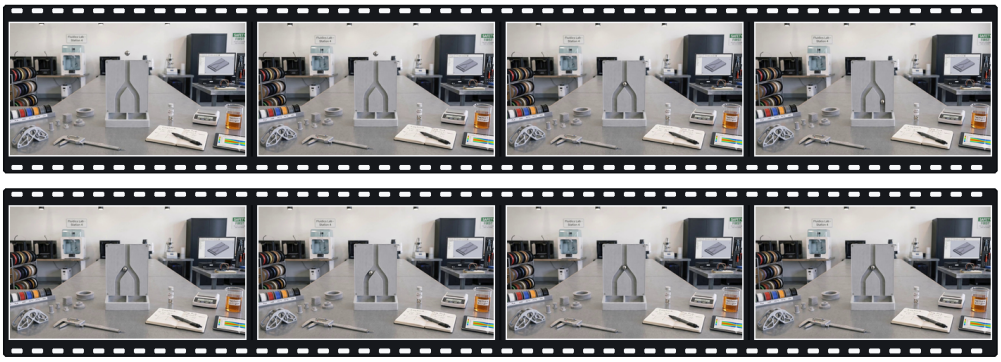


Figure 23: **Example of the scene "Y-track branch" generated by Seedance 2.** The model is instructed to release the ball once. The ball remains stuck at the split rather than traveling down one branch.

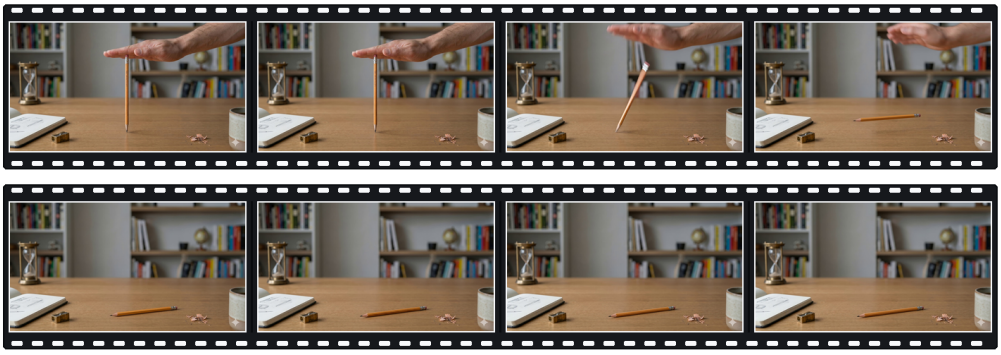


Figure 24: **Example of the scene "Vertical pencil fall" generated by Wan2.2.** The model is instructed to move the hand upward once and let the pencil fall. The generated rollout ends with the pencil falling to the right.



Figure 25: **Example of the scene "Blind ball draw" generated by Wan2.7.** The model is instructed to pick up exactly one ball once. The person selects a red ball and holds it up by the end of the rollout.

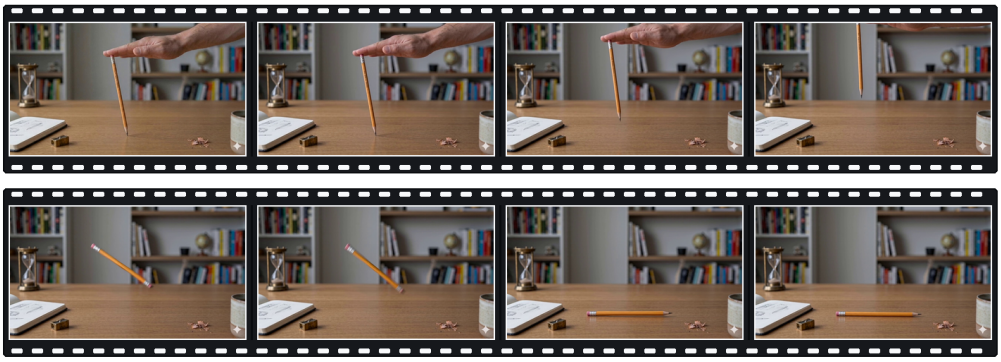


Figure 26: **Example of the scene "Left-Leaning Pencil Fall" generated by Seedance 2.** The model is instructed to move the hand upward once and let the pencil fall. The pencil appears to fall left, but its identity is not preserved cleanly through the motion.

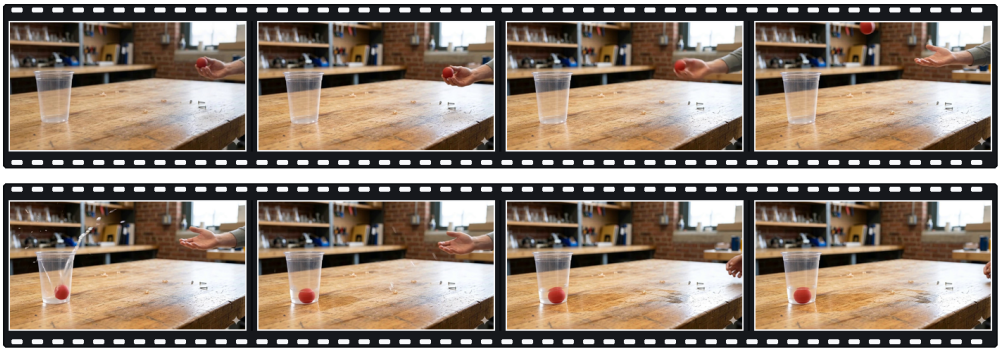


Figure 27: **Example of the scene "Ball Toss Into Cup" generated by HappyHorse.** The model is instructed to toss the ball once from the visible hand pose toward the cup. The generated rollout ends with the ball landing cleanly inside the cup.

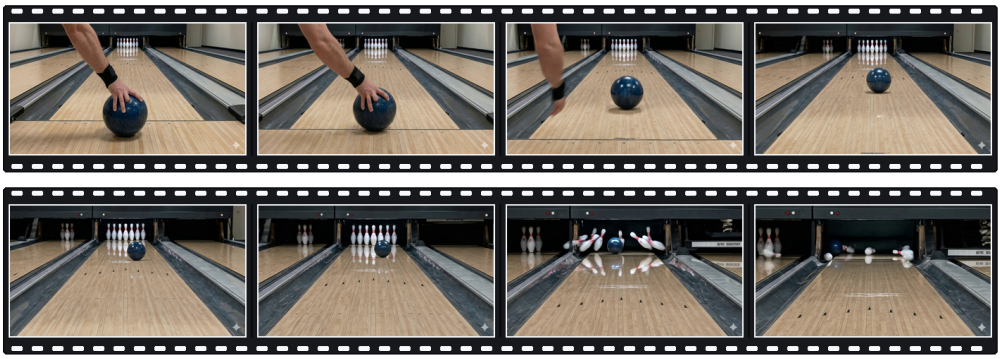


Figure 28: **Example of the scene "Seven-Pin Bowling Roll" generated by Kling 3 Std.** The model is instructed to roll the bowling ball once down the lane. The generated rollout ends with all seven pins knocked down.

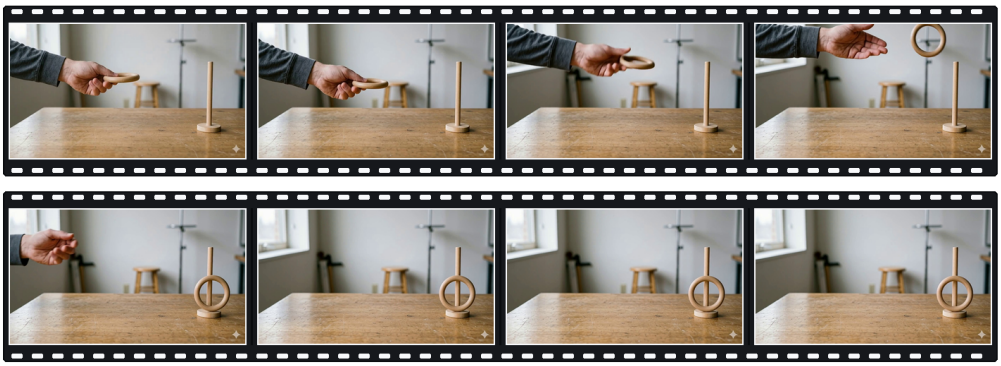


Figure 29: **Example of the scene "Ring toss toward peg" generated by MiniMax H3.** The model is instructed to toss the ring once from the visible hand pose toward the peg. The ring travels toward the peg and remains around it in the final frames.



Figure 30: **Example of the scene "Loose-yarn pull" generated by LTX-2.5 Base.** The model is instructed to pull the loose yarn end once. The loose yarn is pulled upward and visibly extends from the knitted fabric.

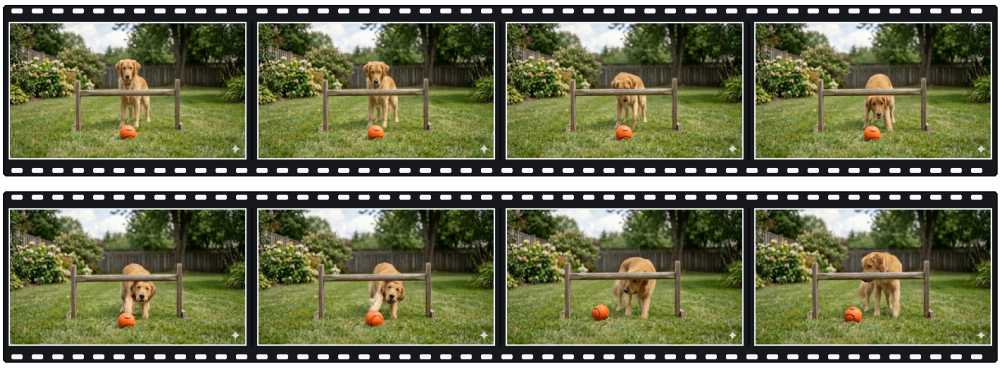


Figure 31: **Example of the scene "Dog Route Choice Toward Toy" generated by Cosmos 3 Super I2V.** The model is instructed to let the dog make one reach attempt toward the toy. The dog lowers its head under the bar, then backs away and remains behind the obstacle.

G.2 Qualitative Examples with Real-World Inputs

We show generated rollouts initialized from real-video frames for coin spinning, die tossing, and bottle flipping in Figs. 32–34. Each row contains the first sampled rollout from one model, with frames at equal time intervals. We report the outcome frequencies in Fig. 4.



Figure 32: **Coin spinning with real-world inputs.** Each row shows the first sampled rollout from one model, read from left to right. The model is instructed to spin the coin once on its edge.

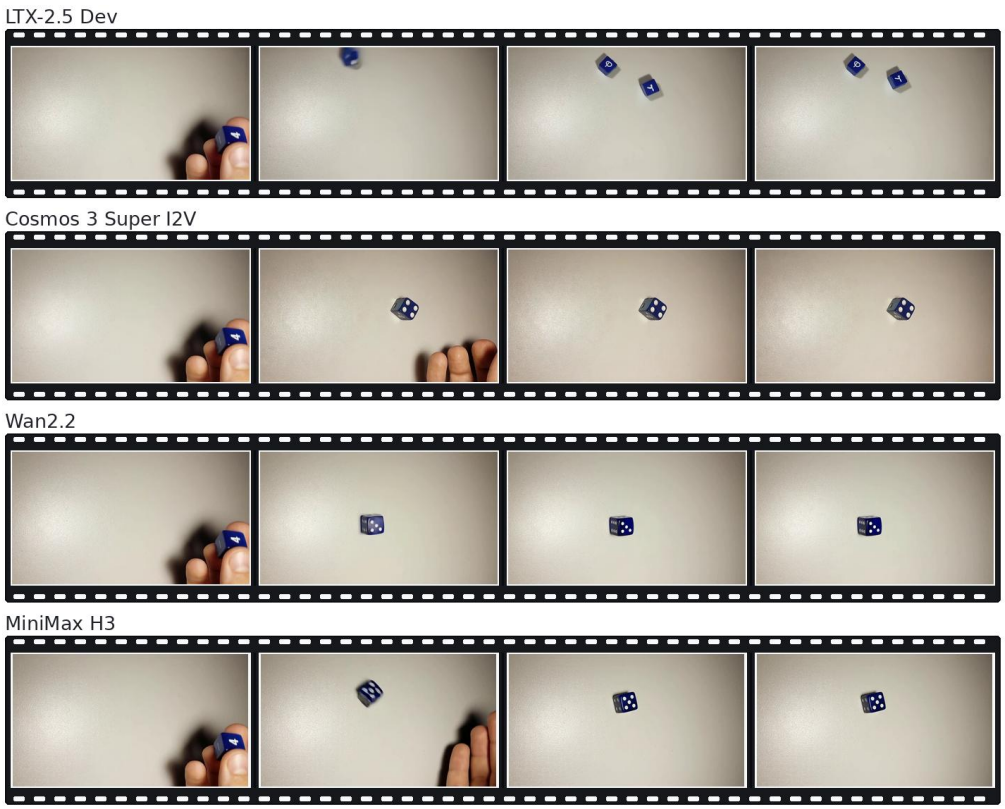


Figure 33: **Die tossing with real-world inputs.** Each row shows the first sampled rollout from one model, read from left to right. The model is instructed to throw the die once onto the table.



Figure 34: **Bottle flipping with real-world inputs.** Each row shows the first sampled rollout from one model, read from left to right. The model is instructed to toss the bottle with an end-over-end rotation and let it land freely.