Out of Sight, Out of Mind? Evaluating State Evolution in Video World Models

Ziqi Ma*[1,2], Mengzhan Liufu*[1,3], and Georgia Gkioxari[1]

[1] California Institute of Technology, Pasadena CA 91125, USA
[2] ziqima@outlook.com
[3] jhanliufu@gmail.com

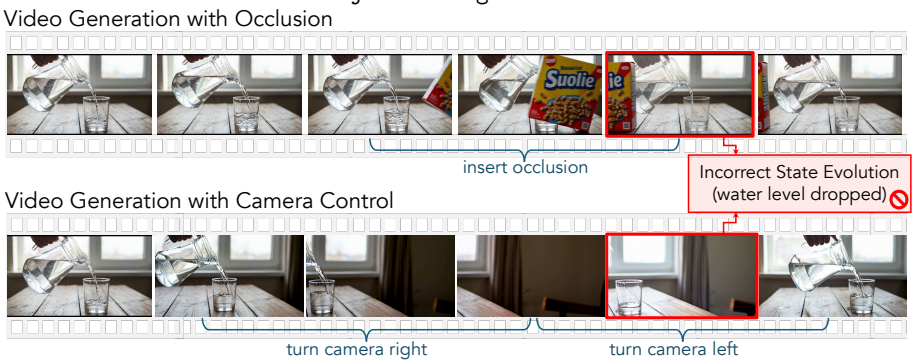

Fig. 1: Today's video world models "simulate" the world by generating pixels. We test whether they can separate state evolution from what's visible by turning the camera away or adding in-scene occlusions. StEvo-Bench evaluates three key capabilities under lookaway/occlusion: whether evolution continues at all, whether it remains physically plausible, and whether the scene stays coherent.

Abstract. Evolutions in the world, such as water pouring or ice melting, happen regardless of being observed. Video world models generate "worlds" via 2D frame observations. Can these generated "worlds" evolve regardless of observation? To probe this question, we design a benchmark to evaluate whether video world models can decouple state evolution from observation. Our benchmark, StEvo-Bench, applies observation control to evolving processes via instructions of occluder insertion, turning off the light, or specifying camera "lookaway" trajectories. By evaluating video models with and without camera control for a diverse set of naturally-occurring evolutions, we expose their limitations in decoupling state evolution from observation. StEvo-Bench proposes an evaluation protocol to automatically detect and disentangle failure modes of video world models across key aspects of natural state evolution. Analysis of StEvo-Bench results provide new insight into potential data and architecture bias of present-day video world models. Benchmark website: https://glab-caltech.github.io/STEVOBench/.

1 Introduction

The world evolves regardless of whether we observe it or not. Consider Fig. 1. When water is being poured into a cup, the amount of water grows, regardless

of whether the cup is observed. Fueled by the rapid progress in video generation, today's world models can generate visual worlds via synthesizing image frames. These image frames, by depicting visual appearance, represent a world with objects and properties, forming the "world state". As objects move or properties change, such as humans moving or fire burning, the "world state" evolves. In the real world, we know that these state evolutions are not dependent on observation – even if they are occluded, or out of view, the state evolves as expected. If video models are truly "world models", they need to possess this property as well: states should correctly evolve, even if not observed.

This is not merely an intellectual investigation. If we want our world models to generate larger worlds and enable longer-horizon interactions, the visible frame observation for any given moment becomes an increasingly small fraction of the generated world. In other words, the majority of the generated world is unobserved. Given this, it becomes critical that the world models can continue evolving the world state, decoupled from observation.

Decoupling evolution from observation has three implicit aspects: 1) The physical plausibility of state evolution: For example, does the water flow correctly based on gravity, and do glass containers not penetrate each other? 2) Consistency of non-evolving aspects: For example, do the cup and jug have the same material and size across the video, without abrupt change or disappearance? 3) The progress of state evolution while not being observed: For example, when the camera turns away and back while the water is still pouring, does water level grow?

Benchmarks define the goal posts for model development, yet existing ones do not support this critical aspect of video world models. Prior benchmarks of world models touch upon 1) and 2). For example, intuitive physics benchmarks like [3,4,20] focus on 1) for physical processes. Consistency benchmarks like [30], or consistency subcategories of [6, 10, 36], focus on 2) under full observation. Memory benchmarks like [34] involve partial observation, but only focus on static settings. Current benchmarks, even when combined, only cover 1) and 2), and thus cannot support the growth of world models towards bigger worlds, longer horizon, which naturally introduces more processes that cannot be observed in their full duration.

We present a benchmark which evaluates all three aspects to verify whether video world models can decouple state evolution from observation. Our benchmark, StEvo-Bench, inserts occlusions or directs the camera to look away during an evolving process. For example, we let the model generate a burning match, and then either insert an occlusion in front of the camera or control the camera to turn away while the match is burning. When the camera turns back or the occlusion is removed, we evaluate whether the match has burned more than previously seen. We focus on video world models, which include image(text)-to-video models and camera-controlled video models. StEvo-Bench contains 225 tasks, encompassing six evolution categories: continuous process, kinematics, relational change, causal change, state transformation, and expected human/animal behavior. To enable automatic evaluation, StEvo-Bench also in-

troduces specialist verifiers built upon VLMs, which automatically evaluate and disentangle critical failure modes.

StEvo-Bench allows us to probe how world models handle evolving processes when observation stops, exposing interesting findings about these models. For video models [8, 11, 21, 28, 33], we find that general-purpose video generation models exhibit evolution stopping or incoherence when observation control (such as occlusion or turning off the light) is applied. For camera-controlled models [7,25,26], we notice a strong bias toward static scenes (no evolution) and increased difficulty in camera control when the models are able to evolve state. We further find that memory-based architectures exacerbate the static-scene bias, and does not help decouple evolution from observation. These findings motivate further investigations about data and architecture bias of current video world models.

We make the following contributions:

- We propose StEvo-Bench, a benchmark to measure whether world models can decouple state evolution from observation.
- We build automatic specialist verifiers to detect and disentangle various axes of failure in state evolutions generated by video world models.
- Quantitative analysis on StEvo-Bench reveals critical findings about current state-of-the-art video world models.

2 Related Works

Video World Models. With the rapid progress in video generation, people started to term video models "world models" due to their capability to generate realistic-looking worlds via video frames. Video "world models" can be categorized into two types: The first type is general-purpose video generation models, such as [8, 11, 21, 28, 33, 38], which can generate worlds based on image or text prompts. The other type is action-conditioned video models [5, 7, 25, 26], which generate videos semi-autoregressively, conditioned on actions. Many works focus on camera control (i.e. navigation) as action [5,7,25,26]. Action space could also be robotic end effector space [1], or time plus camera control [29]. While the definition of "world models" can be broad, in this work, we focus on general-domain world models of the two types above.

World Models Evaluation. Early evaluations of video models focus on video quality and general condition-consistency [10]. However, using video models to generate realistic "worlds", rather than just visually-pleasing videos, poses additional requirements, such as physics correctness, common sense, controllability, and dynamics. General world model benchmarks such as WorldScore [6], WorldModelBench [13] test these aspects in simple settings. Among specialized benchmarks, VideoPhy [3], VideoPhy2 [4] and PAIBench [39] focus on physics realism, World Consistency Score [23], Stable World [12] and MIND [34] focus on consistency and memory. In this work, we evaluate whether world models can successfully evolve processes without full observation, presenting a more rigorous

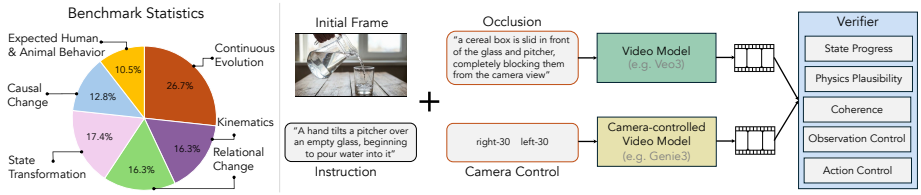

Fig. 2: StEvo-Bench probes whether video-based world models can decouple state evolution from observation. StEvo-Bench, consisting of 225 unique tasks spanning 6 categories, uses (image, text prompt, camera control) tuples to prompt video-based world models to generate evolutions under interrupted observation. Observation interruption is done either using text prompt (such as adding occluders, turning off the light), or using camera control (turning the camera view away). The generated videos go through StEvo-Bench's automatic verifiers for evaluation.

test of physical plausibility and coherence compounded with the challenges of interrupted observation.

State in World Models. Some prior works propose a "stateful" world model that carries an implicit world state representation. Such a formulation, by definition, decouples the state's evolution from observation. Prior works in this direction can be categorized by their different state representations: memory as state, 2.5D/3D as state, and latent state. WorldMem [32], VMem [14], and Long-term Spatial Memory [31] represent the world state as memory. WonderPlay [17], VerseCrafter [37], and Point World [9] use depth point cloud or 3D representations as the "world state". These states, either memory-based or 3D-based, strongly bias towards static scenes. Finally, works like Dino-WM [40], V-JEPA2 [2], Long-context SSM [22], and FloWM [18] use a latent world state. FloWM discusses latent state evolution in simple, synthetic environments like block world. While prior works on stateful world models are generally constrained in simple, synthetic environments, we design a method to probe the "statefulness" in general-purpose video-based world models.

3 Benchmarking State Evolution in Video World Models

StEvo-Bench evaluates whether video world models can continue scene evolution when not the whole process is in view. It consists of 225 unique tasks, spanning 6 different categories of naturally-occurring evolutions, including continuous process, kinematics, relational change, causal change, state transformation, and expected human/animal behavior. The tasks reflect physical events that real-world agents routinely observe: a hand turns on a stove burner and a wet pan begins to heat, a pitcher pours water into a glass, a wall switch is toggled and a lamp turns on, a tire pump inflates a deflated tire, a railroad crossing gate descends as a train approaches, or a hand turns a manual valve and a sprinkler activates. The diversity of tasks and their grounding in everyday physical interactions make StEvo-Bench directly relevant to the challenges faced by embodied agents deployed in the real world.

Category	*Evolution criteria*		*Control criteria*	
	Progress	Physics	Observation	Action
Physics & Common sense [3, 4, 13, 15]	✓	✓	✗	✗
Memory & consistency [23, 30, 34]	✗	✗	✓	✗
Instruction following [6, 36]	✗	✗	✗	✓
StEvo-Bench	✓	✓	✓	✓

Table 1: Evaluating state evolution decoupled from observation requires multiple criteria. While prior benchmarks for video (world) models only touch on subsets of them, StEvo-Bench comprehensively evaluates all aspects to expose and disentangle failure modes of world models when generating evolutions that are not fully observed.

3.1 Task Construction

Tasks in StEvo-Bench require world models to generate a continuous process evolution under interrupted observation (Fig. 2). Each task is specified by an initial image and a text prompt. The initial image shows a main object in its initial state. The text prompt instructs the world model to apply two *controls*: an *action control* that initiates an evolution process of the main object (*e.g.*, hand tips over the first domino), and an *observation control* that interrupts observation of the process mid-evolution. We evaluate the state of the main object before and after observation of it is interrupted.

We implement observation control in a model-specific way. For video generation models, the text prompt instructs the model to apply an in-scene observation interruption for some prolonged period and then remove it to reveal the main object again. That is either a physical occluder placed in front of the camera (*e.g.* a piece of cardboard or curtains) or removing all illumination (*e.g.* turning lights off). For camera-controlled video models, we specify a camera trajectory that moves the main object fully out of frame and then returns to bring it back in view (*e.g.*, "right-30 steps, left-30 steps"). In both cases, the final revealed view indicates whether the state of the main object has evolved correctly during the interval it was out of sight. These strategies are tailored to each model's control interface to ensure reliable observation interruption.

3.2 Evaluation Criteria and Automatic Verifiers

StEvo-Bench evaluates generated videos using a two-stage pipeline. First, we check *control success*: whether the model applied both controls — the observation control successfully hid the scene, and the action control successfully initiated the process. Tasks that fail these control criteria are excluded from further evaluation, since without successful controls the process is either uninitiated or not partially observed. On the passing subset, we evaluate *task success*, which requires three evolution criteria to hold: *state progress* (did the process qualitatively occur during occlusion?), *physical plausibility* (is the evolution physically correct and coherent?) . A video achieves task success only if it first achieves control success and then passes both evolution criteria.

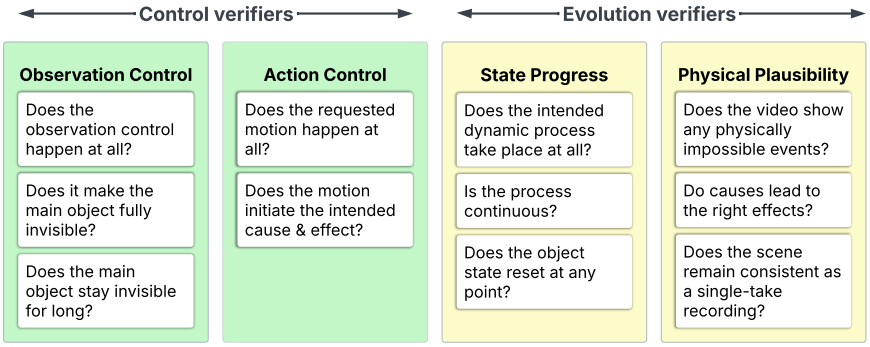

Fig. 3: Automatic verifier pipeline for StEvo-Bench. Four video understanding verifiers independently assess each generated video on one criteria. Observation Control and Action Control jointly determine Control Success; State Progress and Physics Plausibility jointly determine Evolution Success.

Previous benchmarks focus on a subset of these aspects, as shown in Tab. 1. StEvo-Bench evaluates all these aspects to comprehensively quantify state evolution success and disentangle various failure modes.

To evaluate these criteria automatically, we build a specialist verifier that produces a single binary judgment (Fig. 3). All verifiers use `Gemini 3.1 Pro` as the VLM judge. We further validate our verifier with an ensemble of state-of-the-art VLMs to alleviate bias. Decomposing verification into independent specialists serves two purposes: (1) It enables fine-grained diagnosis of why a model fails, since current video models often exhibit multiple entangled failure modes in a single generation; (2) Each verifier poses a single, narrowly scoped yes/no question, which produces more reliable VLM judgments than a monolithic prompt that asks the VLM to simultaneously weigh and disentangle multiple aspects of success. This observation is consistent with findings in prior work on checklist-style prompting [27].

Control verifiers. The two control verifiers check whether the model correctly applied the requested controls:

Observation control verifier: verifies whether the scene was successfully hidden during the evolution, either by an in-scene occluder (cardboard, curtains, or lights off) or by a camera lookaway. The verifier checks that the main object was completely blocked from view for a sustained duration while the process was underway. Partial or transient occlusion, and occlusion that occurs only after the evolution has already completed, are both considered unsuccessful.

Action control verifier: verifies that the requested action event both occurred and took effect. This requires the motion itself to be visible, and it must also correctly initiate the expected physical response. In the case of the requested action being "finger tips over the first domino", the tipping motion must occur,

and it must cause the first domino to tilt over. A completely absent tipping motion, or tipping in mid-ar, are both consider failures by this verifier.

Evolution verifiers. If a generated video passes the control verifiers, then three evolution verifiers check whether the world state evolved correctly in the video:

State progress verifier: verifies whether the intended state change progresses at all during occlusion or lookaway, regardless of physical accuracy or completeness. The key question here is whether the process evolves continuously, as opposed to (a) remaining completely static, (b) discontinued during the unobserved period, or (c) having undone itself during the unobserved period. This verifier first uses the VLM's reasoning to predict, in language, what directional change the evolution should produce, then uses a unanimous-vote ensemble ($n = 3$) of video understanding models to decide whether that change progressed correctly during the unobserved period.

Physical plausibility verifier: verifies that the evolution is physically correct. The verifier prompt covers a three-type taxonomy of violations: Type 1 covers instantaneous single-frame violations (*e.g.*, rigid objects deforming without force, objects teleporting or floating against gravity); Type 2 covers dynamic violations where a visible cause is followed by the wrong effect (*e.g.*, water is poured into a glass yet the level drops); Type 3 covers continuity violations where the background or scene resets in a way impossible for a genuine single-take recording (*e.g.*, an instantaneous wall-color change implying a hidden cut). Events that occur during an occlusion period are excluded, since the hidden cause cannot be assessed. We prompt the verifier to flag only active violations that a casual viewer would find clearly jarring, not the mere absence of the intended change, to avoid conflating a missing evolution with an incorrect one. The same majority-vote ensemble approach was used to suppress hallucinations.

4 Do Video World Models Evolve State Successfully?

StEvo-Bench allows us to probe whether video models can successfully evolve states when part of the dynamic process is occluded or out-of-view.

4.1 Model Candidates

We evaluate both video models which generate videos conditioned on image and text, as well as camera-controlled video models which additionally takes in a specified camera trajectory.

Video Models. We evaluate open- and closed-source general-purpose video generative models, including Veo 3 [8], Sora 2 Pro [21], WAN 2.2 [28], CogVideoX 1.5 [33] and HunyuanVideo 1.5 [11].

Camera-controlled Video Models. We evaluate state-of-the-art camera- controlled video models, including Genie 3 [7], HunyuanWorld 1.0 [25], LingBot-World [26], GEN3C [24], VMem [14], and AETHER [42]. Among these models,

	Model	Success	Progress	Physics
Video models	Veo 3	**9.0**	**17.6**	52.5
	Sora 2 Pro	8.1	13.1	**63.3**
	WAN 2.2	0.9	7.7	57.9
	HY-Video 1.5	0.9	4.1	25.8
	CogVideoX 1.5	0.5	1.4	45.5
Camera-controlled	Genie 3	0.0	2.9	42.6
	HY-WorldPlay	0.0	0.0	50.6
	Lingbot	0.0	**3.4**	**51.8**
	GEN3C	0.0	0.0	50.6

Table 2: Evaluation metrics for video models (top) and camera-controlled models (bottom). Both video models and camera-controlled video models struggle to evolve state correctly when observation control is applied. Metrics are in percentage (%).

Hunyuan-WorldPlay, Lingbot are open-source general camera-conditioned video models. GEN3C incorporates 3D priors, VMem incorporates a memory module, and AETHER pretrains on 4D reconstruction objective. Since VMem always creates severe visual artifacts during camera pan, and AETHER "forgets" the object before camera turn due to its short, 41-frame context, we only evaluate them qualitatively.

4.2 Findings

Can video world models decouple state evolution from observation? Tab. 2 quantitatively evaluates whether video models can successfully evolve states on StEvo-Bench tasks, as well as their subgoal performance.

> Finding 1: Both video models and camera-controlled video models struggle to decouple state evolution from observation.

As seen in Tab. 2, both video models and camera-controlled video models show less than 10% success rate in evolving state under interrupted observation. While closed-source video models such as Veo 3 and Sora 2 Pro can occasionally evolve the process (state progress rate being 17% and 13% respectively), they often fail to maintain coherence and physical plausibility. Camera-controlled models almost never evolve states, with less than 5% state progress across closed- and open-source models. We present findings about these models' specific failure modes in the analysis below.

How do video models fail under observation control? StEvo-Bench exposes two key failure modes of video models when observation control is applied to these models: evolution stopping and incoherence. Fig. 4 highlights both failure modes.

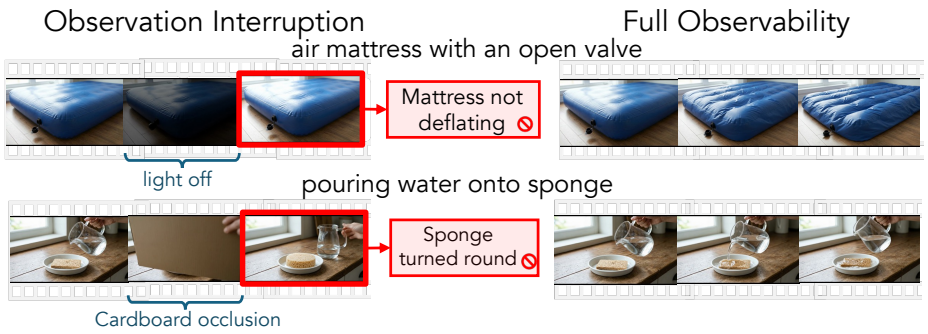

Fig. 4: Representative failure modes of video models when observation of the evolution process is interrupted. The model is capable of simulating the process correctly when the scene is fully visible. However, when observation of the process is temporarily interrupted, the model either stops evolving state, as seen in the mattress deflation example, or fails to preserve object coherence, as seen in the sponge example. These videos are generated by Veo 3.

Finding 2: When observation control (*e.g.*, occlusions or "turn light off") is introduced into an evolving process, video models often stall the evolution or exhibit incoherence.

Evolution stopping. A common failure mode is stalled state evolution. For instance, in Fig. 4, an air mattress with an open valve should continue deflating; yet when we instruct the model to "turn off the light", the mattress stops deflating. Our state progress verifier detects this issue. As shown in Tab. 2, all video models have low state progress rates, with open-source models particularly prone to this failure mode.

Incoherence. A second common failure mode is loss of object coherence once an occluder disappears – for example, the rectangular sponge becoming round in Fig. 4. The coherence verifier captures these errors, with state-of-the-art closed-source models achieving only about a 60% success rate.

To further understand whether such failures are indeed caused by the observation control, we perform additional experiments where the observation control is not inserted, *i.e.*, the process evolves while fully observed. As seen in Fig. 4, both these processes succeed when there is no occluder or "light off" intervention. Tab. 3 provides further quantitative evidence that when observation control is applied, both the state progress rate (which measures whether the state progressed during the occlusion) and task success

	Progress	Success
Observ. Full	84.6	46.2
Observ. Control	17.4	12.4

Table 3: The effect of observation control: state progress rate and task success rate drop significantly when observation control is applied. Metrics are averaged over Veo 3 and Sora 2 Pro.

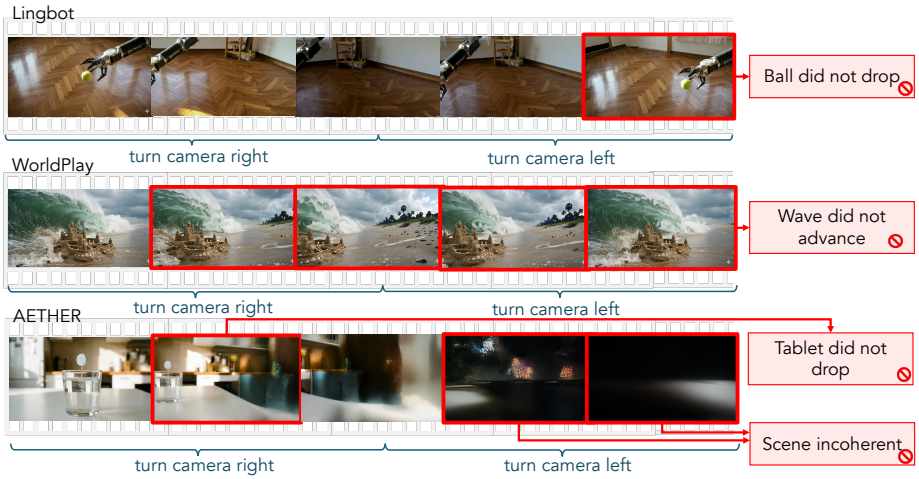

Fig. 5: Open-source camera-controlled video models assume the scene to be static as the camera turns, and fail to correctly evolve state, such as the ball dropping, wave advancing, or tablet dropping.

rate (which measures whether the full evolution is correct and coherent) are significantly lower.

The success of full-observability generation suggests that models possesses internal knowledge of how the process should evolve. However, observation control renders these models incapable of generating these processes correctly. While one might argue that this problem is solvable by incorporating large-scale occlusion-interrupted dynamics videos into training, we believe such failures expose fundamental issues on how video models process context. Since the occluding frames do not provide any useful information on the state evolution, naive all-to-all bidirectional attention might not be efficient in such cases. We hope these failure cases can inspire and motivate new architecture designs for video world models.

How do camera-controlled video models fail under observation control?

Finding 3: Camera-controlled video models tend to "freeze" the scene and fail to continue the dynamic process.

Camera-controlled models, when instructed to "look away" from a process, largely fail to evolve the process, and tend to keep the scene static as the camera moves. We observe similar behavior across models, especially open-source models like Hunyuan-WorldPlay, Lingbot, and GEN3C. This is also seen in the close-to-zero state progress rate for all camera-controlled models in Tab. 2. Fig. 5 shows representative failures of such models, sourced from Hunyuan-WorldPlay and Lingbot. GEN3C, which leverages 3D cache to enhance camera control, biases strongly toward static scene generation. AETHER, a camera-controlled video which mixes in 4D reconstruction objective during training, is also unable

to evolve state, and fails to remember the object due to the its short context window.

Finding 4: When camera-controlled video models successfully evolve the state, camera control becomes difficult.

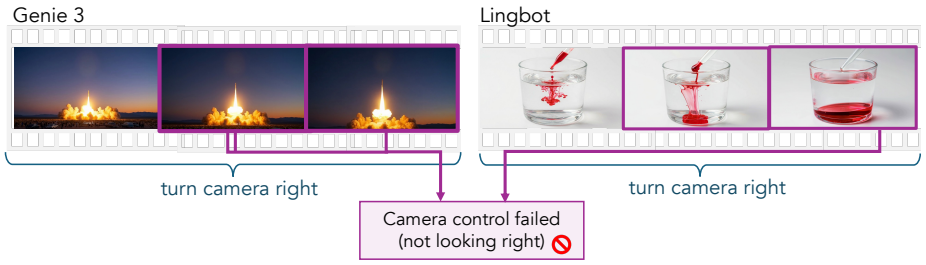

Fig. 6: When camera-controlled models can successfully evolve states, the model tends to ignore camera control while the evolution is happening. Such behavior is seen both in Genie 3 and Lingbot.

The previous finding states that camera-controlled models frequently generate static worlds. In the less frequent event where they do generate dynamics, we observe that camera control becomes difficult. When state evolution is being generated, the model largely ignores the camera control and keeps the camera static. Fig. 6 shows two representative examples. In both cases, the user continuously directs the camera to turn right. In the first example, while the rocket launches, the camera does not turn right but slightly turns up to follow the rocket. In the second example, the camera stays completely static despite the camera trajectory input. While previous findings focus on whether observation control makes evolution difficult, these failure modes show the converse is also true: evolution makes camera control difficult.

These behaviors show another aspect of the evolution-observation coupling: the inability for camera pan and state evolution to happen concurrently suggests a strong coupling between the model's capability to evolve state and generating full pixel descriptions of the process from a largely unchanged viewpoint.

Do memory modules help decouple state evolution from observation? Previous findings have shown that video world models have strong coupling between evolution and observation. We further study whether memory modules, which enable the model to keep track of a "memory state" separate from pixel-level generations, can help decouple state evolution from observation.

While many memory-based architectures [32, 35] are constrained to specialized domains such as Minecraft or RealEstate10k [41], we evaluate VMem [14], which is the most general-domain memory-based model, trained on indoor and outdoor scenes.

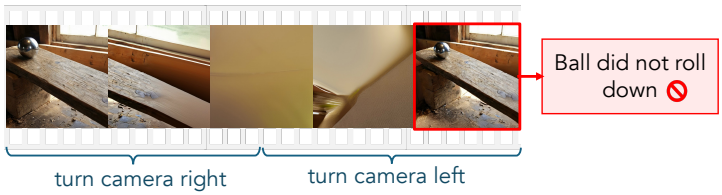

Fig. 7: Qualitative example of VMem. Due to the memory design, VMem can perfectly recall the initial frame, but fails to evolve state, and suffers from scene incoherence.

Finding 5: Memory-based video world models reinforce the bias toward static scene.

As shown in Fig. 7, VMem shows a strong bias toward remembering objects exactly "as is". VMem, although producing artifacts during the camera lookaway process, is able to recall the initial frame quite accurately due to the memory design. We argue that while memory enables the model to store a "state" that is decoupled from observation, the architecture design encourages memorization of object appearance, rather than enabling better modeling of state evolution. StEvo-Bench highlights the need for new architecture designs to enable not only memorization, but also evolution of the world that is not always fully in-view.

5 Analyses and Discussions

5.1 Control success rate analysis

One predicate of our evaluation is that both observation and action control are applied correctly, *i.e.*, the process is correctly initiated, and occlusion or lookaway happen. We verify that current models can support such control by evaluating success rate under each axis. Tab. 4 shows that video models and camera-controlled video models can correctly apply both observation and action control, validating the evaluation protocol of StEvo-Bench.

	Model	Observ.	Action
Video models	Veo 3	**86.0**	**89.6**
	Sora 2 Pro	69.4	80.4
	WAN 2.2	46.2	76.5
	HY-Video 1.5	31.2	81.0
	CogVideoX 1.5	17.7	68.6
Camera-controlled	Genie 3	84.7	**78.6**
	HY-WorldPlay	50.6	60.2
	Lingbot	48.2	75.3
	GEN3C	**90.6**	57.6

Table 4: Success rate of observation and action control. Models are largely able to follow control of StEvo-Bench tasks.

5.2 Evaluation of verifiers against human annotators

The automatic verifiers are only useful if they reliably capture the intended evaluation criteria. Since visually judging diverse videos is a subjective task, we validate the verifiers by comparing them to human annotations. If a verifier

		Evolution criteria				*Control criteria*		
	Metric	Progress	Physics	Coherence	Success	Observation	Action	Success
V-H	Acc	**0.795**	0.700	0.860	**0.810**	**0.878**	**0.903**	**0.849**
	ROC-AUC	**0.644**	**0.672**	**0.864**	**0.713**	**0.800**	**0.918**	**0.785**
	MRA	**0.829**	0.743	0.905	**0.858**	**0.891**	**0.949**	**0.857**
H-H	Acc	0.747	**0.722**	**0.911**	0.737	0.659	0.885	0.692
	ROC-AUC	0.623	0.645	0.629	0.602	0.668	0.782	0.708
	MRA	0.807	**0.757**	**0.919**	0.820	0.825	0.868	0.824

Table 5: Agreement between the automatic verifiers and the human annotators, benchmarked against inter-human agreement across three metrics. **Bold** indicates the higher value per column. The **V-H** row shows agreement between the verifiers and human annotators, and the **H-H** shows agreement within the human annotators.

agrees with human annotators at least as well as annotators agree with each other, it is performing at the ceiling of what any automatic system could reasonably achieve. We recruited three crowdsourced annotators on Upwork and collect binary labels (yes/no) for all five evaluation criteria. We do this on a randomly selected subset ($n = 180$) of the generated videos spanning all models.

We measure agreement using three complementary metrics. `Accuracy` is the fraction of tasks where two raters assign the same binary label, averaged over all rater pairs. `ROC-AUC` addresses potential label class imbalance. Given two raters, one is treated as ground truth and the other as predictor. Since predictions are binary (not soft scores), the `ROC` curve has only one interior point, so AUC reduces to $(\texttt{TPR} + \texttt{TNR})/2$, where `TPR` is the fraction of ground-truth positives the predictor labels positive, and TNR is the fraction of ground-truth negatives the predictor labels negative. For verifier-vs-human comparisons, the human is treated as ground truth. For human-vs-human comparisons, both directions are computed for each pair and averaged symmetrically, since neither annotator is privileged. `Model ranking agreement (MRA)` sidesteps absolute calibration entirely: for every pair of models (A, B) and every shared task, it asks whether two raters agree on which model performed better. Agreement is scored as 1.0 for a full match, 0.5 when one rater calls a tie and the other does not, and 0.0 for a reversal; scores are averaged over all model pairs and tasks.

All three metrics produce values between 0 and 1, with 1 being perfect agreement. The agreement results are shown in Tab. 5. All three metrics converge on consistent conclusions. Human agreement is lowest on physical plausibility and observation control. This reflects genuine ambiguity in the definitions of these conflated, inter-related failure modes. Deciding whether a physical effect constitutes a jarring violation (rather than a slow or imperfect evolution) requires nuanced domain knowledge, and different annotators apply different thresholds. Similarly, judging whether an occlusion was "complete enough" is subjective.

Nevertheless, the verifier matches or exceeds inter-human agreement on nearly all criteria. The verifier-vs-human `MRA` meets or exceeds inter-human `MRA` on every single criteria. With `Accuracy`, the verifier outperforms inter-human agreement

when evaluating state progress, observation control and action control. The only criteria where the verifier lags slightly is physical plausibility, which is also the hardest for humans. Together, these results confirm that our automatic verifiers are reliable proxies for well-educated human judgment: they agree with individual human annotators as well as annotators agree with each other, and they recover the same model ordering that a human panel would produce.

5.3 Evaluation of different VLM verifiers

We chose Gemini 3.1 Pro as our verifier due to its state-of-the-art video understanding capability. To result consistency with respect to the choice of VLM verifier, we repeated evaluation using an ensemble of state-of-the-art VLMs: Gemini 3.1 Pro, GPT 5.4 and Claude Opus 4.7 (majority vote). Evaluated on 100 videos, we see high agreement (84%-94%) between the cross-VLM ensemble result and our current setup across various evaluation axes. This experiment and the Gemini-human agreement experiment (previous section) validate the choice of Gemini 3.1 as a judge.

Physics	Observation	Action	Progress
84.0%	89.0%	90.0%	94.0%

Table 6: Ensemble agreement scores across evaluation metrics.

5.4 Discussions on the tradeoff between camera control and dynamics

Our findings show that on the one hand, models with relatively good camera control loses dynamics (Finding 2). On the other hand, if dynamics is successfully generated, camera control becomes difficult (Finding 3). These findings suggest a potential tradeoff between dynamics and camera control in current models. We hypothesize that this might be attributed to the training data. Camera-controlled video world models usually train on a combination of general video data, rendered 3D data, and gaming data:

- In order to obtain diverse camera trajectories, camera-controlled models train on rendered videos of static scenes. For example, WorldPlay trains on renderings of reconstructed 3D Gaussian Splats. These videos move the camera but do not contain dynamics, which might reinforce the static bias, and furthermore associate complex camera movement with static scenes.
- The general video datasets that models train on, *e.g.*, Sekai [16] and DL3DV [19], encourage the exclusion of dynamic objects to improve 3D consistency for tasks like novel view synthesis.
- The gaming data, which contain (action, video) pairs, usually don't contain rich, natural physical evolutions that occur in the real world.

Enabling simultanous state evolution and camera control in world models might require different synthetic data strategies to mitigate data bias or post-training techniques to steer models toward correct real-world dynamics.

6 Conclusions

We present StEvo-Bench, a benchmark that evaluates whether video world models can evolve state correctly when observation is interrupted. We probe this by applying observation control, either as occlusions or illumination dimming for video models, or as "lookaway" camera commands for camera-controlled video models. StEvo-Bench exposes that video world models struggle to evolve state when the observation control is applied. StEvo-Bench's verifiers disentangle various failure modes to deepen our understanding of the limitations of today's video models. Evaluation on StEvo-Bench yields interesting findings, such as video models' evolution stopping and incoherence failures, camera-controlled models' strong bias towards static scenes, and the increased difficulty in camera control when a dynamic evolution is present. These findings lead us to reflect on the training data and architecture design of video world models, such as training data bias the efficacy of memory-based architectures. StEvo-Bench makes the first step toward evaluating the non-pixel-based "world evolution" aspect of world models. We hope StEvo-Bench inspires new techniques and architectures that enable video world models to truly model processes in the world, and not just generate coherent pixels.

References

1. Ali, A., Bai, J., Bala, M., Balaji, Y., Blakeman, A., Cai, T., Cao, J., Cao, T., Cha, E., Chao, Y.W., et al.: World simulation with video foundation models for physical ai. arXiv preprint arXiv:2511.00062 (2025) 3
2. Assran, M., Bardes, A., Fan, D., Garrido, Q., Howes, R., Muckley, M., Rizvi, A., Roberts, C., Sinha, K., Zholus, A., et al.: V-jepa 2: Self-supervised video models enable understanding, prediction and planning. arXiv preprint arXiv:2506.09985 (2025) 4
3. Bansal, H., Lin, Z., Xie, T., Zong, Z., Yarom, M., Bitton, Y., Jiang, C., Sun, Y., Chang, K.W., Grover, A.: Videophy: Evaluating physical commonsense for video generation. In: International Conference on Learning Representations (2025) 2, 3, 5
4. Bansal, H., Peng, C., Bitton, Y., Goldenberg, R., Grover, A., Chang, K.W.: Videophy-2: A challenging action-centric physical commonsense evaluation in video generation. arXiv preprint arXiv:2503.06800 (2025) 2, 3, 5
5. Bar, A., Zhou, G., Tran, D., Darrell, T., LeCun, Y.: Navigation world models. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 15791–15801 (2025) 3
6. Duan, H., Yu, H.X., Chen, S., Fei-Fei, L., Wu, J.: Worldscore: A unified evaluation benchmark for world generation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 27713–27724 (October 2025) 2, 3, 5
7. Google DeepMind: Genie 3: A new frontier for world models. https://deepmind.google/blog/genie-3-a-new-frontier-for-world-models/ (2025), accessed: 2026-03-03 3, 7
8. Google DeepMind: Veo: a text-to-video generation system. https://storage.googleapis.com/deepmind-media/veo/Veo-3-Tech-Report.pdf (2025), accessed: 2026-03-03 3, 7

9. Huang, W., Chao, Y.W., Mousavian, A., Liu, M.Y., Fox, D., Mo, K., Fei-Fei, L.: Pointworld: Scaling 3d world models for in-the-wild robotic manipulation. arXiv preprint arXiv:2601.03782 (2026) 4
10. Huang, Z., He, Y., Yu, J., Zhang, F., Si, C., Jiang, Y., Zhang, Y., Wu, T., Jin, Q., Chanpaisit, N., Wang, Y., Chen, X., Wang, L., Lin, D., Qiao, Y., Liu, Z.: VBench: Comprehensive benchmark suite for video generative models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2024) 2, 3
11. Kong, W., Tian, Q., Zhang, Z., Min, R., Dai, Z., Zhou, J., Xiong, J., Li, X., Wu, B., Zhang, J., et al.: Hunyuanvideo: A systematic framework for large video generative models. arXiv preprint arXiv:2412.03603 (2024) 3, 7
12. Kwon, S., Kim, J.Y., Go, H., Baek, K.: Toward stable world models: Measuring and addressing world instability in generative environments. Pattern Recognition p. 113351 (2026) 3
13. Li, D., Fang, Y., Chen, Y., Yang, S., Cao, S., Wong, J., Luo, M., Wang, X., Yin, H., Gonzalez, J.E., Stoica, I., Han, S., Lu, Y.: Worldmodelbench: Judging video generation models as world models. In: Advances in Neural Information Processing Systems (2025) 3, 5
14. Li, R., Torr, P., Vedaldi, A., Jakab, T.: Vmem: Consistent interactive video scene generation with surfel-indexed view memory. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 25690–25699 (2025) 4, 7, 11
15. Li, X., Xia, Z., Lu, W., Hao, C., Chen, Y.: Smallworlds: Assessing dynamics understanding of world models in isolated environments. arXiv preprint arXiv:2511.23465 (2025) 5
16. Li, Z., Li, C., Mao, X., Lin, S., Li, M., Zhao, S., Xu, Z., Li, X., Feng, Y., Sun, J., Li, Z., Zhang, F., Ai, J., Wang, Z., Wu, Y., He, T., Pang, J., Qiao, Y., Jia, Y., Zhang, K.: Sekai: A video dataset towards world exploration (2025) 14
17. Li, Z., Yu, H.X., Liu, W., Yang, Y., Herrmann, C., Wetzstein, G., Wu, J.: Wonderplay: Dynamic 3d scene generation from a single image and actions. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9080–9090 (2025) 4
18. Lillemark, H.J., Huang, B., Zhan, F., Du, Y., Keller, T.A.: Flow equivariant world models: Memory for partially observed dynamic environments. arXiv preprint arXiv:2601.01075 (2026) 4
19. Ling, L., Sheng, Y., Tu, Z., Zhao, W., Xin, C., Wan, K., Yu, L., Guo, Q., Yu, Z., Lu, Y., et al.: Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 22160–22169 (2024) 14
20. Mak, C.W., Zhu, G., Zhang, B., Li, H., Chi, X., Zhang, K., Wu, Y., He, Y., Fan, C.K., Lu, W., Ge, K., Fang, X., He, H., Lu, K., Xu, T., Zhang, L., Ni, Y., Li, Y., Zhang, S.: Physicsmind: Sim and real mechanics benchmarking for physical reasoning and prediction in foundational vlms and world models (2026), https://arxiv.org/abs/2601.16007 2
21. OpenAI: Sora 2. https://openai.com/index/sora-2/ (2025), accessed: 2026-03-03 3, 7
22. Po, R., Nitzan, Y., Zhang, R., Chen, B., Dao, T., Shechtman, E., Wetzstein, G., Huang, X.: Long-context state-space video world models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 8733–8744 (2025) 4

23. Rakheja, A., Ashdhir, A., Bhattacharjee, A., Sharma, V.: World consistency score: A unified metric for video generation quality. arXiv preprint arXiv:2508.00144 (2025) 3, 5
24. Ren, X., Shen, T., Huang, J., Ling, H., Lu, Y., Nimier-David, M., Müller, T., Keller, A., Fidler, S., Gao, J.: Gen3c: 3d-informed world-consistent video generation with precise camera control. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6121–6132 (2025) 7
25. Sun, W., Zhang, H., Wang, H., Wu, J., Wang, Z., Wang, Z., Wang, Y., Zhang, J., Wang, T., Guo, C.: Worldplay: Towards long-term geometric consistency for real-time interactive world modeling. arXiv preprint arXiv:2512.14614 (2025) 3, 7
26. Team, R., Gao, Z., Wang, Q., Zeng, Y., Zhu, J., Cheng, K.L., Li, Y., Wang, H., Xu, Y., Ma, S., Chen, Y., Liu, J., Cheng, Y., Yao, Y., Zhu, J., Meng, Y., Zheng, K., Bai, Q., Chen, J., Shen, Z., Yu, Y., Zhu, X., Shen, Y., Ouyang, H.: Advancing open-source world models. arXiv preprint arXiv:2601.20540 (2026) 3, 7
27. Viswanathan, V., Sun, Y., Ma, S., Kong, X., Cao, M., Neubig, G., Wu, T.: Checklists are better than reward models for aligning language models. In: NeurIPS (2025), https://arxiv.org/abs/2507.18624 6
28. Wan, T., Wang, A., Ai, B., Wen, B., Mao, C., Xie, C.W., Chen, D., Yu, F., Zhao, H., Yang, J., Zeng, J., Wang, J., Zhang, J., Zhou, J., Wang, J., Chen, J., Zhu, K., Zhao, K., Yan, K., Huang, L., Feng, M., Zhang, N., Li, P., Wu, P., Chu, R., Feng, R., Zhang, S., Sun, S., Fang, T., Wang, T., Gui, T., Weng, T., Shen, T., Lin, W., Wang, W., Wang, W., Zhou, W., Wang, W., Shen, W., Yu, W., Shi, X., Huang, X., Xu, X., Kou, Y., Lv, Y., Li, Y., Liu, Y., Wang, Y., Zhang, Y., Huang, Y., Li, Y., Wu, Y., Liu, Y., Pan, Y., Zheng, Y., Hong, Y., Shi, Y., Feng, Y., Jiang, Z., Han, Z., Wu, Z.F., Liu, Z.: Wan: Open and advanced large-scale video generative models. arXiv preprint arXiv:2503.20314 (2025) 3, 7
29. Wang, Y., Zhang, Q., Cai, S., Wu, T., Ackermann, J., Kuang, Z., Zheng, Y., Rajič, F., Tang, S., Wetzstein, G.: Bullettime: Decoupled control of time and camera pose for video generation. arXiv preprint arXiv:2512.05076 (2025) 3
30. Wu, M., Mishra, A., Dey, S., Xing, S., Ravipati, N., Wu, H., Li, B., Tu, Z.: Consid-gen: View-consistent and identity-preserving image-to-video generation (2026), https://arxiv.org/abs/2602.10113 2, 5
31. Wu, T., Yang, S., Po, R., Xu, Y., Liu, Z., Lin, D., Wetzstein, G.: Video world models with long-term spatial memory. In: Advances in Neural Information Processing Systems (2025) 4
32. Xiao, Z., Yushi, L., Zhou, Y., Ouyang, W., Yang, S., Zeng, Y., Pan, X.: Worldmem: Long-term consistent world simulation with memory. In: The Thirty-ninth Annual Conference on Neural Information Processing Systems (2025) 4, 11
33. Yang, Z., Teng, J., Zheng, W., Ding, M., Huang, S., Xu, J., Yang, Y., Hong, W., Zhang, X., Feng, G., et al.: Cogvideox: Text-to-video diffusion models with an expert transformer. In: International Conference on Learning Representations (2025) 3, 7
34. Ye, Y., Lu, X., Jiang, Y., Gu, Y., Zhao, R., Liang, Q., Pan, J., Zhang, F., Wu, W., Wang, A.J.: Mind: Benchmarking memory consistency and action control in world models (2026), https://arxiv.org/abs/2602.08025 2, 3, 5
35. Zhao, J., Wei, F., Liu, Z., Zhang, H., Xu, C., Lu, Y.: Spatia: Video generation with updatable spatial memory. arXiv preprint arXiv:2512.15716 (2025) 11
36. Zheng, D., Huang, Z., Liu, H., Zou, K., He, Y., Zhang, F., Zhang, Y., He, J., Zheng, W.S., Qiao, Y., Liu, Z.: VBench-2.0: Advancing video generation benchmark suite for intrinsic faithfulness. arXiv preprint arXiv:2503.21755 (2025) 2, 5

37. Zheng, S., Yin, M., Hu, W., Li, X., Shan, Y., Fu, Y.: Versecrafter: Dynamic realistic video world model with 4d geometric control. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2026) 4
38. Zheng, Z., Peng, X., Lou, Y., Shen, C., Young, T., Guo, X., Wang, B., Xu, H., Liu, H., Jiang, M., Li, W., Wang, Y., Ye, A., Ren, G., Ma, Q., Liang, W., Lian, X., Wu, X., Zhong, Y., Li, Z., Gong, C., Lei, G., Cheng, L., Zhang, L., Li, M., Zhang, R., Hu, S., Huang, S., Wang, X., Zhao, Y., Wang, Y., Wei, Z., You, Y.: Open-sora 2.0: Training a commercial-level video generation model in $200k (2026), https://arxiv.org/abs/2503.09642 3
39. Zhou, F., Huang, J., Li, J., Ramanan, D., Shi, H.: Pai-bench: A comprehensive benchmark for physical ai. arXiv preprint arXiv:2512.01989 (2025) 3
40. Zhou, G., Pan, H., LeCun, Y., Pinto, L.: Dino-wm: World models on pre-trained visual features enable zero-shot planning. In: Forty-second International Conference on Machine Learning (2025) 4
41. Zhou, T., Tucker, R., Flynn, J., Fyffe, G., Snavely, N.: Stereo magnification: Learning view synthesis using multiplane images. ACM Trans. Graph. (Proc. SIGGRAPH) **37** (2018), https://arxiv.org/abs/1805.09817 11
42. Zhu, H., Wang, Y., Zhou, J., Chang, W., Zhou, Y., Li, Z., Chen, J., Shen, C., Pang, J., He, T.: Aether: Geometric-aware unified world modeling. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 8535–8546 (2025) 7