VDAWorld: World Modelling via VLM-Directed Abstraction and Simulation

FELIX O'MAHONY, ROBERTO CIPOLLA, and AYUSH TEWARI, University of Cambridge, United Kingdom

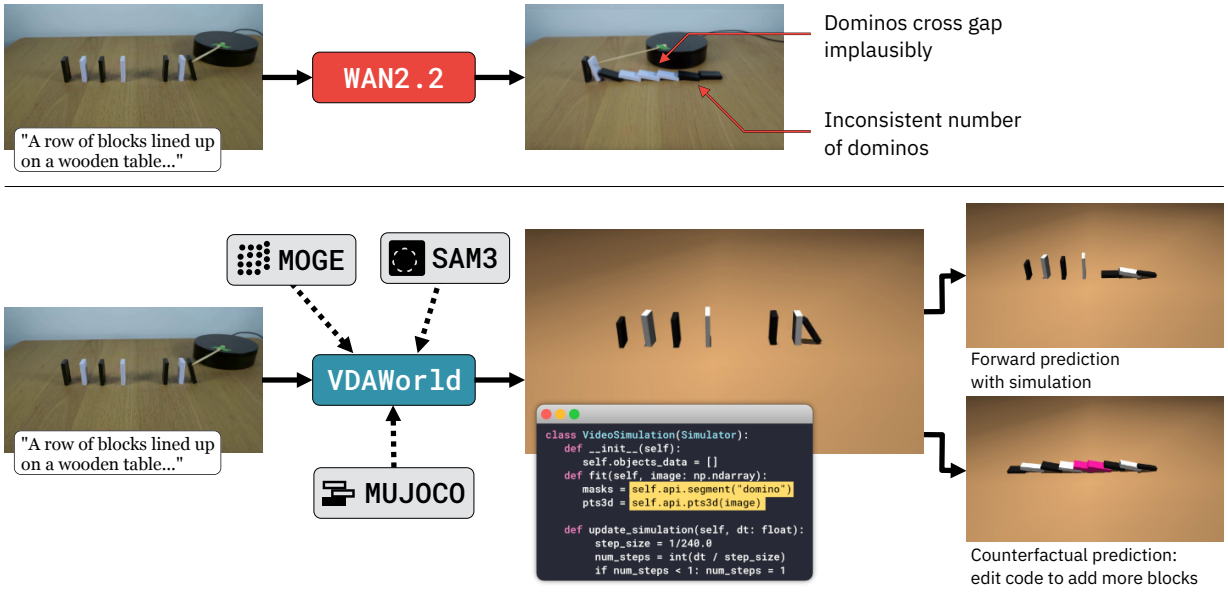

Fig. 1. Given an image and caption pair, VDAWorld builds a complete Pythonic simulator of the scene, resulting in an interactive, interpretable world model

Generative video models, a leading approach to world modeling, face fundamental limitations. They often violate physical and logical rules, lack interactivity, and operate as opaque black boxes ill-suited for building structured, queryable worlds. To overcome these challenges, we propose a new paradigm focused on distilling an image caption pair into a tractable, abstract representation optimized for simulation. We introduce VDAWorld, a framework where a Vision-Language Model (VLM) acts as an intelligent agent to orchestrate this process. The VLM autonomously constructs a grounded (2D or 3D) scene representation by selecting from a suite of vision tools, and accordingly chooses a compatible physics simulator (e.g., rigid body, fluid) to act upon it. VDAWorld can then infer latent dynamics from the static scene to predict plausible future states. Our experiments show that this combination of intelligent abstraction and adaptive simulation results in a versatile world model capable of producing high quality simulations across a wide range of scenarios. We demonstrate several applications of VDAWorld across interactive control and counterfactual generation, and physical and logical reasoning, achieving state-of-the-art results on several benchmarks.

1 Introduction

Understanding and forecasting how the visual world evolves is a core challenge for building intelligent systems. Humans can observe a static scene and infer not only its current structure but also how it might change over time and in response to different actions. This capability underlies essential skills such as planning, decision-making, and causal reasoning. A central hypothesis for how humans perform this reasoning is "world modelling", formalised as the process of constructing an internal, mental representation of the external world that can be used to predict future states and reason about cause and effect [Craik 1944]. Replicating this physical intuition in AI hinges on creating equivalent models to predict potential futures.

In recent years, the dominant paradigm for building such models has been large-scale video generation, e.g., Genie [Bruce et al. 2024]. By training on immense visual corpora, these models have achieved remarkable success in synthesizing complex, dynamic scenes with impressive visual realism, suggesting they are learning a powerful, albeit implicit, model of our world only from 2D observations and text descriptions [Wan et al. 2025]. However, despite their visual prowess, pixel-space models result in critical and systematic failures that limit their use as world models. First, they frequently generate physically implausible scenarios [Kang et al. 2024; Motamed et al. 2025]. Video models learn statistical correlations from pixels and do not enforce any physical plausibility constraints, which can result in their outputs often violating fundamental principles of object permanence, collision, and causality. This failure to deduce underlying principles is not limited to 3D physics; we show that these models are equally unable to infer the structured representations and simple, deterministic rules required to simulate abstract 2D environments like Conway's Game of Life [Conway et al. 1970] (Figure 5). Second, these models operate as opaque "black boxes". The generated scene is not a structured, queryable world but a sequence of pixels. Consequently, it is impossible to inspect the

Authors' Contact Information: Felix O'Mahony, foo222cam.ac.uk; Roberto Cipolla; Ayush Tewari, University of Cambridge, United Kingdom.

environment's underlying state or apply novel physical actions beyond those observed during training. Finally, as most video models are only conditioned on image and text inputs, they lack a native mechanism for physical interaction. While some models incorporate action conditioning [Google Deepmind 2025a; Song et al. 2025; Wang et al. 2023], the action space is typically limited to simple transformations, such as camera viewpoint, rather than complex physical interventions, and where models do incorporate physical interventions, these are typically only for narrow application domains such as driving [Wang et al. 2023].

Separate from video models, another significant line of research has focused on reconstructing 3D or 4D scene representations from images. Foundational models like DUST3R [Wang et al. 2024] have demonstrated impressive capabilities in producing dense and accurate geometric reconstructions, while methods based on Neural Radiance Fields (NeRFs) [Mildenhall et al. 2021] excel at generating photorealistic novel views. However, the primary objective of both lines of work is to capture a scene's geometry and appearance, not its underlying physical nature. This focus on 3D geometry also renders them, by design, inapplicable to abstract 2D environments governed by logical rules, such as cellular automata like Conway's Game of Life. PhysGen3D [Chen et al. 2025] and PhysGen [Liu et al. 2024] reconstruct scenes as simulations within fixed simulators. However, because these methods use a rigid pipeline, they are not applicable in a wide range of settings, and are exclusively able to simulate 3D material point and 2D rigid body scenes respectively.

In this paper, we introduce VDAWorld, a novel framework for world modeling that moves away from direct pixel prediction and instead builds an explicit, structured world representation. Instead of predicting pixels, our primary goal is to distil a visually complex image into a tractable abstract representation (Figure 1). This representation intentionally discards physically irrelevant information (like fine-grained textures or static backgrounds) to create a structured world model optimized for simulation. The use of simulators enforces structural plausibility, and provides an easy framework for intervention and action inputs within the model.

Our framework achieves this through a Vision-Language Model (VLM) that acts as a central agent, orchestrating three core innovations (see Figure 2). First, the VLM acts as an intelligent tool-using agent to construct a grounded representation. It is equipped with a versatile suite of vision modules, including segmentation, 3D reconstruction, and 3D mesh and primitive fitting, and autonomously decides which tools to deploy. This allows the representation to be grounded in the scene's native dimensionality; for instance, applying a full 3D pipeline for spatial environments while recognizing that such tools are irrelevant for planar ones, such as Conway's Game of Life. Second, the choice of representation and simulator is co-dependent and adaptive. The VLM jointly determines the type of abstraction and the most appropriate physics simulator to act upon it: a scene with blocks is abstracted into a rigid body model and paired with a rigid body solver, while one with water is represented as a particle system paired with a fluid dynamics engine. Finally, this structured world model enables VDAWorld to infer latent dynamics, predicting a scene's likely evolution from the image and caption alone based on visual cues and the description of the scene.

Through comprehensive experiments, we show this combination of intelligent abstraction, adaptive simulation, and inferred dynamics results in a world model that significantly outperforms prior methods in producing high-quality, physically and logically plausible simulations across a wide range of scenarios. Unlike video models, the output of our method is natively controllable, amenable to user fine-grained interactions that are far beyond training data, e.g., updating the gravity in a scene.

In summary, our contributions are as follows: (i) A new paradigm for world modelling that builds general purpose world models through structured simulation-ready representations, rather than pixel prediction, (ii) a framework where a VLM acts as an intelligent agent and an inverse dynamics engine to construct a grounded scene representation from an image-caption pair, leveraging a diverse toolbox of state-of-the-art computer vision tools, and (iii) demonstration of key advantages of this approach, including physical plausibility, general interactivity, and simulation times which naturally scale with the complexity of the underlying scene dynamics. Alongside evaluations on two existing benchmarks, we propose two additional benchmarks to evaluate physical and logical reasoning.

2 Previous Work

Representation Learning with Videos. Recent advances in generative methods have established large-scale video models as a dominant paradigm for modeling world dynamics. State-of-the-art models have demonstrated a remarkable ability to synthesize high-fidelity and temporally coherent videos from text and image inputs [Google Deepmind 2025b; OpenAI 2024; Runway 2025; Wan et al. 2025]. These models use diffusion or flow-matching approaches in a compressed latent space for video generation. While most models only condition on image and text inputs, some recent methods have enabled conditioning on other parameters, such as camera parameters, for more controllable video generation [Google Deepmind 2025a; Huang et al. 2025; Song et al. 2025].

Additionally, some models demonstrate conditioning on action inputs. Many of these tend to be limited to action in a controlled domain, such as robotic manipulation tasks [Zhu et al. 2024], egocentric pose [Bai et al. 2025] or driving [Wang et al. 2023]. More general approaches for action conditioning include the Dreamer architecture [Hafner et al. 2025] or DreamDojo [Gao et al. 2026]. However, it is very difficult to interact with these models outside of the specific control parameters developed during training. It is generally impossible to query the state of an object, apply a novel physical force, or explore alternative outcomes under different conditions. These models do not explicitly reason in a structured space, and instead directly perform computations in frame space. This leads their outputs to frequently violate fundamental principles of the real world [Kang et al. 2024; Li et al. 2025; Motamed et al. 2025]. Common failure cases include the violation of object permanence, where objects may inexplicably appear or vanish, and inconsistent causality, where actions do not have plausible consequences.

Video Joint Embedding Predictive Architecture (V-JEPA) [Bardes et al. 2024] learns to predict latent representations of the world rather than raw pixels. Further versions, such as Action Conditioned

V-JEPA 2 [Assran et al. 2025] additionally incorporate action conditioning, allowing the model to condition its predictions on actions and future states. However, the latent space is still learned implicitly from data and does not guarantee physical plausibility. Additionally, the latent representations are not necessarily interpretable or queryable, and the action conditioning is limited to specific transformations rather than general interventions. Finally, these models still operate as black boxes, and do not provide a mechanism for user intervention or control over the generated world.

Scene Reconstruction and Simulation. 3D reconstruction from images has achieved considerable success in recent years. Models like DUST3R [Wang et al. 2024] and its follow-ups [Feng* et al. 2025; Wang et al. 2025a; Zhang et al. 2024a] produce dense geometric reconstructions from images, while methods based on Neural Radiance Fields (NeRFs) [Tewari et al. 2023; Yu et al. 2021] and 3D Gaussians (3DGS) [Charatan et al. 2024; Szymanowicz et al. 2024] and their 4D extensions [Tretschk et al. 2021; Wu et al. 2024; Yang et al. 2023; Yunus et al. 2024] excel at generating photorealistic novel views. While most approaches do not inherently decompose the scene into discrete, object-centric components, some methods have made progress with the help of features from vision-language models [Jatavallabhula et al. 2023; Kerr et al. 2023]. Additionally, most neural radiance methods are optimized for appearance rather than physics, making them poorly suited for interactive simulation. Some work has attempted to model physics, for instance, by leveraging fixed physics engines and 3D scene representations [Chen et al. 2025; Feng et al. 2024; Kairanda et al. 2025; Le et al. 2025; Li et al. 2023; Petitjean et al. 2023; Wu et al. 2015; Xie et al. 2024; Zhang et al. 2024b] or 2D scene representations [Liu et al. 2024]. Two works similar to ours, PhysGen3D [Chen et al. 2025] and PhysGen [Liu et al. 2024], reconstruct object-centric 3D and 2D scenes respectively and perform simulation with a fixed physics engine. The use of a fixed simulator limits the generalisability of these methods across different types of scenes and dynamics, where different simulators might be more appropriate. Additionally, the rigidity of these approaches means they are less robust to failure cases in reconstruction, which often leads to catastrophic failures in simulation. In this work, we use tools from scene reconstruction and simulation methods but do not use a fixed pipeline as the VLM is free to select scene representations and simulators best suitable for any input.

Program Synthesis with VLMs. Our work is informed by recent advances in using Vision-Language Models (VLMs) as agents that synthesize programs to solve complex visual tasks. A foundational paradigm is visual program synthesis for querying. Methods like ViperGPT [Surís et al. 2023] and VisProg [Gupta and Kembhavi 2023] can parse a complex visual query into a sequence of steps, generating code that calls various vision APIs (e.g., object detectors, depth estimators) to arrive at a final answer. LayoutGPT [Feng et al. 2023] uses an LLM to generate a complete scene layout, including the sizes, positions, and relationships of different objects. Other research has shown that a VLM can evolve interpretable visual classifiers [Chiquier et al. 2024] and design interpretable programs to describe underlying scientific laws [Mall et al. 2025]. A second major application is in high-level planning and robotics. This research aims to create agents that can reason about the world to perform actions. VisualPredicator [Liang et al. 2024], for example, learns neuro-symbolic predicates that classify the state of the world for a symbolic planner. Others, like VoxPoser [Huang et al. 2023], use LLMs to synthesize 3D affordances that guide a low-level motion planner. Finally, some work has explored the use of VLMs to guide editing in structured 3D environments, such as BlenderAlchemy [Huang et al. 2024]. Concurrent work, Vision as Inverse Graphics [Yin et al. 2026], also explores the use of VLMs to generate interactive scenes. However, this approach is focused on generating a single 3D scene representation from an image, and does not explore the generation of a simulator or the inference of dynamics. The common thread in this research is that the VLM's role is to generate a plan or a set of actions for an agent to execute within an existing environment.

3 VDAWorld: VLM-Directed Abstraction and Simulation

The input to VDAWorld is a pair consisting of a single image of a scene and a text prompt that describes the scene. The goal of VDAWorld is to convert this static input into a dynamic, interactive world model. This process is orchestrated by a central Vision-Language Model (VLM), which acts as an inverse dynamics model to generate a "world program" in Python. This generated output captures three key components: (1) *A Grounded Abstract Representation*: The VLM selects from a suite of vision tools and writes a program to construct a 2D or 3D model of the scene, optimized for simulation, (2) *Inferred Action Dynamics*: It predicts the most likely abstract action (e.g., an impulse, velocity, or transformation) from the visual and textual cues, which drives the simulation, (3) *A Simulator Program*: It determines the most compatible simulation method (e.g., rigid body, fluid, logic) to progress the scene's dynamics forward in time. Once generated, this world program is executed with the inferred action to predict a plausible future. Because the system cleanly factors the explicit structured world and its action parameterization, both can be modified with novel user-defined interventions to imagine diverse futures.

3.1 Formal Formulation

Formally, we consider the task of world modeling as learning a function that maps an initial observation and action conditioning to samples from the distribution over future states. This process is mediated by a generated initial state program P_s, a transition program P_τ, and an inferred action sequence $a_{0:T}$. The method can be decomposed into two stages: abstraction and simulation.

Abstraction. The abstraction phase begins with a generative function Φ which interprets the raw visual inputs and the caption C to produce a structured world program representation and action sequence:

$$P_s, a_{0:T}, P_\tau, \mathcal{R} = \Phi(\mathcal{I}, C). \tag{1}$$

All of these components are instantiated as methods and objects within a single executable Python script. Specifically, P_s encapsulates the construction of the scene's representation (geometry, objects, and initial parameters), $a_{0:T}$ encapsulates the predicted action dynamics over time, and P_τ encapsulates the dynamic simulation engine and update rules. The last component, $\mathcal{R}$ is a rendering function used to visualize the state of the world at a given timestep.

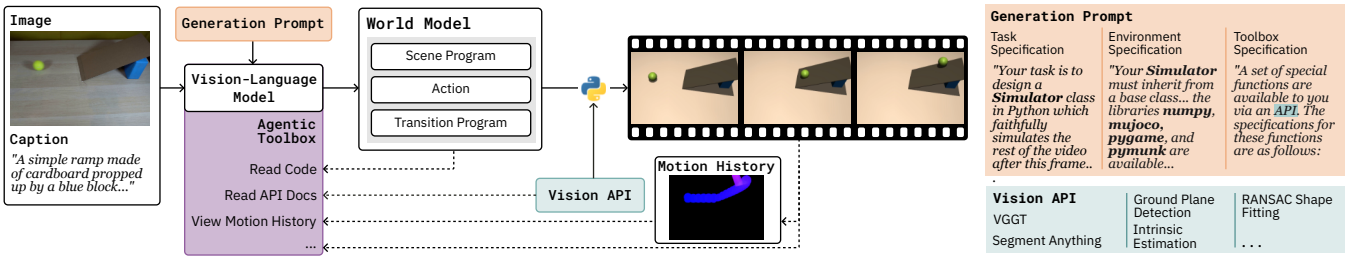

Fig. 2. Illustration of our method. VDAWorld takes a single image and text caption as input. We design a generation prompt that the VLM uses to generate a model of the scene, including the scene program, actions, and a transition program. The simulator code is then executed to generate future predictions. We can also generate other diverse videos by interactively changing the actions. The VLM operates in an agentic workflow, refining its code by reading frames and analysing motion from the output simulation.

The action prediction $a_{0:T}$ is jointly inferred alongside the scene representation, meaning the VLM acts not only as a visual parser but formally as an inverse dynamics model or action inference engine. The inferred action sequence $a_{0:T}$ is highly flexible and general in character. In physical simulation, a_t might consist of the application of an impulse or a velocity at time t. It might also consist of a geometric transformation of the ego camera, resulting in a simulation where the view of the scene transforms over time. It could also be the case that the action is essentially null ($a_t = \emptyset$), and the physics of the simulation evolve naturally on account of the starting conditions of the objects in the simulation.

The initial state s_0 is given by executing P_s on the image, yielding the structured abstract representation of the scene:

$$s_0 = \text{Exec}(P_s, \mathcal{I}). \tag{2}$$

Simulation. The simulation stage then executes these programs alongside the predicted action to rollout the future states of the world over time t. Subsequent states are generated by the transition program P_τ using the state and action at time t:

$$s_{t+1} = \text{Exec}(P_\tau, s_t, a_t), \tag{3}$$

where s_t represents the state at time t. The scene at any time step can be rendered as $\hat{\mathcal{I}}_t = \mathcal{R}(s_t)$, though we note that this rendered output is strictly apart from the world model state, and is used primarily for visualization and evaluation purposes. The core world model is defined by the state s_t and its dynamics under the transition program P_τ and action sequence $a_{0:T}$.

The programmatic nature of VDAWorld allows for explicit user intervention, serving as a powerful mechanism for flexible action conditioning. We will demonstrate four primary modes of intervention: (1) *Caption Modification*: Altering the prompt C to generate a revised pair of programs and actions. (2) *Action Modification*: Explicitly editing the code defining the inferred action sequence $a_{0:T}$. (3) *Transition Function Modification*: Modifying the physical rules or engine logic in P_τ. (4) *Initial State Modification*: Changing the starting configuration or geometry instantiated by P_s. These enable counterfactual reasoning and control, where the simulation rollout follows the new, action-conditioned dynamics $s'_{t+1} = \text{Exec}(P'_\tau, s'_t, a'_t)$. By factorizing the world modeling problem through explicit programs P_s and P_τ and actions $a_{0:T}$, VDAWorld ensures that the generated dynamics are consistent, interpretable, and amenable to structured user intervention.

To practically create the accurate generation of these world states and actions we implement a comprehensive toolbox of vision tools which are available for the VLM to select from.

3.2 Vision API Usage

An important component of VDAWorld is the provision of a suite of computer vision tools via an API which allow the VLM to build a simulator which matches the provided scene image. We implement a suite of perception and geometry modules that the VLM can call to perform a range of tasks. The VLM does not re-implement or update the implementation of these tools, and it is free to ignore these implementations if it does not find any use for them. The toolbox includes perception tools, such as SAM 3 [Carion et al. 2025]; geometry tools such as MoGe-2 [Wang et al. 2025b]; and SAM 3D for mesh extraction from images [Team et al. 2025a]. Primitive fitting methods are also provided to fit geometric shapes such as circles, squares, cubes and cylinders to collections of points using a RANSAC-based approach. A full specification of the toolbox is provided in supplementary material.

3.3 Prompting for World Program Generation

The core of VDAWorld lies in guiding a powerful Vision-Language Model (VLM) to generate a complete, executable world program via an agentic pipeline. Instead of fine-tuning, we steer the model's behavior at inference time using a comprehensive, multi-part prompt.

Task Specification. The prompt begins with a high-level task specification in natural language. This instruction outlines the overall objective: to analyze a user-provided image and text description and produce a self-contained Python script that simulates the scene's future. It directs the VLM to analyze the user-provided inputs (the image $\mathcal{I}$ and caption C) and produce a self-contained Python script representing both the logical programs P_s, P_τ as methods, along with the parameterized action $a_{0:T}$.

Environment Specification. The environment specification provides the formal scaffolding for the VLM's code generation task. Its central element is a Python Simulator base class that the VLM must inherit from in its world program. This base class defines the core methods the VLM must implement, enforcing the entire simulation

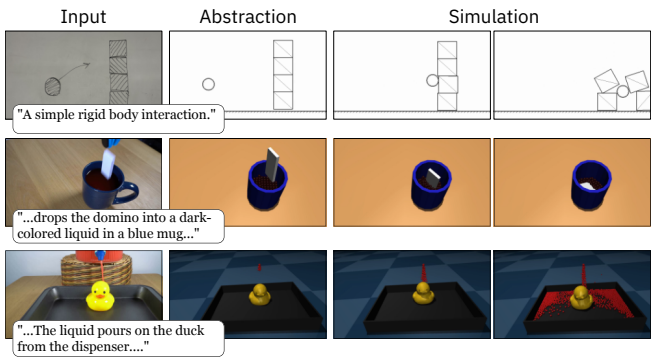

Fig. 3. VDAWorld generates an abstraction of the scene and a simulator that can together be used to predict future scene states. From top to bottom: (Abstraction) A collision drawn on a whiteboard, and (Rigid Body) A domino dropping into a mug of liquid, (Fluid Dynamics) Liquid pouring onto a duck in a tray.

logic from scene setup to frame-by-frame execution. Crucially, the methods of this class explicitly implement the programs defined in our formal formulation: prescribed methods realize the initial state program P_s, the transition program P_τ, and establish the inferred action $a_{0:T}$. Additionally, the environment provides the VLM with a list of existing python libraries that it can rely on, pointing it to common simulation implementations.

API Specification. Finally, we provide the VLM with a specification of the diverse computer vision toolkit implemented to support the creation of a simulator which appropriately matches the provided scene image.

3.4 Agentic Loop

While a single generation pass can produce high-quality results, high complexity in scenes can lead to errors in the initial code or inaccurate scene fits. To enhance the robustness of our system, we utilise an agentic implementation which can iteratively refine and improve its own generations. This is illustrated in Figure 2. In the agentic implementation, the VLM can independently call a set of tools to interpret its own implementation, and iteratively refine the code it has written. The agentic tools provided can be broken into three areas. The first set are implementational tools. These include the ability to write code from scratch; to edit code with unified diff blocks; to read the documentation associated with each API method; and to execute code. The second set allow the model to debug its code. These tools include reading back its own code, reading error messages from the terminal, reading standard output from the terminal, and reading metadata associated with each API call (for example, the number of objects segmented by a single prompt). Finally, the third set allow the VLM to critique a resulting simulation. These tools include the ability to view individual frames of simulation.

An important tool we introduce is the ability to view a motion history image [Bobick and Davis 2001]. The motion history image visualizes all dynamic activity over the simulation's duration. The exact formulation of the motion history image we implement uses image hue to encode the time of motion, and image saturation to encode the frequency of motion, while the value channel encodes whether any motion occurred at all. The full equations for this formulation are provided in the supplementary material. These tools are important for the model's ability to produce coherent accurate simulations (Table 4).

Model	Solid Mech.	Fluid Dyn.	Thermo.	Magnetism	Combined
Best of three trials					
VDAWorld	54.9	44.7	32.6	25.5	49.7
PhysGen	29.7	24.6	29.6	23.0	28.3
PhysGen3D	12.2	10.2	0.4	17.2	11.7
Wan2.2	47.3	49.7	46.5	20.0	46.2
One trial					
VDAWorld	42.6	29.1	17.4	22.0	37.3
PhysGen	26.9	22.0	28.7	21.5	25.7
PhysGen3D	6.6	7.7	0.4	17.2	7.1
Wan2.2	39.2	39.5	23.6	11.3	36.8
VideoPoet†	35.1	24.6	29.1	44.0	32.8
Runway	27.5	27.2	20.7	17.9	27.1
Lumiere†	27.3	23.5	41.5	19.7	26.9
Lumiere	22.0	25.4	33.8	19.5	23.5

Table 1. Results on PhysicsIQ. † indicates methods which take videos as inputs. The best and second best numbers are highlighted. VDAWorld outperforms all baselines in both best-of-three and one-trial settings in the final score, and achieves best or second best performance on all categories.

4 Experiments

We evaluate our approach to demonstrate three core advantages over standard video models: (1) flexibility across diverse target domains and applications, (2) physical plausibility via executable code, and (3) interpretability enabling zero-shot control. Our implementation uses Gemini 3.1 Pro [DeepMind 2025] as the core VLM coding agent. Computations use NVIDIA RTX Pro 6000 Blackwell GPUs. Code will be publicly available.

4.1 Predicting the Future

Figure 3 presents qualitative results of VDAWorld on complex solid body and thermodynamic dynamics. The top row in particular highlights our method's ability to select appropriate abstractions: for a whiteboard drawing, it correctly infers a simple 2D abstraction is sufficient. By intentionally discarding high-frequency visual details, VDAWorld focuses exclusively on producing accurate, physically plausible simulations rather than visual photorealism.

We use the existing PhysicsIQ benchmark [Motamed et al. 2025], and introduce two new benchmarks: HSPBench, and ConwayBench. We benchmark against leading video generation models: Wan2.2 [Wan et al. 2025] (primary open-source baseline), Lumiere [Bar-Tal et al. 2024], VideoPoet [Kondratyuk et al. 2023], and select examples from Veo3 [Google Deepmind 2025b][1]. We also compare against physical simulation approaches, PhysGen [Liu et al. 2024] and PhysGen3D [Chen et al. 2025]. As these rely on specific simulators, they represent a subset of VDAWorld's flexible capabilities.

4.1.1 PhysicsIQ. To demonstrate the flexibility of our approach on standard mechanical reasoning tasks, we evaluate on the PhysicsIQ benchmark [Motamed et al. 2025], which contains real-world videos

[1] A full Veo3 evaluation was cost-prohibitive.

	Simulation → VQA			*VQA*	
Metric	VDAWorld	Wan+Gemini	Veo+Gemini	Gemini Pro	Gemini Flash
VQA Error ↓	75	227	291	133	175
Spatial ↑	0.66	0.19	0.22	-	-
Spatiotemporal ↑	0.25	0.05	0.04	-	-
Weighted Spatial ↑	0.73	0.28	0.32	-	-
Combined ↑	0.55	0.17	0.19	-	-

Table 2. Results on HSPBench. **(Top)** We achieve significantly better scores on VQA on questions about the future state of the scene. **(Bottom)** We achieve more accurate simulations than the baselines, measured using PhysicsIQ metrics. The best and second best numbers are highlighted.

of diverse physical phenomena. We evaluate on solid mechanics, fluid dynamics, magnetism, and thermodynamics, excluding optics as our method abstracts visual appearance. This results in a total of 174 test samples. Following the PhysicsIQ protocol, we compute a final score (out of 100) combining Spatial IoU, Weighted Spatial IoU, and Spatiotemporal IoU to evaluate motion accuracy. We omit MSE, as our goal is physically plausible dynamics rather than pixel-perfect reconstruction. We present results in two settings: a one-trial result as well as a best-of-three-trials result that selects the best performing output among three trials. As the future prediction problem here has inherent uncertainty, we believe the best-of-three setting presents a more accurate picture.

Table 1 shows that VDAWorld outperforms all baselines in the best-of-three setting, achieving a combined score of 49.7 compared to 46.2 for the strongest video baseline, Wan2.2. In the one-trial setting, VDAWorld slightly outperforms Wan2.2 and significantly outperforms all other baselines. These results show that explicit simulation-based world models can match or exceed pixel-space video models on physical prediction metrics, while additionally providing explicit state, interpretability, and direct counterfactual control. Our qualitative results (Figure 4 and supplementary video) reveal that the next-best method, Wan2.2, frequently produces non-physical artifacts, such as objects merging, demonstrating the lack of a true underlying physics model, whereas VDAWorld generates physically plausible results.

As PhysGen and PhysGen3D rely on a fixed pipeline with a predetermined simulator, they struggle to generalize across the diverse dynamics in this benchmark, often producing simulations that fail to capture the nuances of true physical behaviour. Notably, this resulted in a high failure rate of 69.9% for PhysGen3D. VDAWorld is the first demonstration of a simulation-based method that is competitive with state-of-the-art video models.

4.1.2 HSPBench. To explicitly evaluate physical accuracy in a deterministic setting, we present a physics experiment benchmark consisting of 40 high-school level problems (e.g., mass-spring systems), see Figure 11 for an example. Conditioned on an initial state diagram and descriptive caption, these deterministic scenarios cleanly test a model's adherence to physical rules. Models are assessed on accuracy compared to objective deviation from these rules using the PhysicsIQ metrics described earlier. A full description of this benchmark is given in supplementary material. While video models frequently violate basic physical principles, our approach locks the dynamics into a physics engine, producing significantly better simulations (Table 2 (Bottom) and Figure 11).

4.1.3 ConwayBench. Moving beyond physics, we present a benchmark using Conway's Game of Life [Conway et al. 1970], a discrete cellular automaton. This purely logical setting demonstrates VDAWorld's flexibility to generate correct simulations in domains governed by abstract rules rather than physical laws. We task the model to predict board evolution from a single input frame across six scenes with different starting configurations and textures, evaluating accuracy per frame via the F1 score. Because our approach generates an executable simulation, it acts as a guaranteed correct rule engine, achieving a perfect F1 score and significantly outperforming video models (Figure 6 (right), Figure 5). The Gemini Pro baseline solves simple scenes but fails to parse complex initial states, leading to diverging simulations. State-of-the-art video models are unable to capture the dynamics of this problem and achieve low performance.

4.1.4 Analysis.

Simulation Time. We highlight our method's computational efficiency. Unlike video models that incur a large, fixed cost to generate pixels, the execution time (Figure 6, left) of our underlying simulator scales intrinsically with physical complexity. Here, execution time refers specifically to the time taken to run the transition function program P_τ for simulation. This aligns with the intuitive requirement of efficient world modeling: simple systems like Game of Life simulate roughly 100× faster than complex thermodynamics.

Even when considering total computation time, VDAWorld is faster than video models. The total computation time defines the entire inference process. Even with the various required stages, on average, computing scenes for the PhysicsIQ benchmark takes 11 minutes for VDAWorld, compared to 30 minutes for Wan2.2. For the Game of Life benchmark, VDAWorld computes a perfect simulation in 5 minutes on average, while Wan2.2 still takes 30 minutes to generate a diverging video.

Ablation. We ablate the core components of our system on the PhysicsIQ benchmark (best-of-1 setting, central perspective only). As shown in Table 4, all components of our method are essential. We also show that using the Gemini Flash model leads to worse performance. Finally, we evaluate a VLM-only baseline where Gemini 3.1 Pro is asked to directly generate the output simulation code. This effectively removes API, Tool Use, and MHI all at the same time. The VLM-only baseline shows that generic code generation is insufficient. The gains come from the constrained world-program interface, simulation-grounded vision API, action/transition decomposition, and refinement. These low scores demonstrate that all components of our model are essential.

4.2 Applications

Counterfactual Generation. Because VDAWorld generates code (the programs P_s, P_τ, and action $a_{0:T}$) rather than relying on black-box latents, its representation is fundamentally interpretable. This allows for intuitive, zero-shot human intervention: if physics require adjustment or a counterfactual is desired, a user simply edits the code. Figure 7 highlights several intervention modes. *Caption Modification:* Altering the prompt C changes the simulation from a rigid-body collision to simple camera motion, changing the inferred

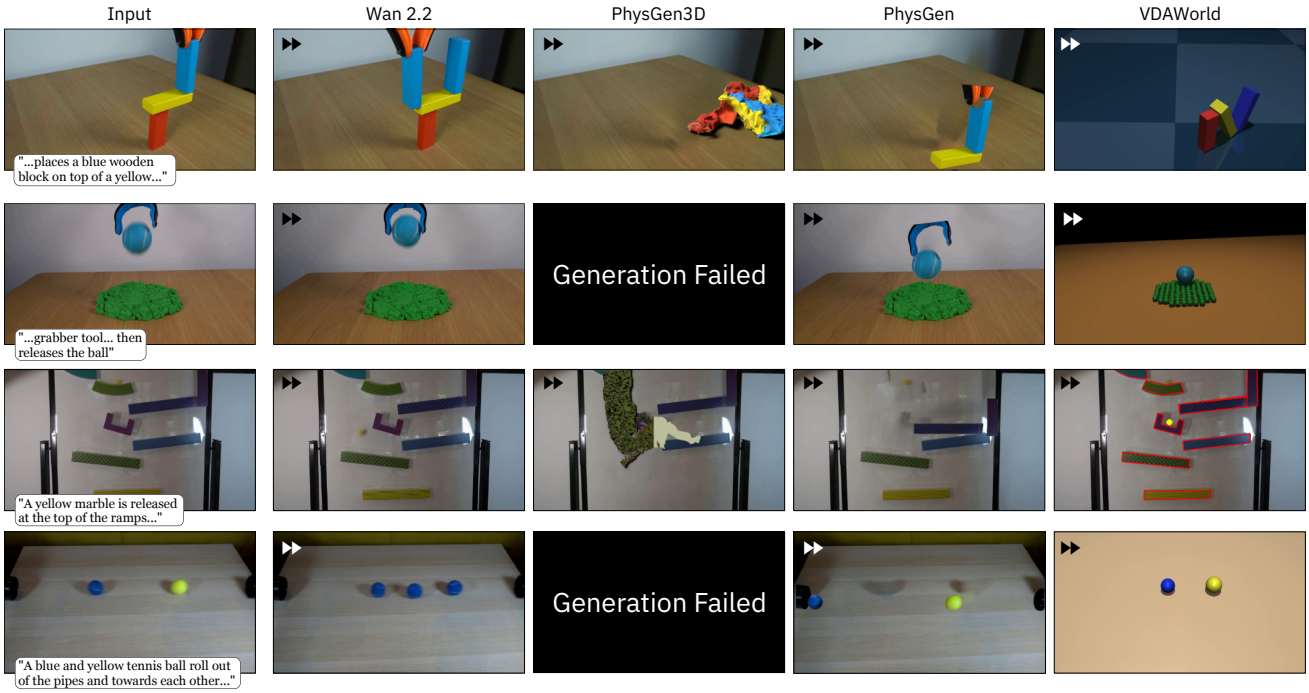

Fig. 4. Qualitative results on the PhysicsIQ benchmark. ▸▸ *denotes selecting a frame after iterating the simulation.* Refer to the supplemental webpage for many more results.

Metric	Veo3.1	Veo3.1-fast	Sora2	Seedance 1.0-pro	Seedance 1.0-fast	Kling-v2.1	Cosmos Predict	VDAWorld
Reasoning Score	55.9	55.9	49.9	35.7	37.5	13.8	28.6	60.5
Consistency Score	47.5	45.3	60.5	41.5	44.5	58.3	37.5	74.7
Average Score	49.5	47.6	60.9	43.9	47.4	65.1	37.8	73.6

Table 3. Comparison of various methods on the MME-CoF-Pro benchmark. The best and second best numbers are highlighted. VDAWorld outperforms a wide range of state-of-the-art baselines.

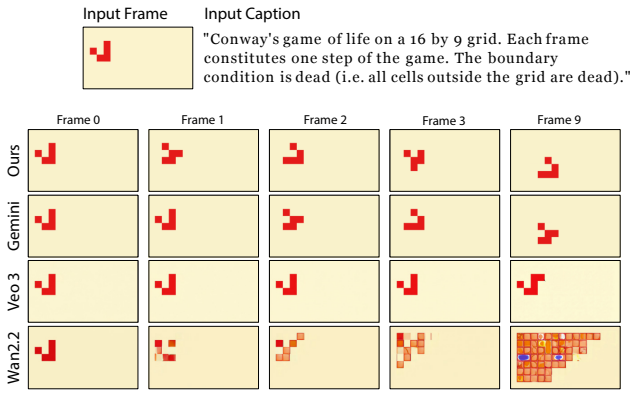

Fig. 5. A sample from our Conway's Game of Life benchmark, Conway-Bench. Our method is the only one that perfectly captures this logic simulation task. Video models in particular generate significant artifacts.

Model	Score	Difference
VDAWorld	40.5	-
Ablate MHI	39.0	-3.8%
Ablate API	36.2	-10.7%
Ablate Caption	35.2	-13.2%
Ablate Image	34.6	-14.6%
Ablate Tool Use	27.0	-33.5%
VDAWorld with Gemini 3.1 Flash	32.4	-20.1%
VLM only	17.7	-56.4%

Table 4. Ablation results demonstrate the significance of the various components of our modelling strategy.

action sequence. *Transition Function Modification:* Modifying the Game of Life code creates a novel cellular automaton. *Action Modification:* Directly editing a robotic arm's trajectory code provides precise kinematic control, specifying the action conditioning $a'_{0:T}$. Finally, we demonstrate *Initial State Modification* in Figure 1, directly modifying the initial geometry instantiated by P_s to add two more dominoes.

We also demonstrate controllable generation results on some complex and cluttered scenes in Figure 8.

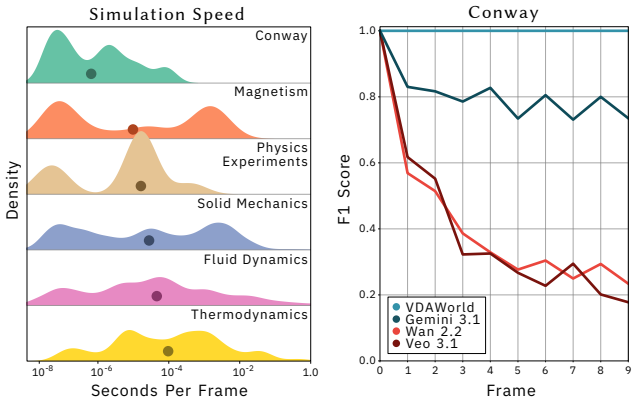

Fig. 6. **(Left)** Comparison of simulation execution time densities over all benchmarks using a Kernel Density Estimation plot [Parzen 1962; Rosenblatt 1956]. Video models inherently pay a fixed pixel-generation cost, whereas VDAWorld scales with physical complexity, resulting in significantly faster execution for simpler dynamics. **(Right)** F1 score computed between ground truth sequences and predicted states on a per-frame basis. On Game of Life, VDAWorld achieves a perfect F1 score, significantly outperforming video models and our Gemini 3.1 Pro baseline.

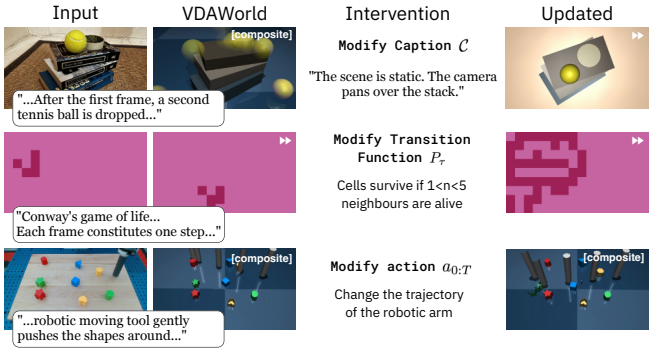

Fig. 7. Our structured, programmatic representation enables zero-shot user control. **Row 1:** Caption modification alters dynamics from a collision to a camera pan. **Row 2:** Transition function modification invents novel Conway rules. **Row 3:** Action modification provides fine-grained trajectory control over a robotic arm. *[composite] denotes that the image is a composite to visualize the trajectory.*

Visual Reasoning. We first evaluate visual reasoning capabilities of our method using the HSPBench dataset. For every scene, we ask two questions about the future that has a deterministically correct answer, e.g., where would the ball in the simulation be on the x-axis after 5 seconds? The input and caption fully specify the initial conditions, making the future evolution deterministic. This gives a total of 80 question-answer pairs. We report errors using a VQA error metric that measures the difference between the predicted and GT answer. We pass the output simulations of the baselines to Gemini and ask it to interpret the answer. We also report the performance of Gemini VLM when it is directly asked for the answer. Table 2 demonstrates that we significantly outperform all baselines.

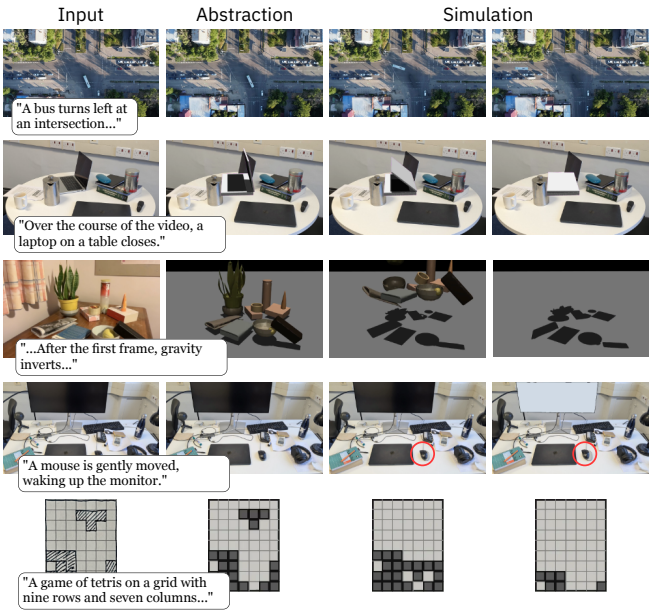

Fig. 8. A selection of more complex scenes, highlighting the flexibility and versatility of VDAWorld. These scenes involve aerial traffic scenes, large numbers of interacting objects, and a requirement to abstract a world from a sketch. Note that VDAWorld is required to select the most appropriate level of simulation fidelity to match the complexity of each scene and the desired interaction. Red circle added to emphasise movement.

We also report results on the MME-CoF-Pro benchmark [Qi et al. 2025] that includes 303 test samples across 16 categories evaluating capabilities ranging from visual logic to scientific reasoning evaluation. Example tasks include generating the correct gripper trajectory to place a spatula into a pot and rotating a scene clockwise to make a person upright. We use their code and metrics (no-hint setting) and report results in Table 3. This benchmark allows us to compare to several contemporary video models, Veo 3.1 [Google Deepmind 2025b], Sora2 [OpenAI 2025], SeedDance 1.0 [Gao et al. 2025], Kling 2.1 [Team et al. 2025b], and Cosmos Transfer [NVIDIA et al. 2025]. VDAWorld outperforms all baselines in this challenging benchmark, demonstrating that code-based simulation can advance visual reasoning significantly.

5 Discussion and Conclusion

While a central advantage of VDAWorld is that the world model is kept abstract, and while we demonstrate the advantage of this representation in several downstream applications, there may be settings in which reconstructing pixels from simulated videos is desirable. Generating photorealistic videos is not the focus of this paper, and we do not present comprehensive experiments to this effect; however, several off-the-shelf content-transfer methods could potentially address this task. We demonstrate initial results with Cosmos Transfer [NVIDIA et al. 2025] on simulations returned by VDAWorld in Figure 9. Although these are not pixel-perfect reconstructions of the original scene, they suggest that appearance

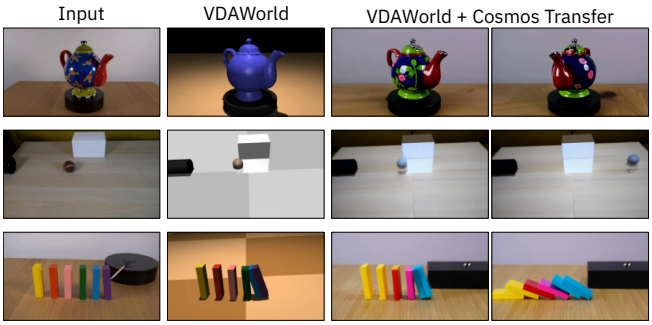

Fig. 9. We use Cosmos Transfer [NVIDIA et al. 2025] to generate photorealistic results. While the appearance is not perfectly maintained, the results indicate feasibility of this task.

transfer from executable simulations is a promising direction for future work.

VDAWorld is also limited by the quality and coverage of its perception tools, simulator library, and coding VLM. Errors in segmentation or 3D reconstruction can produce simulations that are internally consistent but do not correctly match the input scene. As visual foundation models improve, the fidelity of scenes that VDAWorld can handle will improve as well.

Conclusion. In this work we introduced VDAWorld, a new paradigm for building dynamic world models from static images. We have shown that by tasking a Vision-Language Model with world program synthesis, it is possible to generate explicit, executable simulations that are physically plausible, interactive, and versatile. Our experiments demonstrate that this approach avoids common physical artifacts of pixel-prediction models and excels at tasks requiring precise, rule-based reasoning. This programmatic approach represents a significant step towards creating more grounded and interactive world models. We believe our work points to a broader shift in how we build autonomous agents. Instead of relying on monolithic, end-to-end models that learn an opaque representation of the world, VDAWorld functions as a compositional agent that reasons about the world and writes code to model it.

6 Acknowledgements

We would like to thank Dr Matthew Johnson for discussions and advice. We acknowledge the use of resources provided by the Isambard-AI National AI Research Resource (AIRR). Isambard-AI is operated by the University of Bristol and is funded by the UK Government's Department for Science, Innovation and Technology (DSIT) via UK Research and Innovation; and the Science and Technology Facilities Council [ST/AIRR/I-A-I/1023] [McIntosh-Smith et al. 2024].

References

Mahmoud Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba Komeili, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, Sergio Arnaud, Abha Gejji, Ada Martin, Francois Robert Hogan, Daniel Dugas, Piotr Bojanowski, Vasil Khalidov, Patrick Labatut, Francisco Massa, Marc Szafraniec, Kapil Krishnakumar, Yong Li, Xiaodong Ma, Sarath Chandar, Franziska Meier, Yann LeCun, Michael Rabbat, and Nicolas Ballas. 2025. V-JEPA 2: Self-Supervised Video Models Enable Understanding, Prediction and Planning.

Yutong Bai, Danny Tran, Amir Bar, Yann LeCun, Trevor Darrell, and Jitendra Malik. 2025. Whole-Body Conditioned Egocentric Video Prediction. arXiv:2506.21552 [cs.CV] https://arxiv.org/abs/2506.21552

Omer Bar-Tal, Hila Chefer, Omer Tov, Charles Herrmann, Roni Paiss, Shiran Zada, Ariel Ephrat, Junhwa Hur, Guanghui Liu, Amit Raj, et al. 2024. Lumiere: A space-time diffusion model for video generation. In *SIGGRAPH Asia 2024 Conference Papers*. 1–11.

Adrien Bardes, Quentin Garrido, Jean Ponce, Michael Rabbat, Yann LeCun, Mahmoud Assran, and Nicolas Ballas. 2024. Revisiting Feature Prediction for Learning Visual Representations from Video.

A.F. Bobick and J.W. Davis. 2001. The recognition of human movement using temporal templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23, 3 (2001), 257–267. doi:10.1109/34.910878

Jake Bruce, Michael Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, Yusuf Aytar, Sarah Bechtle, Feryal Behbahani, Stephanie Chan, Nicolas Heess, Lucy Gonzalez, Simon Osindero, Sherjil Ozair, Scott Reed, Jingwei Zhang, Konrad Zolna, Jeff Clune, Nando de Freitas, Satinder Singh, and Tim Rocktäschel. 2024. Genie: Generative Interactive Environments. arXiv:2402.15391 [cs.LG] https://arxiv.org/abs/2402.15391

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, Jie Lei, Tengyu Ma, Baishan Guo, Arpit Kalla, Markus Marks, Joseph Greer, Meng Wang, Peize Sun, Roman Rädle, Triantafyllos Afouras, Effrosyni Mavroudi, Katherine Xu, Tsung-Han Wu, Yu Zhou, Liliane Momeni, Rishi Hazra, Shuangrui Ding, Sagar Vaze, Francois Porcher, Feng Li, Siyuan Li, Aishwarya Kamath, Ho Kei Cheng, Piotr Dollár, Nikhila Ravi, Kate Saenko, Pengchuan Zhang, and Christoph Feichtenhofer. 2025. SAM 3: Segment Anything with Concepts. arXiv:2511.16719 [cs.CV] https://arxiv.org/abs/2511.16719

David Charatan, Sizhe Lester Li, Andrea Tagliasacchi, and Vincent Sitzmann. 2024. pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 19457–19467.

Boyuan Chen, Hanxiao Jiang, Shaowei Liu, Saurabh Gupta, Yunzhu Li, Hao Zhao, and Shenlong Wang. 2025. PhysGen3D: Crafting a Miniature Interactive World from a Single Image. *CVPR* (2025).

Mia Chiquier, Utkarsh Mall, and Carl Vondrick. 2024. Evolving interpretable visual classifiers with large language models. In *European Conference on Computer Vision*. Springer, 183–201.

John Conway et al. 1970. The game of life. *Scientific American* 223, 4 (1970), 4.

Kenneth Craik. 1944. The nature of explanation. https://api.semanticscholar.org/CorpusID:41364251

Google DeepMind. 2025. Introducing Gemini 3 — blog.google. https://blog.google/products-and-platforms/products/gemini/gemini-3-collection/. [Accessed 02-03-2026].

Haiwen Feng*, Junyi Zhang*, Qianqian Wang, Yufei Ye, Pengcheng Yu, Michael J. Black, Trevor Darrell, and Angjoo Kanazawa. 2025. St4RTrack: Simultaneous 4D Reconstruction and Tracking in the World. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Weixi Feng, Wanrong Zhu, Tsu-jui Fu, Varun Jampani, Arjun Akula, Xuehai He, Sugato Basu, Xin Eric Wang, and William Yang Wang. 2023. Layoutgpt: Compositional visual planning and generation with large language models. *Advances in Neural Information Processing Systems* 36 (2023), 18225–18250.

Yutao Feng, Yintong Shang, Xuan Li, Tianjia Shao, Chenfanfu Jiang, and Yin Yang. 2024. Pie-nerf: Physics-based interactive elastodynamics with nerf. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4450–4461.

Shenyuan Gao, William Liang, Kaiyuan Zheng, Ayaan Malik, Seonghyeon Ye, Sihyun Yu, Wei-Cheng Tseng, Yuzhu Dong, Kaichun Mo, Chen-Hsuan Lin, Qianli Ma, Seungjun Nah, Loic Magne, Jiannan Xiang, Yuqi Xie, Ruijie Zheng, Dantong Niu, You Liang Tan, K.R. Zentner, George Kurian, Suneel Indupuru, Pooya Jannaty, Jinwei Gu, Jun Zhang, Jitendra Malik, Pieter Abbeel, Ming-Yu Liu, Yuke Zhu, Joel Jang, and Linxi "Jim" Fan. 2026. DreamDojo: A Generalist Robot World Model from Large-Scale Human Videos. *arXiv preprint arXiv:2602.06949* (2026).

Yu Gao, Haoyuan Guo, Tuyen Hoang, Weilin Huang, Lu Jiang, Fangyuan Kong, Huixia Li, Jiashi Li, Liang Li, Xiaojie Li, et al. 2025. Seedance 1.0: Exploring the boundaries of video generation models. *arXiv preprint arXiv:2506.09113* (2025).

Google Deepmind. 2025a. Genie 3: A new frontier for world models. https://deepmind.google/discover/blog/genie-3-a-new-frontier-for-world-models/.

Google Deepmind. 2025b. *Veo: a text-to-video generation system.* Technical Report. https://storage.googleapis.com/deepmind-media/veo/Veo-3-Tech-Report.pdf

Tanmay Gupta and Aniruddha Kembhavi. 2023. Visual programming: Compositional visual reasoning without training. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 14953–14962.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. 2025. Mastering diverse control tasks through world models. *Nature* (2025), 1–7.

Ian Huang, Guandao Yang, and Leonidas Guibas. 2024. Blenderalchemy: Editing 3d graphics with vision-language models. In *European Conference on Computer Vision*.

Springer, 297–314.
Tianyu Huang, Wangguandong Zheng, Tengfei Wang, Yuhao Liu, Zhenwei Wang, Junta Wu, Jie Jiang, Hui Li, Rynson WH Lau, Wangmeng Zuo, and Chunchao Guo. 2025. Voyager: Long-Range and World-Consistent Video Diffusion for Explorable 3D Scene Generation. *arXiv preprint arXiv:2506.04225* (2025).
Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. 2023. Voxposer: Composable 3d value maps for robotic manipulation with language models. *arXiv preprint arXiv:2307.05973* (2023).
Krishna Murthy Jatavallabhula, Alihusein Kuwajerwala, Qiao Gu, Mohd Omama, Tao Chen, Alaa Maalouf, Shuang Li, Ganesh Iyer, Soroush Saryazdi, Nikhil Keetha, et al. 2023. Conceptfusion: Open-set multimodal 3d mapping. *arXiv preprint arXiv:2302.07241* (2023).
Navami Kairanda, Marc Habermann, Shanthika Naik, Christian Theobalt, and Vladislav Golyanik. 2025. Thin-Shell-SfT: Fine-Grained Monocular Non-rigid 3D Surface Tracking with Neural Deformation Fields. In *Proceedings of the Computer Vision and Pattern Recognition Conference*. 11373–11383.
Bingyi Kang, Yang Yue, Rui Lu, Zhijie Lin, Yang Zhao, Kaixin Wang, Gao Huang, and Jiashi Feng. 2024. How Far is Video Generation from World Model? – A Physical Law Perspective. *arXiv preprint arXiv:2411.02385* (2024).
Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. 2023. Lerf: Language embedded radiance fields. In *Proceedings of the IEEE/CVF international conference on computer vision*. 19729–19739.
Dan Kondratyuk, Lijun Yu, Xiuye Gu, José Lezama, Jonathan Huang, Grant Schindler, Rachel Hornung, Vighnesh Birodkar, Jimmy Yan, Ming-Chang Chiu, et al. 2023. Videopoet: A large language model for zero-shot video generation. *arXiv preprint arXiv:2312.14125* (2023).
Long Le, Ryan Lucas, Chen Wang, Chuhao Chen, Dinesh Jayaraman, Eric Eaton, and Lingjie Liu. 2025. Pixie: Fast and Generalizable Supervised Learning of 3D Physics from Pixels. *arXiv preprint arXiv:2508.17437* (2025).
Chenyu Li, Oscar Michel, Xichen Pan, Sainan Liu, Mike Roberts, and Saining Xie. 2025. PISA Experiments: Exploring Physics Post-Training for Video Diffusion Models by Watching Stuff Drop. *arXiv preprint arXiv:2503.09595* (2025).
Xuan Li, Yi-Ling Qiao, Peter Yichen Chen, Krishna Murthy Jatavallabhula, Ming Lin, Chenfanfu Jiang, and Chuang Gan. 2023. Pac-nerf: Physics augmented continuum neural radiance fields for geometry-agnostic system identification. *arXiv preprint arXiv:2303.05512* (2023).
Yichao Liang, Nishanth Kumar, Hao Tang, Adrian Weller, Joshua B Tenenbaum, Tom Silver, João F Henriques, and Kevin Ellis. 2024. Visualpredicator: Learning abstract world models with neuro-symbolic predicates for robot planning. *arXiv preprint arXiv:2410.23156* (2024).
Shaowei Liu, Zhongzheng Ren, Saurabh Gupta, and Shenlong Wang. 2024. Physgen: Rigid-body physics-grounded image-to-video generation. In *European Conference on Computer Vision*. Springer, 360–378.
Utkarsh Mall, Cheng Perng Phoo, Mia Chiquier, Bharath Hariharan, Kavita Bala, and Carl Vondrick. 2025. DiSciPLE: Learning Interpretable Programs for Scientific Visual Discovery. In *Proceedings of the Computer Vision and Pattern Recognition Conference*. 29258–29267.
Simon McIntosh-Smith, Sadaf R Alam, and Christopher Woods. 2024. Isambard-AI: a leadership class supercomputer optimised specifically for Artificial Intelligence. arXiv:2410.11199 [cs.DC] https://arxiv.org/abs/2410.11199
Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. 2021. Nerf: Representing scenes as neural radiance fields for view synthesis. *Commun. ACM* 65, 1 (2021), 99–106.
Saman Motamed, Laura Culp, Kevin Swersky, Priyank Jaini, and Robert Geirhos. 2025. Do generative video models understand physical principles? *arXiv preprint arXiv:2501.09038* (2025).
NVIDIA, Arslan Ali, Junjie Bai, Maciej Bala, Yogesh Balaji, Aaron Blakeman, Tiffany Cai, Jiaxin Cao, Tianshi Cao, Elizabeth Cha, Yu-Wei Chao, Prithvijit Chattopadhyay, Mike Chen, Yongxin Chen, Yu Chen, Shuai Cheng, Yin Cui, Jenna Diamond, Yifan Ding, Jiaojiao Fan, Linxi Fan, Liang Feng, Francesco Ferroni, Sanja Fidler, Xiao Fu, Ruiyuan Gao, Yunhao Ge, Jinwei Gu, Aryaman Gupta, Siddharth Gururani, Imad El Hanafi, Ali Hassani, Zekun Hao, Jacob Huffman, Joel Jang, Pooya Jannaty, Jan Kautz, Grace Lam, Xuan Li, Zhaoshuo Li, Maosheng Liao, Chen-Hsuan Lin, Tsung-Yi Lin, Yen-Chen Lin, Huan Ling, Ming-Yu Liu, Xian Liu, Yifan Lu, Alice Luo, Qianli Ma, Hanzi Mao, Kaichun Mo, Seungjun Nah, Yashraj Narang, Abhijeet Panaskar, Lindsey Pavao, Trung Pham, Morteza Ramezanali, Fitsum Reda, Scott Reed, Xuanchi Ren, Haonan Shao, Yue Shen, Stella Shi, Shuran Song, Bartosz Stefaniak, Shangkun Sun, Shitao Tang, Sameena Tasmeen, Lyne Tchapmi, Wei-Cheng Tseng, Jibin Varghese, Andrew Z. Wang, Hao Wang, Haoxiang Wang, Heng Wang, Ting-Chun Wang, Fangyin Wei, Jiashu Xu, Dinghao Yang, Xiaodong Yang, Haotian Ye, Seonghyeon Ye, Xiaohui Zeng, Jing Zhang, Qinsheng Zhang, Kaiwen Zheng, Andrew Zhu, and Yuke Zhu. 2025. World Simulation with Video Foundation Models for Physical AI. *arXiv preprint arXiv:2511.00062* (2025).
OpenAI. 2024. Video generation models as world simulators. https://openai.com/index/sora/.
OpenAI. 2025. Sora 2 is here. https://openai.com/index/sora-2/.
Emanuel Parzen. 1962. On Estimation of a Probability Density Function and Mode. *The Annals of Mathematical Statistics* 33, 3 (1962), 1065 – 1076. doi:10.1214/aoms/1177704472
Automne Petitjean, Yohan Poirier-Ginter, Ayush Tewari, Guillaume Cordonnier, and George Drettakis. 2023. ModalNeRF: Neural Modal Analysis and Synthesis for Free-Viewpoint Navigation in Dynamically Vibrating Scenes. In *Computer Graphics Forum*, Vol. 42. Wiley Online Library, e14888.
Yu Qi, Xinyi Xu, Ziyu Guo, Siyuan Ma, Renrui Zhang, Xinyan Chen, Ruichuan An, Ruofan Xing, Jiayi Zhang, Haojie Huang, Pheng-Ann Heng, Jonathan Tremblay, and Lawson L. S. Wong. 2025. MME-CoF-Pro: Evaluating Reasoning Coherence of Video Generative Models. *arXiv preprint arXiv:2603.20194v1* (2025).
Murray Rosenblatt. 1956. Remarks on Some Nonparametric Estimates of a Density Function. *Annals of Mathematical Statistics* 27 (1956), 832–837. https://api.semanticscholar.org/CorpusID:16643156
Runway. 2025. Gen-4. https://runwayml.com/research/introducing-runway-gen-4.
Kiwhan Song, Boyuan Chen, Max Simchowitz, Yilun Du, Russ Tedrake, and Vincent Sitzmann. 2025. History-guided video diffusion. *arXiv preprint arXiv:2502.06764* (2025).
Dídac Surís, Sachit Menon, and Carl Vondrick. 2023. Vipergpt: Visual inference via python execution for reasoning. In *Proceedings of the IEEE/CVF international conference on computer vision*. 11888–11898.
Stanislaw Szymanowicz, Chrisitian Rupprecht, and Andrea Vedaldi. 2024. Splatter image: Ultra-fast single-view 3d reconstruction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 10208–10217.
Kling Team, Jialu Chen, Yikang Ding, Zhixue Fang, Kun Gai, Yuan Gao, Kang He, Jingyun Hua, Boyuan Jiang, Mingming Lao, et al. 2025b. Klingavatar 2.0 technical report. *arXiv preprint arXiv:2512.13313* (2025).
SAM 3D Team, Xingyu Chen, Fu-Jen Chu, Pierre Gleize, Kevin J Liang, Alexander Sax, Hao Tang, Weiyao Wang, Michelle Guo, Thibaut Hardin, Xiang Li, Aohan Lin, Jiawei Liu, Ziqi Ma, Anushka Sagar, Bowen Song, Xiaodong Wang, Jianing Yang, Bowen Zhang, Piotr Dollár, Georgia Gkioxari, Matt Feiszli, and Jitendra Malik. 2025a. SAM 3D: 3Dfy Anything in Images. (2025). arXiv:2511.16624 [cs.CV] https://arxiv.org/abs/2511.16624
Ayush Tewari, Tianwei Yin, George Cazenavette, Semon Rezchikov, Josh Tenenbaum, Frédo Durand, Bill Freeman, and Vincent Sitzmann. 2023. Diffusion with forward models: Solving stochastic inverse problems without direct supervision. *Advances in Neural Information Processing Systems* 36 (2023), 12349–12362.
Edgar Tretschk, Ayush Tewari, Vladislav Golyanik, Michael Zollhöfer, Christoph Lassner, and Christian Theobalt. 2021. Non-rigid neural radiance fields: Reconstruction and novel view synthesis of a dynamic scene from monocular video. In *Proceedings of the IEEE/CVF international conference on computer vision*. 12959–12970.
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. 2025. Wan: Open and Advanced Large-Scale Video Generative Models. *arXiv preprint arXiv:2503.20314* (2025).
Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. 2025a. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*. 5294–5306.
Ruicheng Wang, Sicheng Xu, Yue Dong, Yu Deng, Jianfeng Xiang, Zelong Lv, Guangzhong Sun, Xin Tong, and Jiaolong Yang. 2025b. MoGe-2: Accurate Monocular Geometry with Metric Scale and Sharp Details. arXiv:2507.02546 [cs.CV] https://arxiv.org/abs/2507.02546
Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. 2024. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 20697–20709.
Xiaofeng Wang, Zheng Zhu, Guan Huang, Xinze Chen, Jiagang Zhu, and Jiwen Lu. 2023. Drivedreamer: Towards real-world-driven world models for autonomous driving. *arXiv preprint arXiv:2309.09777* (2023).
Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 2024. 4D Gaussian Splatting for Real-Time Dynamic Scene Rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 20310–20320.
Jiajun Wu, Ilker Yildirim, Joseph J Lim, Bill Freeman, and Josh Tenenbaum. 2015. Galileo: Perceiving physical object properties by integrating a physics engine with deep learning. *Advances in neural information processing systems* 28 (2015).
Tianyi Xie, Zeshun Zong, Yuxing Qiu, Xuan Li, Yutao Feng, Yin Yang, and Chenfanfu Jiang. 2024. Physgaussian: Physics-integrated 3d gaussians for generative dynamics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4389–4398.

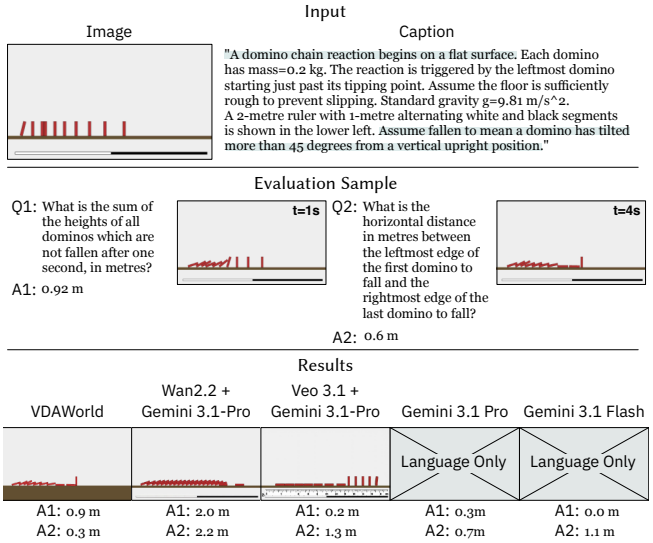

Fig. 11. A sample from our High School Physics Benchmark, HSPBench. VDAWorld produces a simulation which reproduces the physics of the original setting more accurately, enabling it to provide high-quality answers to the questions associated with the samples.

Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. 2023. Deformable 3D Gaussians for High-Fidelity Monocular Dynamic Scene Reconstruction. *arXiv preprint arXiv:2309.13101* (2023).

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

Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. 2021. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 4578–4587.

Raza Yunus, Jan Eric Lenssen, Michael Niemeyer, Yiyi Liao, Christian Rupprecht, Christian Theobalt, Gerard Pons-Moll, Jia-Bin Huang, Vladislav Golyanik, and Eddy Ilg. 2024. Recent Trends in 3D Reconstruction of General Non-Rigid Scenes. In *Computer Graphics Forum*, Vol. 43. Wiley Online Library, e15062.

Junyi Zhang, Charles Herrmann, Junhwa Hur, Varun Jampani, Trevor Darrell, Forrester Cole, Deqing Sun, and Ming-Hsuan Yang. 2024a. Monst3r: A simple approach for estimating geometry in the presence of motion. *arXiv preprint arXiv:2410.03825* (2024).

Tianyuan Zhang, Hong-Xing Yu, Rundi Wu, Brandon Y. Feng, Changxi Zheng, Noah Snavely, Jiajun Wu, and William T. Freeman. 2024b. PhysDreamer: Physics-Based Interaction with 3D Objects via Video Generation. In *European Conference on Computer Vision*. Springer.

Fangqi Zhu, Hongtao Wu, Song Guo, Yuxiao Liu, Chilam Cheang, and Tao Kong. 2024. IRASim: Learning Interactive Real-Robot Action Simulators. *arXiv:2406.12802* (2024).

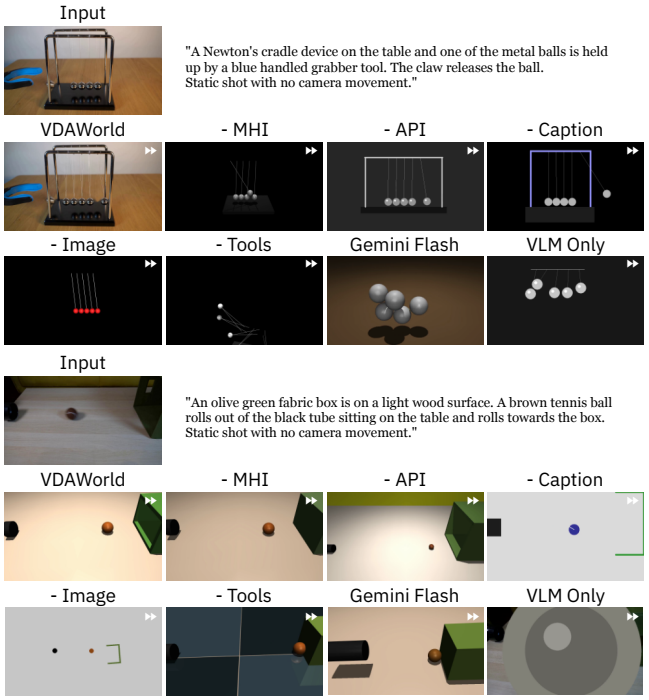

Fig. 10. Ablations of VDAWorld. Note that when the image is missing, the appearance of some objects is incorrectly constructed. Without the caption, objects often travel in the wrong direction (for example the ball rolls into the box, rather than behind it). Without the toolbox, objects cannot be accurately matched to their true dimensions. Finally, without the critic, errors in scene construction or dynamics often occur due to coding errors.