RIGIDBENCH: EVALUATING RIGID-BODY PHYSICS IN VIDEO GENERATION MODELS

Swarnim Jain
University of Cambridge
sj697@cam.ac.uk

Shangzhe Wu
University of Cambridge
sw2181@cam.ac.uk

ABSTRACT

Video generation models are increasingly deployed as world model backbones for physical AI, yet their ability to predict rigid-body dynamics remains unreliable. Existing benchmarks either lack precise ground-truth annotations (relying on VLM judgment) or render synthetic primitives against plain backgrounds, introducing a visual domain gap from natural video. We introduce RIGIDBENCH, a benchmark combining Blender physics simulation with photorealistic interior scenes to provide exact 3D trajectories, segmentation masks, and depth maps across ten rigid-body physics tasks. Our evaluation protocol spans object localization, trajectory tracking, depth consistency, and perceptual quality, enabling controlled comparison across models. Evaluating seven models spanning open-source diffusion transformers and closed-source commercial systems, we find that trajectory accuracy and perceptual quality are essentially uncorrelated (r=0.002): models that best predict object motion often score worst on perceptual metrics. This demonstrates that standard video quality metrics cannot assess physical understanding, motivating evaluation with precise physics annotations. We further show that fine-tuning on RIGIDBENCH data improves physics prediction on held-out tasks, suggesting a path toward more physically grounded video generation.

1 INTRODUCTION

World models predict how environments evolve over time, enabling agents to plan by simulating future states before acting (Ha & Schmidhuber, 2018). Video generation models are increasingly proposed as world model backbones, learning scene dynamics and physical interactions from internet-scale video (OpenAI, 2024). This paradigm now underpins physical AI systems for robotic manipulation, autonomous driving, and embodied planning (NVIDIA, 2025).

However, for video models to serve as reliable world simulators, they must accurately predict physical dynamics. Current models fail at basic physics (Motamed et al., 2025; Bansal et al., 2024): objects hover mid-air, collisions violate momentum conservation, and multi-step causal chains break down. These failures persist even when all information needed for prediction is visible: Figure 2 shows models predicting incorrect trajectories on a simple ramp task despite having the full initial scene. When world models guide real-world actions, such physics errors propagate into unsafe behavior.

Rigorously evaluating physical understanding requires benchmarks with precise ground-truth annotations, yet existing approaches each have limitations. Text-prompt benchmarks like PhyGenBench (Meng et al., 2025), VideoPhy (Bansal et al., 2024), and T2VPhysBench (Guo et al., 2025) assess physical plausibility through VLM-based or human judgment, lacking ground-truth trajectories for quantitative evaluation. Real-video benchmarks like Physics-IQ (Motamed et al., 2025) and Morpheus (Zhang et al., 2025) capture authentic dynamics but provide limited object-level annotations. PISA (Li et al., 2025) offers ground-truth trajectories through Kubric simulation (Greff et al., 2022), but evaluates only freefall onto flat ground with HDRI backgrounds rather than complete scenes.

We introduce RIGIDBENCH, a benchmark designed around two complementary goals: **(1)** precise evaluation through exact ground-truth object trajectories for rigid-body dynamics, collisions, and momentum transfer; and **(2)** photorealism through complete interior scenes with realistic 3D objects,

closing the sim-to-real gap. Our key insight is that photorealistic synthetic data provides both the precision needed for quantitative evaluation and the visual fidelity needed to match the distribution of internet video that models are trained on. Indoor scenes further close the domain gap to the environments where robotic manipulation systems operate.

Our contributions:

- A **data generation pipeline** combining Blender physics simulation with photorealistic BlenderKit (BlenderKit, 2024) scenes and objects.
- **Ten physics tasks** spanning single-object freefall to multi-body collision cascades, with a train/eval split to measure generalization.
- **Comprehensive evaluation** of seven video generation models, spanning open-source diffusion transformers and closed-source commercial systems, revealing that trajectory accuracy and perceptual quality are decoupled.
- **Physics-focused fine-tuning pipeline** for Wan 2.2 on RIGIDBENCH data, enabling targeted post-training for physics prediction.

2 RELATED WORK

Physics Benchmarks for Video Generation. General video benchmarks like VBench (Huang et al., 2024) and VBench-2.0 (Huang et al., 2025) evaluate broad quality dimensions but assess physics only through VLM judgment, which lacks reproducibility and cannot provide training signal. PhyGenBench (Meng et al., 2025), VideoPhy (Bansal et al., 2024), and T2VPhysBench (Guo et al., 2025) use text prompts with VLM or human evaluation, but without ground-truth trajectories such assessments are noisy and non-reproducible. Physics-IQ (Motamed et al., 2025) uses 396 real videos with diverse physics scenarios but provides no object-level annotations for quantitative trajectory comparison. Morpheus (Zhang et al., 2025) proposes physics-informed metrics based on conservation laws but requires real-world video collection and style transfer for evaluation.

Most relevant to our work, PISA (Li et al., 2025) renders physics simulations with Kubric (Greff et al., 2022) using Google Scanned Objects (Downs et al., 2022), enabling exact ground-truth comparison. PISA demonstrates that supervised fine-tuning on just 5,000 simulated samples dramatically improves physics accuracy, and introduces reward optimization with segmentation, optical flow, and depth signals. However, PISA evaluates only freefall onto flat ground, uses HDRI backgrounds rather than complete 3D scenes, and shows that performance degrades on out-of-distribution depths and heights. RIGIDBENCH extends this direction with three key differences: (1) ten diverse physics tasks including rolling, collisions, and momentum transfer to test compositional generalization; (2) complete BlenderKit interior scenes rather than HDRI backgrounds to minimize sim-to-real domain gap; and (3) additional evaluation modalities (point tracking, depth) beyond mask-based metrics.

Video Generation Models. Video generation models achieve impressive visual quality through diffusion (Lipman et al., 2023; Liu et al., 2023) or autoregressive architectures. Open-source models include Wan 2.2 (Team Wan, 2025) (5B parameters, diffusion transformer) and Cosmos 2.5 (NVIDIA, 2025) (2B parameters, diffusion transformer). Closed-source models include Sora 2 (OpenAI), Veo 3.1 (Google), and Kling 2.6 (Kuaishou). We evaluate both open and closed-source models to characterize physics prediction across architectures.

Physics-Aware Video Generation. Several works incorporate physics knowledge into video generation. PhysGen (Liu et al., 2024) uses explicit rigid-body dynamics for image-to-video generation. PhysMaster (Ji et al., 2025) learns physics representations through reinforcement learning. Recent work (Le et al., 2025) proposes verifiable rewards for Newton's laws during post-training. Our fine-tuning study provides empirical evidence that training on precise physics data improves prediction on held-out tasks.

3 METHOD

3.1 DATA GENERATION PIPELINE

Unlike Kubric (Greff et al., 2022), which renders geometric primitives against HDRI backgrounds, we render within complete BlenderKit (BlenderKit, 2024) interior scenes featuring furniture, walls, and realistic lighting. Objects and materials are also sourced from BlenderKit (BlenderKit, 2024): nine assets with physically-based rendering (PBR) materials, including fruits (apple, orange, watermelon, lemon, coconut), sports equipment (baseball, tennis ball), and household items (soda can, toy duck).

Simulation. Objects are automatically placed to rest on surfaces, with rolling objects aligned to ramp slopes. We use Blender's Bullet physics engine with 20 substeps and 20 solver iterations per frame for accurate collision detection. Object physics parameters (mass, friction, restitution) and damping coefficients (linear: 0.1, angular: 0.5) are set to physically plausible values for each object category (Appendix F).

Rendering. Videos are rendered at 1280×704 resolution, matching Wan 2.2 TI2V-5B's (Team Wan, 2025) native resolution. Each sample comprises 49 frames at 24 FPS ($\sim$2 seconds) with per-object segmentation masks, depth maps, and 3D object trajectories. A text prompt generated from task-specific templates (e.g., "Apple rolls down a ramp toward orange") conditions each video generation model to produce a comparable scene.

Randomization. Camera position is computed from the bounding sphere of task objects, with azimuth randomized within task-specific ranges and elevation fixed at 10°. We randomize across multiple levels: five interior scenes provide varied backgrounds; task orientation spans 0–360°; object identities are sampled from the asset pool; and continuous parameters (ramp dimensions, drop heights, inter-object gaps) are drawn from specified ranges. This multi-level randomization creates combinatorial diversity: a single task like `ramp_to_target` has 5 rolling objects $\times$ 7 targets $\times$ 2 materials $\times$ continuous parameter ranges $\times$ 360° rotation, yielding effectively unlimited unique configurations from just 9 base objects. Deterministic seeding ensures reproducibility: a single seed fully specifies scene geometry, object placement, and physics simulation.

3.2 PHYSICS TASKS

A critical design choice in RIGIDBENCH is that *all tasks begin from static initial states*, with motion triggered solely by gravity acting on objects at rest. This constraint is essential for rigorous physics evaluation: when objects start stationary, all information needed to predict future motion is visible in the first frame. There are no hidden initial velocities to infer, no off-screen forces to estimate. The future trajectory is *deterministic* given the visible geometry and physical laws.

This design isolates **physics prediction** from **state estimation**: objects entering mid-motion would conflate inferring current state with predicting future state. Static initialization ensures failures directly indicate gaps in physics understanding rather than ambiguity in initial conditions.

We use two initialization mechanisms: **drops** (objects released from height) and **ramps** (objects placed on inclined surfaces). Drops test gravitational acceleration, air time estimation, and restitution upon landing. Ramps convert gravitational potential energy into kinetic energy through rolling, with exit velocity fully determined by visible ramp geometry (height, angle, length). Both mechanisms use gravity as the sole driving force, ensuring that a model with correct physics understanding could, in principle, predict exact trajectories from frame one.

RIGIDBENCH includes ten tasks spanning single-object dynamics to multi-body cascades (Table 1).

Tasks are organized by physics complexity: single-object dynamics (`free_fall`, `ramp_launch`), two-body interactions (`ramp_to_target`, `rolling_collision`), and multi-body cascades (`ball_chain`, `pyramid_topple`). The evaluation set includes both tasks seen during training and held-out tasks to test compositional generalization: `ball_chain` combines ramp rolling (seen) with sequential momentum transfer (novel); `double_ramp` chains two ramp segments (novel con-

Figure 1: Example tasks from RIGIDBENCH across three scenes. **Top**: training tasks (`pyramid_topple`, `ramp_to_stack`). **Bottom**: `ramp_launch` (train + eval) and `double_ramp` (eval only). All tasks begin from static initial states with motion triggered solely by gravity.

Task	Physics Tested	# of Objects	Split
`drop_scatter`	Multi-body collision	2–4	Train
`ramp_to_target`	Rolling, collision	2	Train
`drop_to_ramp`	Fall-to-roll transition	2	Train
`ramp_to_stack`	Collision cascade	3–4	Train
`pyramid_topple`	Collision cascade	4–7	Train
`rolling_collision`	Momentum conservation	2	Train
`free_fall`	Gravity, restitution	1	Both
`ramp_launch`	Projectile motion	1	Both
`ball_chain`	Sequential momentum	4	Eval
`double_ramp`	Multi-stage dynamics	2	Eval

Table 1: Physics tasks in RIGIDBENCH. Train = training only; Both = training and evaluation; Eval = evaluation only (tests generalization).

figuration of familiar physics). Ramps are generated procedurally as triangular meshes with randomized PBR wood materials; `ramp_launch` uses parabolic curves for projectile motion.

3.3 EVALUATION PROTOCOL

Given a sample's first frame and text prompt (with an evaluation-only suffix encouraging stationary camera and realistic timing; see Appendix D), each model generates a video. Since models produce videos at different native frame rates (16–30 FPS), we interpolate predictions to align with ground-truth frames at 24 FPS before computing metrics. Our metrics capture complementary aspects of physics prediction: object localization, trajectory accuracy, 3D structure, and visual quality. The entire protocol is automatic and deterministic: given a model's generated video, evaluation requires no human judgment, enabling reproducible comparison across models.

Mask-Based Evaluation. Following PISA (Li et al., 2025), we evaluate object localization using mask-based metrics. We extract object masks frame-by-frame using SAM2 (Ravi et al., 2024),

initialized with the ground-truth segmentation mask from the first frame. We compute three complementary metrics against ground-truth masks: **IoU** (Intersection over Union) measures overall mask overlap, capturing both position and shape errors; **Dist** measures centroid distance normalized by image height, isolating position error independent of shape; **Chamfer Distance (CD)** measures bidirectional nearest-neighbor distance between mask pixels, normalized by image height, capturing shape deformation. Together, these metrics distinguish between models that predict correct positions but wrong shapes versus those with shape-preserving but mislocalized predictions.

Point Tracking Evaluation. While mask metrics capture per-frame accuracy, they do not directly measure temporal trajectory consistency. We sample $K{=}20$ surface points on each object's mask at frame 0 and compute their ground-truth 2D trajectories by unprojecting to 3D using depth maps, transforming through the object's simulated motion, and reprojecting. We track the same query points through generated videos using CoTracker3 (Karaev et al., 2024). **ATE** (Average Trajectory Error) (Sturm et al., 2012) measures mean pixel displacement normalized by image height:

$$\text{ATE} = \frac{1}{K \cdot T \cdot H} \sum_{k=1}^{K} \sum_{t=1}^{T} \|\mathbf{p}_k^{(t)} - \hat{\mathbf{p}}_k^{(t)}\|_2, \tag{1}$$

where $\mathbf{p}_k^{(t)}, \hat{\mathbf{p}}_k^{(t)} \in \mathbb{R}^2$ are the ground-truth and predicted positions of point k at frame t, K is the number of tracked points, T is the number of frames, and H is image height. ATE captures trajectory smoothness and physical plausibility that mask metrics may miss.

Depth Evaluation. We estimate depth maps from generated videos using Video-Depth-Anything (Chen et al., 2025) and compute **SI-MSE** (Scale-Invariant Mean Squared Error) (Eigen et al., 2014) against ground-truth depth, measuring depth accuracy up to global scale.

Perceptual Evaluation. We compute **LPIPS** (Zhang et al., 2018) (deep perceptual similarity) and **SSIM** (Wang et al., 2004) (structural similarity) between generated and ground-truth frames to assess whether physics errors correlate with visual quality degradation.

3.4 Fine-Tuning Approach

PISA (Li et al., 2025) demonstrated that fine-tuning on simulated physics data improves model accuracy. We investigate whether training on RigidBench's photorealistic scenes yields similar benefits. We fine-tune Wan 2.2 TI2V-5B (Team Wan, 2025), a 5B-parameter flow-matching DiT with 3D VAE ($4{\times}16{\times}16$ compression) and UMT5 text encoder. We precompute VAE latents and text embeddings offline to maximize GPU utilization during training.

During training, we replace the noised frame-0 latent with the clean conditioning frame and compute MSE loss only on frames $1{:}T$. We train the DiT in BF16 with gradient checkpointing using AdamW (lr=10^{-4}, weight decay=0.01), batch $4{\times}2$ accumulation, EMA (0.99), for 2K steps on 8 A100s.

4 Experiments

4.1 Experimental Setup

Dataset. We generate 5,000 training samples across eight training tasks and four scenes, and 100 evaluation samples across four evaluation tasks and two scenes, including held-out tasks, scenes, and objects to test generalization.

Models. We evaluate seven video generation models: **Open-source**: Wan 2.2 TI2V-5B (Team Wan, 2025) (49 frames @ 24 FPS) and Cosmos 2.5 (NVIDIA, 2025) (33 frames @ 16 FPS). **Closed-source**: Sora 2 and Sora 2 Pro (OpenAI), Veo 3.1 Fast and Veo 3.1 (Google), and Kling 2.6 Pro (Kuaishou), accessed via API.

4.2 Main Results

Table 2 presents results across all models and metrics. The most striking finding is a stark trade-off between trajectory accuracy and perceptual quality. Sora 2 achieves the best trajectory metrics

	Method	IoU (↑)	Dist (↓)	CD (↓)	ATE (↓)	Depth (↓)	LPIPS (↓)	SSIM (↑)
Closed	Sora 2	0.206	**0.252**	**0.404**	**0.360**	0.367	0.326	0.602
	Sora 2 Pro	0.199	0.285	0.471	0.416	0.352	0.302	0.610
	Veo 3.1 Fast	0.214	0.353	0.611	0.518	0.319	0.130	0.809
	Veo 3.1	0.211	0.356	0.616	0.524	0.326	0.134	0.801
	Kling 2.6 Pro	0.216	0.391	0.689	0.565	0.313	0.138	**0.894**
Open	Wan 2.2	0.234	0.326	0.560	0.494	0.316	**0.104**	0.841
	Wan 2.2 (fine-tuned)	**0.244**	0.373	0.648	0.522	**0.240**	0.213	0.801
	Cosmos-Predict2.5-2B	0.193	0.418	0.736	0.591	0.331	0.241	0.676

Table 2: Results on RIGIDBENCH. **Bold** = best, underlined = second best. IoU = mask overlap; Dist = centroid distance; CD = Chamfer Distance for shape error (all normalized); ATE = trajectory error; Depth = SI-MSE; LPIPS/SSIM = perceptual quality. ↑: higher is better; ↓: lower is better.

Figure 2: Trajectory comparison on ramp_launch. Object centroids tracked across frames with ground-truth (green), Wan 2.2 (blue), Sora 2 Pro (purple), and Veo 3.1 (red). All models track the initial roll correctly, but diverge during projectile motion. Sora models follow closest to the parabolic arc, while others predict slower horizontal motion.

(Dist 0.252, CD 0.404, ATE 0.360) but the worst perceptual scores (LPIPS 0.326, SSIM 0.602). Qualitative inspection reveals the cause: Sora models animate background elements (leaves rustling, surface reflections) that should remain static, penalizing pixel-level metrics despite accurate foreground physics. This produces a 3× gap in LPIPS (0.326 vs 0.104) despite superior object tracking.

Conversely, Kling 2.6 Pro achieves the best SSIM (0.894) by preserving static backgrounds, but shows weaker trajectory accuracy. Wan 2.2 balances both dimensions, achieving the highest IoU among base models (0.234) alongside the best perceptual quality (LPIPS 0.104). Fine-tuning Wan 2.2 on RIGIDBENCH data improves IoU and depth prediction but degrades perceptual quality, suggesting physics-focused training shifts the model away from background preservation.

4.3 PER-TASK ANALYSIS

	Sora 2			Wan 2.2			Wan 2.2 (ft)		
Task	IoU (↑)	ATE (↓)	Depth (↓)	IoU (↑)	ATE (↓)	Depth (↓)	IoU (↑)	ATE (↓)	Depth (↓)
free_fall	0.07	0.40	0.40	0.07	**0.37**	0.35	0.06	0.41	**0.28**
ramp_launch	0.12	**0.71**	0.40	0.13	1.03	0.29	**0.22**	1.13	**0.23**
ball_chain	0.36	**0.10**	0.34	**0.47**	0.14	0.33	0.43	0.11	**0.23**
double_ramp	**0.27**	**0.23**	0.33	**0.27**	0.42	0.31	**0.27**	0.42	**0.23**

Table 3: Per-task breakdown. Sora 2 achieves lowest ATE across tasks. Fine-tuning improves depth across all tasks and IoU on ramp_launch.

Table 3 reveals task-specific strengths and weaknesses. ball_chain achieves the highest IoU (0.36–0.47) across all models, likely because the chain of stationary balls provides strong visual anchors, and collisions produce small displacements rather than large motions. ramp_launch shows the highest trajectory error (ATE 0.8–1.1), indicating that projectile motion after ramp exit remains challenging; models struggle to predict correct parabolic arcs under gravity. free_fall has consistently low IoU (0.06–0.07) despite moderate ATE, suggesting models predict plausible falling trajectories but fail to preserve object appearance through the motion.

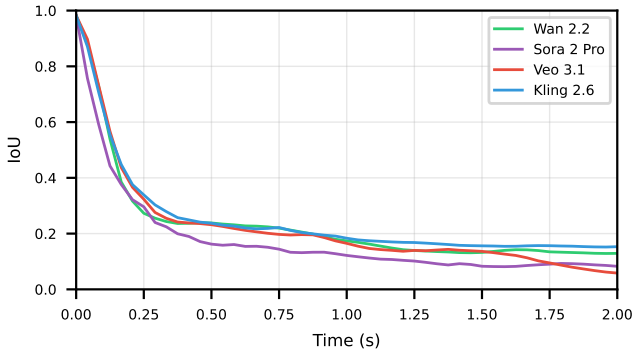

Figure 3: Mask IoU over time. Models start near-perfect (first frame is conditioning) but IoU drops rapidly once motion begins, stabilizing around 0.15 by t=0.5s. All models show similar decay, suggesting error accumulation is task-inherent.

Sora 2 excels on trajectory metrics across all tasks, while Wan 2.2 leads on mask overlap (IoU 0.47 on `ball_chain`). Fine-tuning improves depth prediction across all tasks (SI-MSE drops from 0.29–0.35 to 0.23–0.28) and boosts IoU on `ramp_launch` from 0.13 to 0.22, suggesting physics-focused training helps projectile motion prediction.

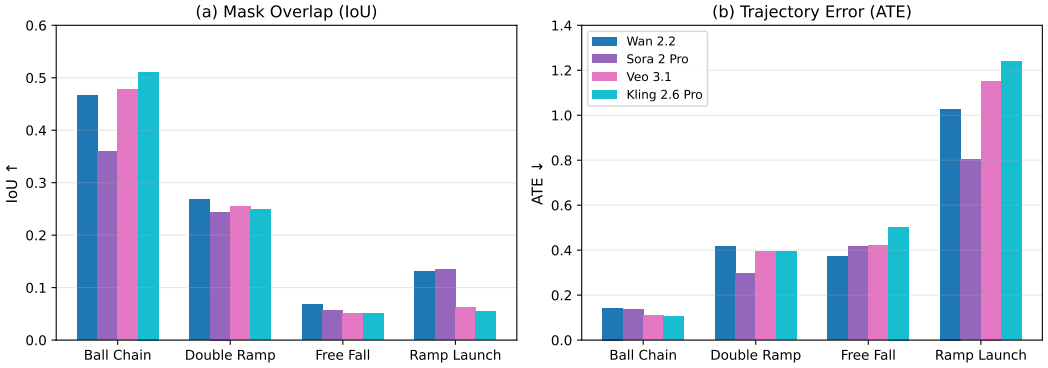

Figure 4: Per-task performance breakdown. (a) Mask overlap (IoU, higher is better): all models achieve reasonable IoU on `ball_chain` but struggle with `free_fall`. (b) Trajectory error (ATE, lower is better): Sora 2 Pro achieves lowest ATE among models shown.

4.4 Qualitative Analysis

Figure 2 visualizes trajectory divergence on `ramp_launch`. Correct physics produces a parabolic arc after the object leaves the ramp, determined by exit velocity and gravity. Sora models follow the ground-truth arc most closely, consistent with their best ATE scores. Wan 2.2 and Veo 3.1 predict slower horizontal motion, causing the object to land short. On `ball_chain`, momentum should transfer sequentially through stationary balls; models often move multiple balls simultaneously or fail to propagate the collision. These failures occur despite the initial frame containing all information needed: ramp geometry determines exit velocity; ball arrangement determines collision sequence. The gap between visible information and model predictions indicates a fundamental limitation in physical reasoning.

5 Discussion

Visual heuristics vs. physical reasoning. Our results suggest models rely on visual correlations rather than physics. Objects "usually fall down" in training videos, so models predict downward motion, but at wrong speeds and bounce angles. All information needed to predict RigidBench

trajectories is visible in frame one, yet models fail dramatically, suggesting the bottleneck is capacity for physical reasoning rather than information availability (Motamed et al., 2025).

Perceptual metrics measure the wrong thing for physics. LPIPS and SSIM compute pixel-level or feature-level similarity between generated and ground-truth frames. When Sora 2 animates background foliage that the ground-truth renders as static, every moved leaf pixel counts against these metrics despite accurate foreground physics. IoU and ATE, by contrast, measure only the object of interest. The 3× LPIPS gap between Sora 2 (0.326) and Wan 2.2 (0.104), despite Sora 2 having the best trajectory accuracy, demonstrates that perceptual metrics conflate background fidelity with physics prediction. Across all models and samples (n=693), ATE and LPIPS show near-zero correlation (r=0.002), confirming these metrics capture orthogonal aspects of generation quality. For robotics applications, trajectory metrics (Dist, ATE) provide the most informative signal.

Model-specific failure patterns. Our evaluation reveals distinct failure modes: Sora 2 produces the most accurate trajectories but animates static backgrounds, while Wan 2.2 and Kling 2.6 Pro preserve backgrounds faithfully but predict less accurate motion. Whether these patterns stem from architectural differences, training data, or post-training optimization remains an open question requiring controlled experiments, but the consistent pattern suggests current models face a physics–fidelity trade-off whose balance varies across systems.

Physics-focused fine-tuning. Our initial fine-tuning experiments show that training on physics-rich synthetic data improves depth prediction across all tasks, both seen and held-out (Table 3). Encouragingly, trajectory error (ATE) improves on the held-out `ball_chain` task (0.14→0.11), suggesting the model learned generalizable momentum transfer rather than memorizing training scenarios. However, we have only explored a single fine-tuning configuration with standard objectives; further investigation into physics-based rewards, dataset scale, and training curricula is needed to fully characterize this approach.

Limitations. RIGIDBENCH focuses on rigid-body dynamics; deformable objects, fluids, and soft-body physics remain untested. Our indoor scenes with static cameras do not capture all physical scenarios. Two-second videos test short-horizon prediction; whether improvements transfer to longer horizons is an open question.

6 CONCLUSION

We introduced RIGIDBENCH, a benchmark and evaluation protocol combining photorealistic rendering with precise physics annotations for evaluating physical understanding in video generation models. Our evaluation reveals three key findings: (1) A trade-off exists between trajectory accuracy and perceptual quality, with Sora 2 leading on physics metrics but trailing on LPIPS/SSIM due to background animation. (2) Task-specific analysis shows models struggle most with projectile motion (`ramp_launch` ATE 1.0) while handling constrained collisions better (`ball_chain` IoU 0.47). (3) Physics-focused fine-tuning consistently improves depth prediction across all tasks, though effects on trajectory metrics are task-dependent and come at the cost of perceptual fidelity.

These findings demonstrate that perceptual metrics alone cannot assess physical understanding, and that targeted benchmarks like RIGIDBENCH are essential for developing reliable video generation models for physical AI. Future work includes extending to deformable objects, investigating reward modeling with physics-specific signals, and controlled studies isolating how model architecture influences physical reasoning.

REFERENCES

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 *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024.

BlenderKit. Blenderkit: Online asset library for blender. `https://www.blenderkit.com/`, 2024.

Sili Chen, Hengkai Guo, Shengnan Zhu, Feihu Zhang, Zilong Huang, Jiashi Feng, and Bingyi Kang. Video depth anything: Consistent depth estimation for super-long videos. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. Highlight.

Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Hickman, Krista Reymann, Thomas B McHugh, and Vincent Vanhoucke. Google scanned objects: A high-quality dataset of 3d scanned household items. In *International Conference on Robotics and Automation (ICRA)*, pp. 2553–2560, 2022.

David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 27, 2014.

Klaus Greff, Francois Belletti, Lucas Beyer, Carl Doersch, Yilun Du, Daniel Duckworth, David J Fleet, Dan Gnanapragasam, Florian Golemo, Charles Herrmann, et al. Kubric: A scalable dataset generator. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3749–3761, 2022.

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 (NeurIPS)*, volume 31, pp. 2451–2463, 2018.

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

Ziqi Huang, Fan Zhang, Xiaojie Shi, Yinan He, Jiashuo Yu, Ziyue Wang, Rui Zhang, Xin Chen, Xintao Li, Yaohui Wang, et al. VBench-2.0: Advancing video generation benchmark suite for intrinsic faithfulness. *arXiv preprint arXiv:2503.21755*, 2025.

Sihui Ji, Xi Chen, Xin Tao, Pengfei Wan, and Hengshuang Zhao. Physmaster: Mastering physical representation for video generation via reinforcement learning. *arXiv preprint arXiv:2510.13809*, 2025.

Nikita Karaev, Iurii Makarov, Jianyuan Wang, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker3: Simpler and better point tracking by pseudo-labelling real videos. *arXiv preprint arXiv:2410.11831*, 2024.

Minh-Quan Le, Yuanzhi Zhu, Vicky Kalogeiton, and Dimitris Samaras. What about gravity in video generation? post-training newton's laws with verifiable rewards. *arXiv preprint arXiv:2512.00425*, 2025.

Chenyu Li, Oscar Michel, Xichen Pan, Sainan Liu, Mike Roberts, and Saining Xie. PISA experiments: Exploring physics post-training for video diffusion models by watching stuff drop. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, 2025.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations (ICLR)*, 2023.

Shaowei Liu, Zhongzheng Ren, Saurabh Gupta, and Shenlong Wang. Physgen: Rigid-body physics-grounded image-to-video generation. In *European Conference on Computer Vision (ECCV)*, 2024.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations (ICLR)*, 2023.

Fanqing Meng, Jiaqi Liao, Xinyu Tan, Quanfeng Lu, Wenqi Shao, Kaipeng Zhang, Yu Cheng, Dianqi Li, and Ping Luo. Towards world simulator: Crafting physical commonsense-based benchmark for video generation. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, volume 267, pp. 43781–43806. PMLR, 2025.

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

NVIDIA. Cosmos world foundation model platform for physical ai. *arXiv preprint arXiv:2501.03575*, 2025.

OpenAI. Video generation models as world simulators. `https://openai.com/research/video-generation-models-as-world-simulators`, 2024. Sora technical report.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. SAM 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024.

Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of rgb-d slam systems. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 573–580, 2012.

Team Wan. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612, 2004.

Chenyu Zhang, Daniil Cherniavskii, Antonios Tragoudaras, Antonios Vozikis, Thijmen Nijdam, Derck W. E. Prinzhorn, Mark Bodracska, Nicu Sebe, Andrii Zadaianchuk, and Efstratios Gavves. Morpheus: Benchmarking physical reasoning of video generative models with real physical experiments. *arXiv preprint arXiv:2504.02918*, 2025.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

A METRIC DEFINITIONS

Let N denote the number of frames. For frame i: let M_i^{gen}, $M_i^{\text{gt}} \in \{0,1\}^{H\times W}$ be generated and ground-truth binary masks, and let $\mathbf{c}_i^{\text{gen}}, \mathbf{c}_i^{\text{gt}} \in \mathbb{R}^2$ be their centroids.

IoU (Intersection over Union):

$$\text{IoU} = \frac{1}{N}\sum_{i=1}^{N}\frac{|M_i^{\text{gen}} \cap M_i^{\text{gt}}|}{|M_i^{\text{gen}} \cup M_i^{\text{gt}}|}. \tag{2}$$

Centroid Distance (normalized):

$$\text{Dist} = \frac{1}{N \cdot H}\sum_{i=1}^{N}\|\mathbf{c}_i^{\text{gen}} - \mathbf{c}_i^{\text{gt}}\|_2. \tag{3}$$

Chamfer Distance (shape error, normalized):

$$\text{CD} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{H}\left(\frac{1}{|P_i|}\sum_{\mathbf{p}\in P_i}\min_{\mathbf{q}\in Q_i}\|\mathbf{p}-\mathbf{q}\|_2 + \frac{1}{|Q_i|}\sum_{\mathbf{q}\in Q_i}\min_{\mathbf{p}\in P_i}\|\mathbf{q}-\mathbf{p}\|_2\right), \tag{4}$$

where P_i, Q_i are pixel coordinates of generated and ground-truth masks at frame i.

SI-MSE (Scale-Invariant Mean Squared Error): Let $\mathcal{V} = \{(i,p) : d_{i,p}^{\text{gt}} > 0 \wedge d_{i,p}^{\text{gen}} > 0\}$ be valid (frame, pixel) pairs. Define log-difference $\delta_{i,p} = \log d_{i,p}^{\text{gen}} - \log d_{i,p}^{\text{gt}}$. Then:

$$\text{SI-MSE} = \frac{1}{|\mathcal{V}|}\sum_{(i,p)\in\mathcal{V}}\delta_{i,p}^2 - \left(\frac{1}{|\mathcal{V}|}\sum_{(i,p)\in\mathcal{V}}\delta_{i,p}\right)^2. \tag{5}$$

This measures depth prediction variance up to a global scale shift, computed over all valid pixels pooled across frames.

B FRAME RATE ALIGNMENT

Video generation models produce videos at different native frame rates: Wan 2.2, Veo 3.1, and Kling 2.6 output 24 FPS (matching ground-truth); Cosmos 2.5 outputs 16 FPS; Sora 2 outputs 30 FPS. To enable fair comparison, we temporally align all predictions to the ground-truth frame rate of 24 FPS.

Direction of Interpolation. We always interpolate *generated predictions* to match ground-truth timestamps, never vice versa. This ensures the ground-truth remains unmodified. Interpolating ground-truth would introduce artifacts into the reference signal and compromise evaluation integrity. The question we ask is: "At physical time t, where does the model predict the object is?"

Time-Based Alignment. We use physical time, not frame indices, for alignment. A 16 FPS video covering 2 seconds contains 32 frames; a 24 FPS video covering 2 seconds contains 48 frames. Frame 16 at 16 FPS corresponds to $t = 1.0$s, which should compare to frame 24 at 24 FPS ($t = 1.0$s), not frame 16. Time-based alignment ensures we compare predictions at the same physical moments.

Given a generated video with N frames at f_{src} FPS and a target of M frames at f_{dst} FPS, we compute aligned frame i as follows. First, we compute the target timestamp $t_i = i/f_{\text{dst}}$ and clip it to the source video duration: $t_i \leftarrow \min(t_i, (N-1)/f_{\text{src}})$. This clipping avoids extrapolation; if a model produces a shorter video than ground-truth, we compare only the overlapping duration. We then interpolate the generated video at time t_i.

Interpolation is data-type dependent. For RGB frames, point tracks, and depth maps, we use linear interpolation between adjacent frames, assuming smooth motion. For binary masks, we use nearest-neighbor selection to preserve the $\{0,1\}$ constraint required for IoU computation.

C Ground Truth Trajectory Computation

We compute ground-truth 2D point trajectories by combining rendered depth maps with 3D object poses from physics simulation:

1. **Sample** K=20 points uniformly on the eroded object mask at frame 0 (erosion prevents edge artifacts).
2. **Unproject to camera space**: For each pixel (u, v) with depth d:

$$\mathbf{P}_{\text{cam}} = \left(\frac{(u - c_x) \cdot d}{f_x}, \frac{(v - c_y) \cdot d}{f_y}, d \right). \tag{6}$$

3. **Transform to Blender world space**: Apply coordinate system conversion (flip Y and Z) and camera extrinsics:

$$\mathbf{P}_{\text{world}} = \mathbf{R}_{\text{cam}} \cdot (\mathbf{P}_{\text{cam}} \odot [1, -1, -1]) + \mathbf{t}_{\text{cam}}, \tag{7}$$

   where $\mathbf{R}_{\text{cam}}, \mathbf{t}_{\text{cam}}$ are camera rotation and translation.
4. **Store in object-local coordinates**: Using object pose at frame 0:

$$\mathbf{P}_{\text{local}} = \mathbf{R}_0^{-1} \cdot (\mathbf{P}_{\text{world}} - \mathbf{t}_0). \tag{8}$$

5. **For each frame** t: Transform local points through object pose $(\mathbf{R}_t, \mathbf{t}_t)$, convert back to camera space, and project to 2D:

$$(u_t, v_t) = \left(f_x \frac{X_t}{Z_t} + c_x, f_y \frac{Y_t}{Z_t} + c_y \right). \tag{9}$$

D Text Prompt Templates

Each task uses a templated prompt with object names filled dynamically. Templates use placeholders: {`rolling`} for rolling objects, {`falling`} for falling objects, {`target`} for collision targets.

Task	Prompt Template
`free_fall`	"{falling} falls."
`drop_scatter`	"{falling} falls onto objects."
`drop_to_ramp`	"{falling} falls onto a ramp and rolls."
`ramp_to_target`	"{rolling} rolls down a ramp toward {target}."
`ramp_to_stack`	"{rolling} rolls down a ramp toward stacked objects."
`pyramid_topple`	"{rolling} rolls down a ramp toward objects."
`rolling_collision`	"Two objects roll toward each other."
`ball_chain`	"{rolling} rolls down a ramp toward a row of objects."
`ramp_launch`	"{rolling} rolls down a curved ramp."
`double_ramp`	"{rolling} rolls down two ramps."

Evaluation suffix (appended during evaluation only): "Locked-off wide shot, stationary camera, no camera movement, no zoom, no pan. Normal speed, no slow motion. Realistic physics, natural motion."

This suffix encourages models to generate videos with a stationary camera and realistic timing, matching the ground-truth rendering conditions. Training samples use only the base templates without this suffix, allowing the model to learn physics from diverse prompt formulations.

E MODEL SPECIFICATIONS

Model	Source	Native FPS	Duration	Resolution
Wan 2.2 TI2V-5B	Open	24	49 frames	1280×704
Cosmos-Predict2.5-2B	Open	16	33 frames	1280×704
Sora 2	API	30	4s	1280×720
Sora 2 Pro	API	30	4s	1280×720
Veo 3.1	API	24	4s	1280×720
Veo 3.1 Fast	API	24	4s	1280×720
Kling 2.6 Pro	API	24	5s	1280×720

F OBJECT ASSETS

All objects are assets from BlenderKit with physically-based rendering (PBR) materials.

Object	Category	Mass (kg)	Restitution	Split
Apple	Fruit	0.2	0.3	Both
Orange	Fruit	0.25	0.4	Both
Watermelon	Fruit	5.0	0.2	Train
Lemon	Fruit	0.1	0.3	Train
Coconut	Fruit	1.5	0.3	Eval
Baseball	Sports	0.15	0.5	Train
Tennis ball	Sports	0.06	0.8	Eval
Soda can	Household	0.35	0.2	Train
Toy duck	Household	0.1	0.4	Train

G SCENE DESCRIPTIONS

We use five photorealistic interior scenes from BlenderKit (BlenderKit, 2024):

Scene	Split	Description
Hallway Indoor Interior	Both	Interior hallway with soft lighting, PBR textures at real-world scale
Modern House Kitchen	Train	Photorealistic contemporary residential kitchen
Modern Loft Apartment	Train	2-story loft with open floor plan including living room, dining area, and kitchen
Butterfly Mural Room	Train	Archviz interior with artistic butterfly mural decoration
Kyra's Living Room	Eval	Modern living room with Cycles materials

H EXTENDED QUALITATIVE RESULTS

I EVALUATION VISUALIZATIONS

Point Tracking. Figure 7 visualizes CoTracker3 point tracking used to compute ATE. We sample 20 surface points on each object at frame 0 and track them through the video. GT tracks (green) follow the simulated object trajectory; predicted tracks (red) show the model's output. Divergence between green and red trails directly reflects trajectory error.

Metric Correlation. Figure 8 visualizes the relationship between trajectory error (ATE) and perceptual quality (LPIPS) across all models and samples. The near-zero correlation (r=0.002) confirms that these metrics capture orthogonal aspects of generation quality: a model can have excellent physics prediction with poor perceptual scores (Sora 2) or vice versa.

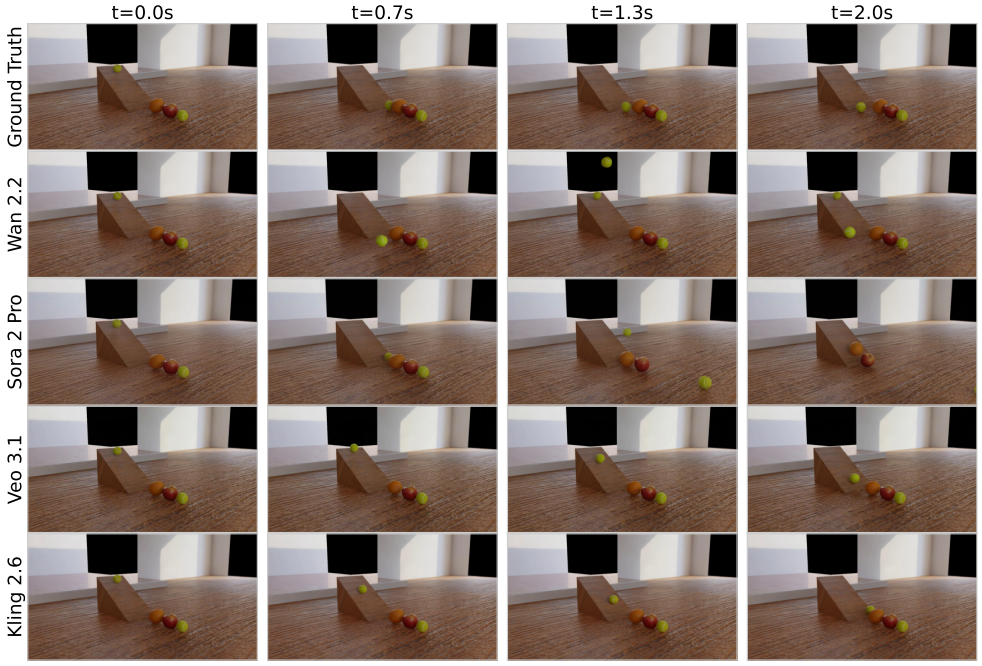

Figure 5: `ball_chain`: Momentum transfer through stationary balls. GT shows correct sequential collision cascade; models vary in their ability to propagate momentum through the chain.

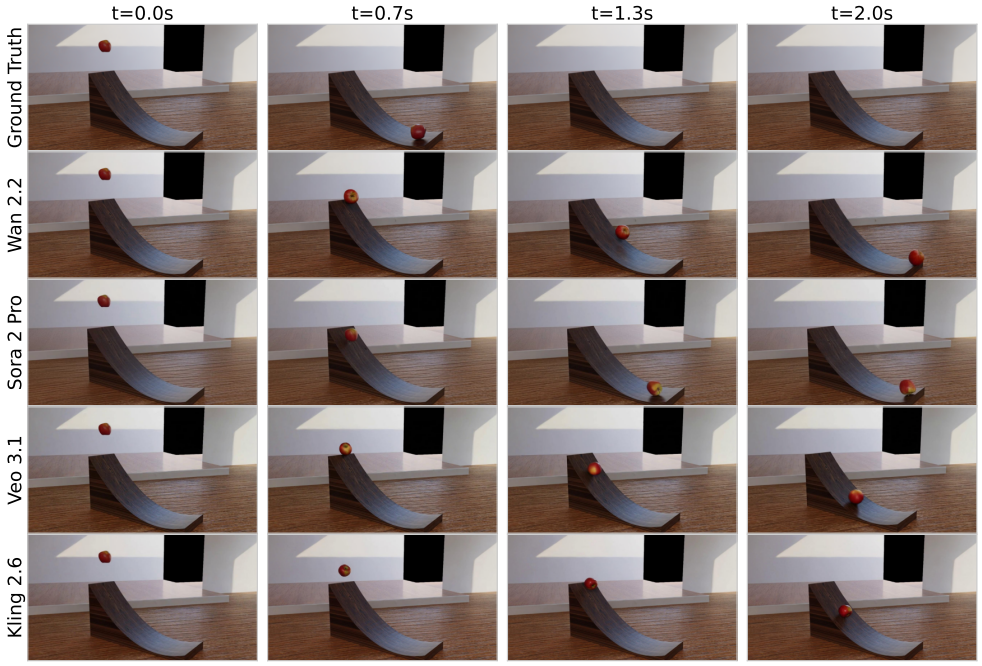

Figure 6: `ramp_launch`: Projectile motion after curved ramp. GT shows correct parabolic trajectory.

Figure 7: Point tracking visualization on `ramp_launch`. GT trajectories (green) show correct parabolic motion after ramp exit. Wan 2.2 predictions (red) diverge during projectile phase, predicting slower horizontal motion.

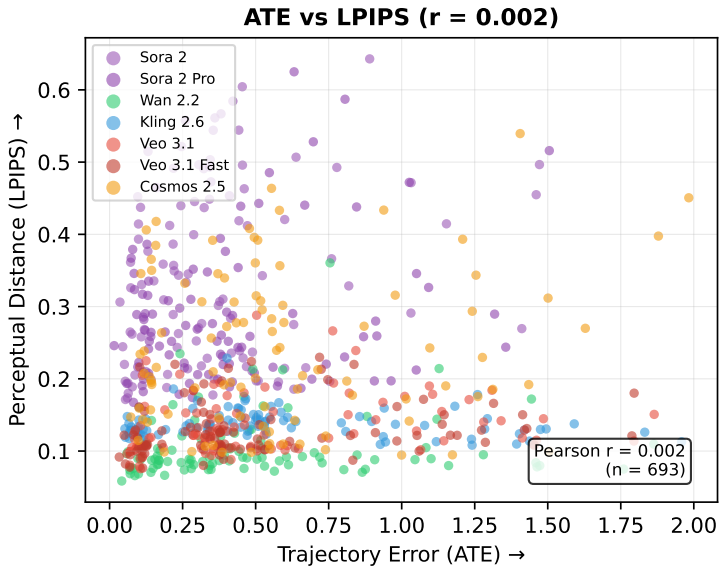

Figure 8: Scatter plot of ATE vs LPIPS across all models and samples (n=693). Pearson correlation r=0.002 indicates trajectory accuracy and perceptual quality are essentially uncorrelated.