API WORLD MODEL: AGENTIC, PROGRAMMABLE, AND INTERACTIVE WORLD MODEL WITH AGENT-DRIVEN VIRTUAL-WORLD ROLLOUTS

Anonymous authors
Paper under double-blind review

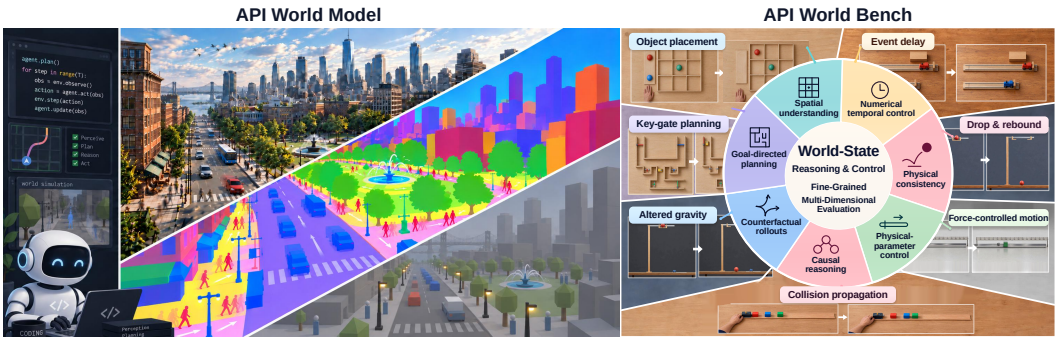


Figure 1: Overview of API World Model and API World Bench. Left: a coding agent constructs and evolves an executable world, whose semantic renderings guide realistic video generation. Right: API World Bench evaluates world models across diverse tasks involving world states, interactions, and physical dynamics.

ABSTRACT

World models are just at a tipping point where they could really take off, with the promise of enabling agents to understand, predict, and interact with the physical world. Recent efforts have largely pursued this vision by leveraging video generation models as world simulators to predict plausible visual observations. However, these models capture the world only through its visual projections, rather than the underlying principles that govern how it operates, evolves, and responds to interventions. This fundamental gap between visual plausibility and physical consistency limits their ability to reliably simulate how the world behaves. In this paper, we introduce *API World Model*, a new agentic paradigm that grounds world modeling in executable programs and controllable virtual environments. Given a natural-language instruction, we first employ a coding agent to construct an interactive world in a virtual engine, where scene geometry, object states, interactions, and physical dynamics are explicitly instantiated and governed by executable programs. The world is then rendered into semantic videos and translated into photorealistic observations by a tailored video diffusion model, bridging the gap between executable physical simulation and realistic visual observations. To further support long-horizon world evolution, we devise a structured memory mechanism that maintains and updates the relevant world state across time, enabling persistent consistency across extended interactions. Moreover, we develop an agentic pipeline to construct a comprehensive benchmark, *API World Bench*, for evaluating world models across diverse dimensions, including spatiotemporal consistency, physical controllability, causal reasoning, and counterfactual prediction. Extensive experiments show that *API World Model* enables realistic, physically consistent, and controllable world rollouts, providing a promising path toward world models that capture not only how the world looks, but also how it works.

1 INTRODUCTION

World models have emerged as a promising path toward Artificial General Intelligence (AGI). At their core, they provide agents with an internal simulator of the physical world, enabling them to

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

understand how the environment is structured, predict how it evolves, anticipate the consequences of their actions, and reason about alternative outcomes under different interventions Craik (1967); Ha & Schmidhuber (2018a). Such capabilities have far-reaching implications for a broad range of applications, including robotics, autonomous systems, and scientific discovery, where modeling and reasoning about complex environments are essential.

Spurred by rapid progress in generative modeling, a dominant line of work pursues world models by scaling video generation models into world simulators: trained on large-scale internet video data, these models learn to roll out plausible visual futures conditioned on past frames, text prompts, or actions MiniMax (2026); Wan et al. (2025); Kong et al. (2024). However, *videos are low-dimensional visual projections of the physical world*. Video generators solely learn the statistical regularities of *how the world looks*, not the underlying principles that govern *how the world works*. As a consequence, purely video-centric world models can't guarantee that the underlying world state, physical dynamics, or causal structure is correctly modeled. For example, a model may generate a visually convincing object trajectory while violating basic physical constraints, maintain inconsistent object states across long-horizon interactions, or fail to produce the expected consequences when an agent intervenes in the environment.

This limitation becomes particularly important when world models are used not merely as passive predictors, but as interactive simulators for agents. An agent operating in the real world must reason about consequences of its actions, test alternative strategies, and maintain a consistent understanding of the environment over extended time horizons. Such capabilities require a world model to answer questions that go beyond visual prediction: What is the state of the world? What will happen if an object is moved? Which changes are caused by an intervention? What would have happened under a different action? These questions require the model to represent a structured world whose state can be explicitly modified and whose evolution follows executable rules. In this view, a reliable world model should not only generate observations, but also expose a controllable mechanism through which the world can be instantiated, evolved, and queried.

These insights point naturally toward executable virtual worlds as a substrate for such world models. Modern virtual engines already offer programmatic abstractions for scene geometry, object states, interaction logic, and physical simulation. More importantly, recent advances in coding agents Achiam et al. (2023) make these capabilities directly accessible through natural language: coding agents can translate high-level instructions into executable programs and invoke the APIs of virtual engines to construct, modify, and control virtual environments. This creates a practical bridge between language-level descriptions of a world and its executable realization. Therefore, instead of asking a generative model to implicitly encode all these mechanisms in its visual space, we can delegate the construction and evolution of the world to executable programs, which naturally support intervention, state inspection, deterministic or rule-governed evolution, and counterfactual rollouts.

In this work, we introduce API World Model, a new agentic paradigm that grounds world modeling in executable programs and controllable virtual environments. The central idea is to use a coding agent as a bridge between high-level language and executable virtual environments. Given a natural-language instruction, the coding agent constructs a complete interactive world in a virtual engine, instantiating scene geometry, object properties, interaction rules, and physical dynamics as executable programs. While virtual environments can provide precise physical and causal structure, their rendered observations may differ substantially from the visual distribution encountered by real-world agents. To reconcile physical fidelity with visual realism, we separate world evolution from visual realization. The virtual engine first produces semantic videos that faithfully reflect the simulated state and dynamics. A tailored video diffusion model then translates these semantic renderings into photorealistic observations while preserving the relevant temporal and structural information. This decomposition allows the executable environment to serve as the source of physical and causal consistency, while generative modeling provides the flexibility and realism required for visual observations. In this way, instead of requiring one model to simultaneously learn what the world looks like and how it behaves, we explicitly factor these two roles into a controllable simulation module and a visual generation module. Then, we introduce a structured memory mechanism to maintain world consistency over long horizons. The key enabler is the semantic video representation, which provides an explicit correspondence between visual observations and the underlying world state. By leveraging this intermediate representation, our memory can associate the appearance of objects in generated photorealistic videos with their corresponding entities in the executable world, allowing object identities and states to be consistently tracked across time.

Existing evaluations of generative world models often emphasize visual quality, temporal consistency, or similarity to observed trajectories. While these metrics are useful, they do not directly measure whether a model understands the mechanisms governing a physical world. To this end, we develop API World Bench, an agentically constructed benchmark designed to evaluate world models from the perspective of world behavior rather than visual appearance alone. The benchmark covers multiple dimensions of interactive world modeling, including spatiotemporal consistency, physical controllability, causal reasoning, and counterfactual prediction. By explicitly testing how a world responds to actions and hypothetical interventions, API World Bench provides a complementary evaluation axis to conventional video-generation metrics.

In summary, we make the following contributions: (1) ***API World Model.*** We introduce a new agentic paradigm that constructs executable and controllable world models through coding agents and virtual engines. (2) ***Structured World Memory.*** We develop a structured memory mechanism that grounds visual observations to their corresponding entities in the executable world, maintaining world consistency over long-horizon interactions. (3) ***API World Bench.*** We present an agentically constructed benchmark for evaluating world models beyond visual fidelity, covering spatiotemporal consistency, physical controllability, causal reasoning, and counterfactual prediction.

2 Related Work

World Models. World models aim to learn environment dynamics to support prediction, planning, and policy learning. Early methods (Ha & Schmidhuber, 2018b; Micheli et al., 2022) predict action-conditioned transitions in a compact representation space, allowing agents to learn from imagined experience. GameGAN (Kim et al., 2020) learns to simulate game environments directly from visual observations and keyboard actions. With advances in video generation, subsequent work extends learned simulation to more diverse environments and control conditions. UniSim (Yang et al., 2023) learns an interactive simulator from heterogeneous real-world data, while Genie (Bruce et al., 2024) learns latent actions from unlabeled videos for controllable environment generation. iVideoGPT (Wu et al., 2024) models observations, actions, and rewards as token sequences for video prediction and visual planning. Diffusion-based methods further improve visual simulation. DIAMOND (Alonso et al., 2024) studies diffusion world models for reinforcement learning, and GameNGen (Valevski et al., 2025) demonstrates real-time neural game simulation. To broaden the interaction interface, Pandora (Xiang et al., 2024) combines a language model with a video model to support natural-language actions, and Matrix-Game 2.0 (He et al., 2025) enables streaming generation conditioned on frame-level keyboard and mouse inputs. Another line of work addresses scene consistency over extended interactions. WorldMem (Xiao et al., 2026) retrieves historical observations using poses and timestamps, Context-as-Memory (Yu et al., 2025) selects past frames according to field-of-view overlap, and WorldPlay (Sun et al., 2025) combines historical context reconstruction with memory-aware distillation. These methods improve interactive visual prediction and the reuse of past observations.

LLM Agents. Large language models have been used to translate natural-language instructions into actions through planning, tool use, and program generation. These capabilities enable LLM-based agents to interact with external environments and accomplish increasingly complex tasks in a flexible and autonomous manner. SayCan (Ahn et al., 2022) combines language-based task planning with robotic affordances to select executable skills, while ReAct (Yao et al., 2022) interleaves reasoning and actions with environment feedback. Programs provide a flexible action representation for composing operations and expressing control logic. Code as Policies (Liang et al., 2023) generates robot policies that connect perception outputs with control APIs, and CodeAct (Wang et al., 2024b) uses executable Python actions and execution feedback for multi-step tasks. Voyager (Wang et al., 2023) extends code-based interaction to continual exploration in Minecraft through an expanding library of reusable skills. Beyond acting in existing environments, LLMs can also construct scenes and simulation tasks. GenSim (Wang et al., 2024a) generates robotic simulation programs to increase task diversity. Holodeck (Yang et al., 2024) selects and arranges 3D assets according to language-derived spatial constraints, and Holodeck 2.0 (Bian et al., 2025) incorporates visual guidance and iterative scene editing. A related direction represents world dynamics as programs: WorldCoder (Tang et al., 2024) builds transition and reward models through code generation and environment interaction for planning. More recently, LingBot-World 2.0 (Gao et al., 2026) introduces pilot and director agents for character control and environmental interventions in video generation.

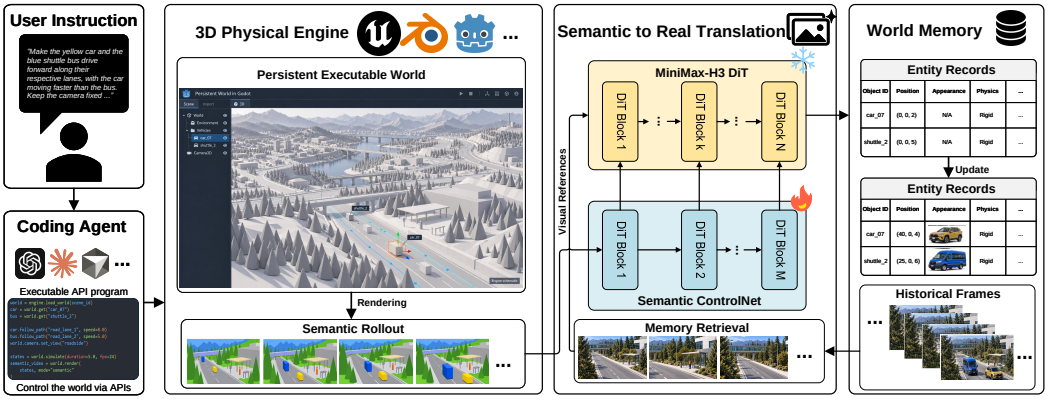


Figure 2: An overview of our proposed API World Model.

3 API World Model and API World Bench

We present API World Model, an agentic framework that enables agents to construct, manipulate, and evolve a persistent world through executable programs and structured APIs. In addition, we present API World Bench, an agentically constructed benchmark designed to evaluate world models from the perspective of world behavior. Figure 2 illustrates the overall framework.

3.1 Agent-driven Virtual-world Simulation

Unlike video-based world models that directly predict future observations, our goal is to allow an agent to explicitly construct a world and control how it evolves through interaction. To this end, we employ a powerful coding agent to construct and operate a persistent 3D world using virtual engines, such as Godot Engine Godot (2024).

Virtual World Building. Given a natural language description q_1, we employ a coding agent to generate executable programs P_1, which can invoke virtual engines to construct coarse scene geometry, create and arrange entities, and configure their physical properties, states, and behavior scripts. Then, we execute P^{init} to obtain the virtual world that fully determines a dynamic 3D scene described in the input natural language description:

$$W_1 = \text{Execute}(W_0, P_1), \tag{1}$$

where W_0 is the default empty engine world. The constructed world W_1 can be viewed as a sequence of world states. Each state fully specifies the configuration of the world at time step t, including the entities present in the scene, their geometry and spatial arrangement, physical properties, internal states, and behavioral states. The transition between consecutive world states is governed by the physical dynamics and executable behavior scripts. Thus, the dynamic 3D world is represented as a temporally evolving sequence of executable world states.

Continuous World Evolution. After the initial world is constructed, users can continuously modify and interact with the maintained world through natural language instructions. At interaction n, given the current world W_n and a user instruction q_n, the coding agent generates an executable program P_n that implements the requested world changes. The program invokes engine APIs to access entities through their persistent identifiers, modify their attributes and relations, and configure their behavior scripts. The engine executes P_n based on the current world state, during which the specified actions and interactions are carried out according to the engine scripts and physical simulation. After the execution, the resulting world is retained as

$$W_{n+1} = \text{Execute}(W_n, P_n), \tag{2}$$

where W_{n+1} denotes the evolved world with the interactive instruction q_n.

3.2 Semantic-guided World Rendering

Constructing a photorealistic scene directly in the 3D engine would require detailed assets, materials, and lighting for every interaction, which is expensive and difficult to scale while still suffering

from an inevitable gap between synthetic renderings and real-world appearance. To this end, we use a video diffusion model as a generative renderer of the maintained virtual world. Specifically, the 3D engine determines the world evolution, while the video model synthesizes realistic visual observations conditioned on semantic renderings of the evolving world.

Semantic Rollout. Since the world is explicitly maintained in the 3D engine, we can observe its evolving state from a desired viewpoint at any sampled time during execution. Let $W_n(\tau)$ denote the world state at time τ during interaction n, and let $C_n(\tau) = (K_n(\tau), T_n(\tau))$ denote the corresponding observation camera, where $K_n(\tau)$ and $T_n(\tau)$ represent the camera intrinsics and pose, respectively. Given the world state $W_n(\tau)$ and camera $C_n(\tau)$, the engine renders a semantic observation as

$$S_n(\tau) = \mathrm{Render}_{\mathrm{semantic}}\left(W_n(\tau), C_n(\tau)\right). \tag{3}$$

By observing the evolving world at a sequence of sampled times $\{\tau_l\}_{l=1}^{L}$, we obtain a temporally ordered semantic rollout:

$$\mathbf{S_n} = \{S_n(\tau_l)\}_{l=1}^{L}. \tag{4}$$

Each semantic observation assigns category-level labels to the visible scene entities, thereby preserving their spatial layout, occlusions, and temporal motion throughout the rollout. Entities belonging to the same category share the same semantic label, while their persistent instance identities are maintained separately in the engine.

Semantic to Real Translation. Given the semantic rollout $\mathbf{S}_n$, our goal is to translate it into a photorealistic video $\mathbf{V}_n$ that follows the spatial configuration and temporal evolution specified by the engine world. We build this upon MiniMax H3 (MiniMax, 2026) and introduce an additional ControlNet branch (Zhang et al., 2023) to incorporate the semantic rollout as a temporal control signal. The conditional rendering process at interaction n is formulated as

$$\mathbf{V}_n \sim p_{\theta,\phi}\left(\mathbf{V}_n \mid \mathbf{S}_n, \mathcal{T}_n, \mathcal{B}_n, p_n\right), \tag{5}$$

where θ and ϕ denote the parameters of the MiniMax H3 backbone and the semantic ControlNet branch, respectively. $\mathcal{T}_n$ denotes the object appearance references, $\mathcal{B}_n$ denotes the background references, and p_n denotes the rendering prompt. The reference sets $\mathcal{T}_n$ and $\mathcal{B}_n$ are retrieved from World Memory according to the current world state and observation, as described in the following Sec. 3.3.

The semantic rollout $\mathbf{S}_n$ provides the spatial and temporal structure of the target video, while $\mathcal{T}_n$ and $\mathcal{B}_n$ provide appearance information accumulated from previous observations. Based on the current world evolution and the retrieved appearance references, the agent further generates and refines a rendering prompt p_n to describe the target scene and associate the references with the corresponding entities according to their categories, positions, and actions. These conditions are incorporated into the renderer through complementary conditioning pathways: $\mathcal{T}_n$ and $\mathcal{B}_n$ follow the original reference-conditioning pathway of MiniMax H3, p_n is introduced through its text-conditioning pathway, and $\mathbf{S}_n$ is injected through the additional ControlNet branch for semantic control.

3.3 World Memory

The virtual engine maintains the geometry, states, and behaviors of scene entities across interactions, while their visual appearance is synthesized by the video model. Generating each video independently may therefore produce inconsistent appearances for the same object or previously observed background regions. To address this issue, we introduce World Memory to retain visual observations and update them as the world evolves. After interaction n, World Memory is represented as

$$\mathcal{M}_n = \left(\mathcal{M}_n^{\mathrm{obj}}, \mathcal{M}_n^{\mathrm{bg}}\right), \tag{6}$$

where $\mathcal{M}_n^{\mathrm{obj}}$ stores object appearance references indexed by persistent entity IDs, together with records of changes to their appearance and visible states. The background memory $\mathcal{M}_n^{\mathrm{bg}}$ stores historical frames together with their camera parameters and records when each frame was generated. Before generating the video for interaction $n+1$, we retrieve object references $\mathcal{T}_{n+1}$ and background references $\mathcal{B}_{n+1}$ that match the current world state and cover the scene content visible in the upcoming camera views.

Object Appearance Memory. After generating $\mathbf{V}_n$, we extract object appearance references and associate them with the persistent entity IDs in the engine world. Specifically, we use the object locations and IDs provided by the engine to guide segmentation in the generated frames and associate

each segmented object with its corresponding entity. For each entity i, the extracted observations form $\mathcal{O}_{n,i}$, and the agent updates its appearance memory as

$$\mathcal{M}^{\text{obj}}_{n,i} = \text{Select}\left(\mathcal{M}^{\text{obj}}_{n-1,i} \cup \mathcal{O}_{n,i}\right). \tag{7}$$

The agent uses the recorded changes to the entity to discard references that no longer match its appearance or visible state. Among the remaining observations, it retains at most four reference images, prioritizing clear views with limited occlusion and complementary viewing directions.

Before generating $\mathbf{V}_{n+1}$, we identify the moving entities visible along the upcoming camera trajectory. The agent retrieves their appearance references through persistent IDs and selects those relevant to the current interaction. These references form the object appearance condition $\mathcal{T}_{n+1}$, with their correspondence to the objects in the semantic video specified through the generation prompt.

Background Memory Retrieval. To preserve background appearance across interactions, we retrieve historical frames that observe scene regions visible along the upcoming camera trajectory. Following Context-as-Memory (Yu et al., 2025), we use the stored camera parameters to filter historical frames according to field-of-view overlap with the upcoming views.

Among the candidate frames, we favor recent observations when multiple frames cover similar views, allowing the retrieved references to better reflect the current scene state. We further reduce redundancy by selecting representative frames from groups of consecutive candidates. The selected frames form the background appearance condition $\mathcal{B}_{n+1}$ for video generation.

World Memory is initially empty, and the first video is generated without historical references. After each interaction, we update the object appearance references and store the generated frames with their camera parameters and generation times for subsequent retrieval.

3.4 API World Bench

Evaluating world models actually requires more than visual plausibility: generated videos must faithfully reflect the instructed state changes and interactions, including their timing, physical consequences, and execution order. We therefore introduce API World Bench, which evaluates instruction-driven world evolution through semantic verification of actions and interactions, alongside quantitative measures of spatial, temporal, and physical consistency.

Task design. We follow three principles in designing the tasks. *Controlled variation.* We vary the requested actions or parameters while keeping the initial image fixed, enabling comparisons of how generated videos respond to different instructions. *Process-level evaluation.* We assess both the final state and the actions leading to it, including the order, timing, and prerequisites specified in the instruction. *Observable outcomes.* Each task is designed to yield outcomes that are directly measurable from the generated videos. Quantitative tasks specify explicit spatial references and temporal constraints to support precise evaluation. In summary, API World Bench comprises 38 task families across seven categories. Each family includes diverse laboratory and everyday-object scenes, with multiple instruction variants per scene, yielding a total of 1064 evaluation episodes.

Benchmark construction. We construct each scene in Blender with task objects, interaction mechanisms, and visible geometric references. We render preview videos in Blender to check task feasibility and the required interactions. A generative image model then converts the initial frame into a realistic image, which is checked for consistency in object identities, dimensions, spatial relations, occlusions, and camera viewpoint. Each evaluation episode pairs this image with a text prompt specifying the requested actions and constraints.

Evaluation protocol. We evaluate generated videos through semantic verification and quantitative measurement. A vision–language model checks the required actions, object relations, and interaction conditions throughout the video. Numerical measurements are performed only when their semantic prerequisites are satisfied; a failed prerequisite assigns zero to the affected component. We use segmentation and tracking, together with first-frame calibration and original video timestamps, to measure spatial configurations, motion, and event timing. Selected physical tasks also compare matched variants to assess responses to changes in the specified parameters. Numerical errors are converted into scores in $[0, 100]$ using task-specific tolerances and combined with semantic checks. We average episode scores within each task family, family scores within each category, and the seven category scores equally to obtain the overall score.

Table 1: Performance comparisons with state-of-the-art video-based world models on API World Bench. There are 7 dimensions in total, including Spatial Understanding (SU), Temporal Controlling (TC), Physical Factuality (PF), Causal Reasoning (CR), Planning (PL), Cross-View Consistency (CC), and Counterfactual Prediction (CP). Higher values indicate better performance for all metrics.

Model	SU	TC	PF	CR	PL	CC	CP	Overall
Proprietary Models								
Kling 3.0 Kuaishou (2025)	37.8	31.8	16.4	26.0	11.4	11.5	17.6	21.8
Wan 3.0 Alibaba (2025)	36.4	30.2	15.6	24.5	11.2	11.0	17.8	21.0
Veo 3.1 DeepMind (2025)	38.1	31.2	16.0	25.5	11.6	11.2	17.9	21.6
Seedance 2.5 ByteDance (2025)	40.3	47.2	26.5	40.4	11.8	13.3	40.0	31.4
Open-sourced Models								
HunyuanVideo-13B Kong et al. (2024)	31.8	27.6	12.9	21.7	10.3	9.4	15.4	18.4
LTX-Video HaCohen et al. (2026)	33.6	29.1	14.3	22.9	10.9	10.2	16.7	19.7
Wan2.2-14B (Wan et al., 2025)	35.2	30.0	15.1	24.1	11.3	10.7	17.5	20.6
MiniMax-H3 (MiniMax, 2026)	40.7	27.1	6.4	16.8	26.4	12.9	30.6	23.0
API World Model (Ours)	**83.3**	**80.8**	**83.5**	**96.0**	**65.0**	**60.4**	**78.9**	**78.3**

4 Experiments

4.1 Experimental Settings

Baselines and Evaluation. We evaluate our method on the API World Bench, which contains seven task categories focusing on the model's ability to spatiotemporal consistency, physical controllability, causal reasoning, and counterfactual prediction. We compare against a diverse set of video generation models, including open-source models such as MiniMax-H3 (MiniMax, 2026), Wan2.2-14B (Wan et al., 2025), LTX-Video HaCohen et al. (2026), and HunyuanVideo-13B Kong et al. (2024), as well as proprietary models such as Seedance 2.5 ByteDance (2025), Veo 3.1 DeepMind (2025), Wan 3.0 Alibaba (2025), and Kling 3.0 Kuaishou (2025). Each method receives the same initial image and instruction to generate a continuous video depicting the requested world evolution. Evaluation follows the protocol in Sec. 3.4, combining task-specific semantic checks with quantitative measurements of spatial, temporal, and physical constraints.

Implementation Details. We use GPT-6 OpenAI (2026) as the coding agent to construct and operate the virtual world in Godot Godot (2024), and initialize the visual renderer from pretrained MiniMax-H3 Ref2VA (MiniMax, 2026). During training, we optimize only the ControlNet branch while keeping all other model parameters frozen. We construct approximately 200,000 semantic-to-real video pairs from SpatialVID (Wang et al., 2026) and Sekai Li et al. (2026b) datasets, including both static and dynamic scenes. We use Qwen3-VL-32B-Instruct (Bai et al., 2025) to generate the accompanying prompts following the prompt format of MiniMax-H3. We use a learning rate of 1×10^{-5} and train on 124-frame clips at a resolution of 832×480.

4.2 Quantitative Comparison

Table 1 summarizes the quantitative comparison results on API World Bench. Overall, API World Model achieves the best performance across all seven task categories, obtaining an overall score of 78.3 compared with 23.0 for the strongest open-source baseline, MiniMax-H3. In particular, our method improves the scores for causal reasoning and physical factuality from 16.8 to 96.0 and from 6.4 to 83.5, respectively. These results demonstrate the effectiveness of explicitly executing world dynamics and using the resulting semantic videos to guide visual generation. Specifically, the agent translates instructions into executable programs that control object states and interactions in the engine. The resulting semantic videos provide explicit spatial and temporal guidance, reducing the need for the video model to infer these dynamics from text alone. This decoupled way of modeling world dynamics and visual generation enables more accurate and controllable world modeling. This advantage is particularly evident in tasks that require reasoning about physical causality and object interactions. By executing the underlying dynamics in a virtual engine, our method can explicitly simulate how changes in one object or action affect subsequent states, rather than relying solely on the video model to infer such relationships from textual descriptions.

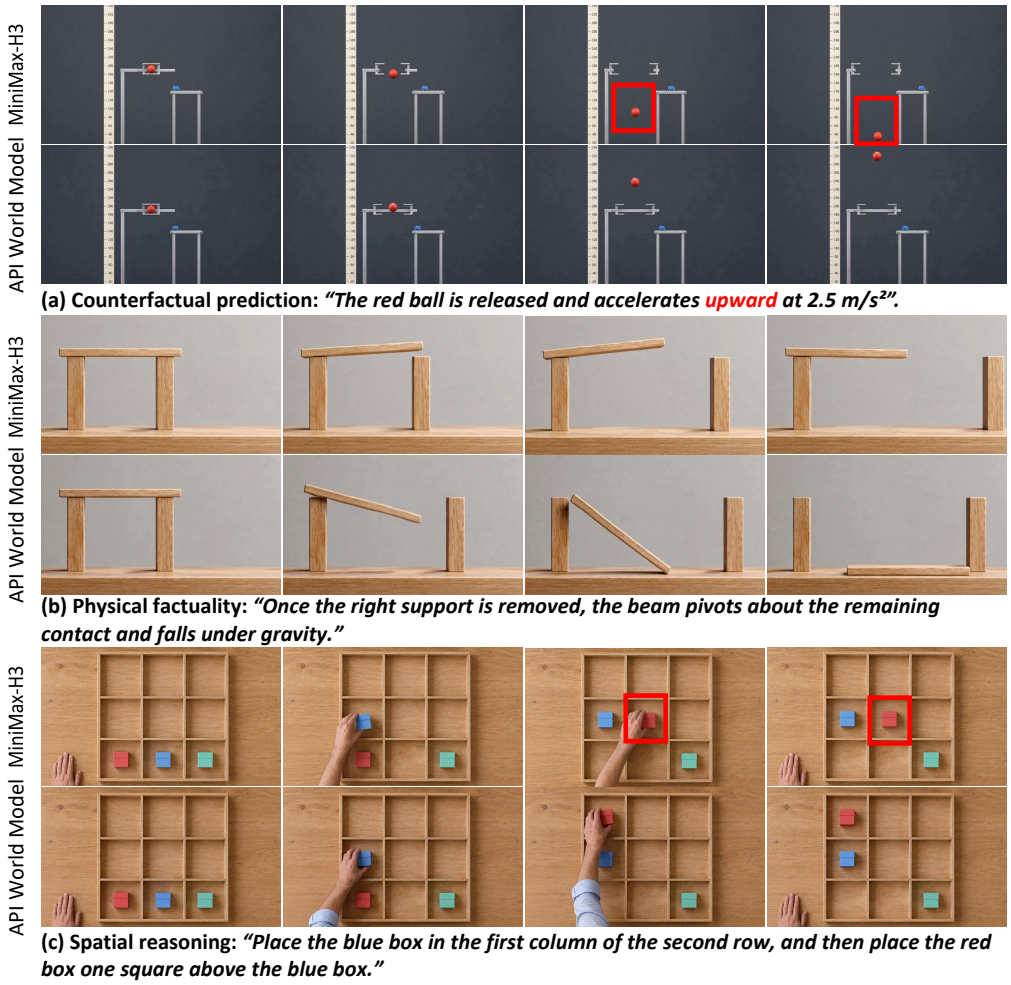


Figure 3: Qualitative comparisons with MiniMax-H3 on API World Bench, covering (a) counterfactual prediction, (b) physical factuality, and (c) spatial reasoning.

4.3 Qualitative Comparison

Figure 3 (a-c) presents qualitative comparisons with MiniMax-H3 (MiniMax, 2026) across three representative tasks. For counterfactual prediction, our method follows the specified *upward* acceleration, while MiniMax-H3 generates a falling trajectory (a). We examine physical factuality by removing the right support: our method shows the beam rotating downward and falling, while the baseline leaves it suspended (b). For spatial reasoning, our method places the red box above the blue box as instructed, whereas MiniMax-H3 places them side by side (c).These examples highlight the advantage of explicitly grounding video generation in executable world dynamics. By simulating object states and interactions in the virtual engine, our method can faithfully preserve the specified physical consequences and spatial relationships, whereas the baseline often produces visually plausible but physically or spatially inconsistent results.

Moreover, we test the model's ability to evolve the world state beyond its current visual observation. Specifically, we examine whether the model can correctly predict how objects evolve when they are temporarily out of sight. A world model should maintain a consistent internal representation of the underlying world dynamics, such that the state continues to evolve correctly even in the absence of direct visual observation. As shown in Figure 4, the person and chair move to the right while the camera looks left. Then, the camera returns to the scene. Our API World Model results demonstrate that the person and chair have continued to move to the right during the period of unobservability. However, Video-based world models simply update world states based on the most recently observed frames, failing to account for the evolution of the underlying world.

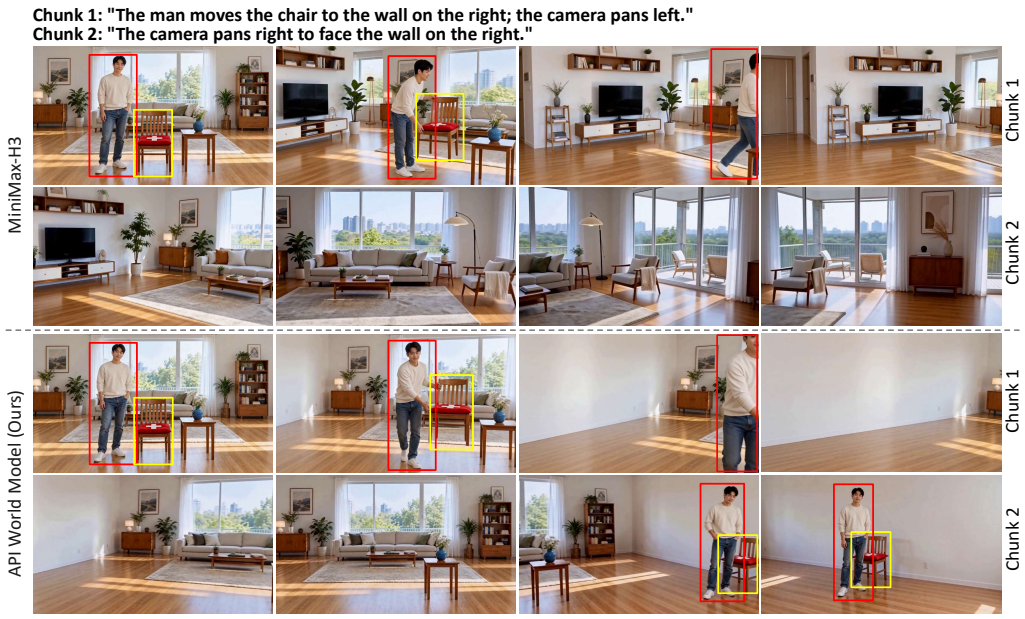


Figure 4: API World Model maintains an executable world whose entity states continue to evolve regardless of whether we observe it or not.



Figure 5: Qualitative ablation of World Memory. The comparison is designed to use the same world evolution and semantic control with and without historical object and background references.

4.4 Ablation Study

We design a qualitative ablation comparing the full model with a variant that removes World Memory. In the ablated variant, neither the object appearance references $\mathcal{T}_n$ nor the historical background references $\mathcal{B}_n$ are supplied to the video renderer. The comparison keeps the initial input, engine state sequence, camera trajectory, and semantic control fixed, so that it examines the effect of historical appearance references under the same world evolution. As shown in Figure 5, removing World Memory leads to noticeable inconsistencies in object appearance across frames, whereas incorporating historical appearance references helps preserve consistent object identities and visual attributes throughout the world evolution.

5 Conclusion

We presented API World Model, an agentic framework that separates world evolution from visual generation. A coding agent constructs and operates a persistent virtual world, while a tailored video diffusion model generates realistic observations conditioned on semantic videos rendered from the engine. We devise a World Memory mechanism to maintain object and background appearance across successive interactions. We also introduced API World Bench to evaluate instruction-driven world evolution. Experiments demonstrate improved adherence to specified interactions and spatial, temporal, and physical constraints, supporting agent-driven world evolution as a promising direction for programmable and interactive world models.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022.

Alibaba. Wan 3.0, 2025. URL `https://create.wan.video/explore`.

Eloi Alonso, Adam Jelley, Vincent Micheli, Anssi Kanervisto, Amos Storkey, Tim Pearce, and François Fleuret. Diffusion for world modeling: Visual details matter in atari. In *NeurIPS*, 2024.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Zixuan Bian, Ruohan Ren, Yue Yang, and Chris Callison-Burch. Holodeck 2.0: Vision-language-guided 3d world generation with editing. *arXiv preprint arXiv:2508.05899*, 2025.

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

ByteDance. Seedance 2.5, 2025. URL `https://seed.bytedance.com/seedance2_5`.

Kunlin Cai, Rui Song, Jinghuai Zhang, Kaiyuan Zhang, Pranav Bodapati, Alicia Yu, Fnu Suya, Mohammad Rostami, Jiaqi Ma, and Yuan Tian. What-if world: A causal benchmark for general world models in embodied scenarios. *arXiv preprint arXiv:2605.27589*, 2026.

Harold Haodong Chen, Disen Lan, Wen-Jie Shu, Qingyang Liu, Zihan Wang, Sirui Chen, Wenkai Cheng, Kanghao Chen, Hongfei Zhang, Zixin Zhang, et al. Tivibench: Benchmarking think-in-video reasoning for video generative models. *arXiv preprint arXiv:2511.13704*, 2025.

Kenneth James Williams Craik. *The nature of explanation*. CUP Archive, 1967.

Google DeepMind. Veo 3, 2025. URL `https://deepmind.google/models/veo/`.

Haoyi Duan, Hong-Xing Yu, Sirui Chen, Li Fei-Fei, and Jiajun Wu. Worldscore: A unified evaluation benchmark for world generation. In *ICCV*, 2025.

Weixi Feng, Jiachen Li, Michael Saxon, Tsu-jui Fu, Wenhu Chen, and William Yang Wang. Tc-bench: Benchmarking temporal compositionality in text-to-video and image-to-video generation. *arXiv preprint arXiv:2406.08656*, 2024.

Zelin Gao, Qiuyu Wang, Jiapeng Zhu, Jingye Chen, Zichen Liu, Qingyan Bai, Jiahao Wang, Yufeng Yuan, Hanlin Wang, Yichong Lu, et al. Infinite worlds with versatile interactions. *arXiv preprint arXiv:2607.07534*, 2026.

Godot. Godot engine. `https://godotengine.org/`, 2024.

Jing Gu, Xian Liu, Yu Zeng, Ashwin Nagarajan, Fangrui Zhu, Daniel Hong, Yue Fan, Qianqi Yan, Kaiwen Zhou, Ming-Yu Liu, et al. $phyworldbench$: A comprehensive evaluation of physical realism in text-to-video models. In *ICLR*, 2026.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In *NeurIPS*, 2018a.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018b.

Yoav HaCohen, Benny Brazowski, Nisan Chiprut, Yaki Bitterman, Andrew Kvochko, Avishai Berkowitz, Daniel Shalem, Daphna Lifschitz, Dudu Moshe, Eitan Porat, et al. Ltx-2: Efficient joint audio-visual foundation model. *arXiv preprint arXiv:2601.03233*, 2026.

Xianglong He, Chunli Peng, Zexiang Liu, Boyang Wang, Yifan Zhang, Qi Cui, Fei Kang, Biao Jiang, Mengyin An, Yangyang Ren, et al. Matrix-game 2.0: An open-source real-time and streaming interactive world model. *arXiv preprint arXiv:2508.13009*, 2025.

Ziqi Huang, Fan Zhang, Xiaojie Xu, Yinan He, Jiashuo Yu, Ziyue Dong, Qianli Ma, Nattapol Chanpaisit, Chenyang Si, Yuming Jiang, et al. Vbench++: Comprehensive and versatile benchmark suite for video generative models. *TPAMI*, 2025.

Rahul Jain, Mayank Patel, Asim Unmesh, and Karthik Ramani. Mechverse: Evaluating physical motion consistency in video generation models. *arXiv preprint arXiv:2605.14843*, 2026.

Seung Wook Kim, Yuhao Zhou, Jonah Philion, Antonio Torralba, and Sanja Fidler. Learning to simulate dynamic environments with gamegan. In *CVPR*, 2020.

Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. Hunyuanvideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024.

Kuaishou. Kling video, 2025. URL `https://kling.ai/app`.

Dacheng Li, Yunhao Fang, Yukang Chen, Shuo Yang, Shiyi Cao, Justin Wong, Michael Luo, Xiaolong Wang, Hongxu Yin, Joseph Gonzalez, et al. Worldmodelbench: Judging video generation models as world models. In *NeurIPs*, 2026a.

Zhen Li, Chuanhao Li, Xiaofeng Mao, Shaoheng Lin, Ming Li, Shitian Zhao, Zhaopan Xu, Xinyue Li, Yukang Feng, Jianwen Sun, et al. Sekai: A video dataset towards world exploration. In *NeurIPs*, 2026b.

Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. Code as policies: Language model programs for embodied control. In *ICRA*, 2023.

Vincent Micheli, Eloi Alonso, and François Fleuret. Transformers are sample-efficient world models. *arXiv preprint arXiv:2209.00588*, 2022.

MiniMax. MiniMax H3: An Open Model Breaking the Boundaries Between Tasks and Modalities. MiniMax Research blog, July 2026. URL `https://www.minimax.io/blog/minimax-h3`.

Saman Motamed, Laura Culp, Kevin Swersky, Priyank Jaini, and Robert Geirhos. Do generative video models understand physical principles? In *WACV*, 2026.

OpenAI. Gpt-6, 2026. URL `https://openai.com/`.

Wenqiang Sun, Haiyu Zhang, Haoyuan Wang, Junta Wu, Zehan Wang, Zhenwei Wang, Yunhong Wang, Jun Zhang, Tengfei Wang, and Chunchao Guo. Worldplay: Towards long-term geometric consistency for real-time interactive world modeling. *arXiv preprint arXiv:2512.14614*, 2025.

Hao Tang, Darren Key, and Kevin Ellis. Worldcoder, a model-based llm agent: Building world models by writing code and interacting with the environment. 2024.

Antonios Tragoudaras, Chenyu Zhang, Daniil Cherniavskii, Antonios Vozikis, Thijmen Nijdam, Derck WE Prinzhorn, Mark Bodracska, Nicu Sebe, Andrii Zadaianchuk, and Efstratios Gavves. Evaluating newtonian mechanics in video generative models with real physical systems. *arXiv preprint arXiv:2504.02918*, 2025.

Dani Valevski, Yaniv Leviathan, Moab Arar, and Shlomi Fruchter. Diffusion models are real-time game engines. In *ICLR*, volume 2025, 2025.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023.

Jiahao Wang, Yufeng Yuan, Rujie Zheng, Youtian Lin, Jian Gao, Lin-Zhuo Chen, Yajie Bao, Chang Zeng, Yanxi Zhou, Xiao-Xiao Long, et al. Spatialvid: A large-scale video dataset with spatial annotations. In *CVPR*, 2026.

Lirui Wang, Yiyang Ling, Zhecheng Yuan, Mohit Shridhar, Chen Bao, Yuzhe Qin, Bailin Wang, Huazhe Xu, and Xiaolong Wang. Gensim: Generating robotic simulation tasks via large language models. In *ICLR*, 2024a.

Xingyao Wang, Yangyi Chen, Lifan Yuan, Yizhe Zhang, Yunzhu Li, Hao Peng, and Heng Ji. Executable code actions elicit better llm agents. *arXiv preprint arXiv:2402.01030*, 2024b.

Jialong Wu, Shaofeng Yin, Ningya Feng, Xu He, Dong Li, Jianye Hao, and Mingsheng Long. ivideogpt: Interactive videogpts are scalable world models. In *NeurIPS*, 2024.

Jiannan Xiang, Guangyi Liu, Yi Gu, Qiyue Gao, Yuting Ning, Yuheng Zha, Zeyu Feng, Tianhua Tao, Shibo Hao, Yemin Shi, et al. Pandora: Towards general world model with natural language actions and video states. *arXiv preprint arXiv:2406.09455*, 2024.

Zeqi Xiao, Yushi Lan, Yifan Zhou, Wenqi Ouyang, Shuai Yang, Yanhong Zeng, and Xingang Pan. Worldmem: Long-term consistent world simulation with memory. In *NeurIPS*, 2026.

Sherry Yang, Yilun Du, Kamyar Ghasemipour, Jonathan Tompson, Leslie Kaelbling, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. *arXiv preprint arXiv:2310.06114*, 2023.

Yue Yang, Fan-Yun Sun, Luca Weihs, Eli VanderBilt, Alvaro Herrasti, Winson Han, Jiajun Wu, Nick Haber, Ranjay Krishna, Lingjie Liu, et al. Holodeck: Language guided generation of 3d embodied ai environments. In *CVPR*, 2024.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

Jiwen Yu, Jianhong Bai, Yiran Qin, Quande Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Xihui Liu. Context as memory: Scene-consistent interactive long video generation with memory retrieval. In *SIGGRAPH Asia*, 2025.

Ailing Zhang, Lina Lei, Dehong Kong, Zhixin Wang, Jiaqi Xu, Fenglong Song, Chun-Le Guo, Chang Liu, Fan Li, and Jie Chen. Ui2v-bench: An understanding-based image-to-video generation benchmark. *arXiv preprint arXiv:2509.24427*, 2025.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *ICCV*, 2023.

A Appendix

In this appendix, we provide additional details of API World Bench. We first describe the use of AI systems in our work, including their roles as components of the proposed framework and auxiliary tools for manuscript preparation. We then provide additional details of benchmark design, including capability coverage, task examples, and evaluation protocols. Finally, we discuss the limitations of our current framework and potential directions for future work.

A.1 AI Use Statement

The use of AI systems in this work serves two distinct purposes. First, several AI models are integrated as essential components of our proposed framework. Specifically, we employ GPT-6 OpenAI (2026) as the coding agent to construct and operate executable virtual worlds through virtual engine APIs. We also use Qwen3-VL-32B-Instruct Bai et al. (2025) to generate textual prompts for semantic-to-real video translation, and initialize our visual renderer from the pretrained MiniMax-H3 MiniMax (2026) model. These AI models constitute the core components of API World Model rather than auxiliary tools. Second, we use AI-based tools to assist with language polishing and improve the readability of the manuscript. All technical ideas, methodology design, experimental settings, analysis, and conclusions are developed, verified, and finalized by the authors.

Table 2: Comparison of benchmarks. Spatial: spatial relations and geometric constraints; Time: specified event times, durations, and delays; Phys. Fact.: physical factuality; Phys. Quant.: responses to specified physical quantities; Causal: consequences of interventions; Counterf.: evolution under modified conditions or rules; Plan.: goal-directed planning under constraints. A checkmark indicates dedicated tasks or metrics for a dimension; a cross indicates no explicit evaluation.

Benchmark	Spatial	Time	Phys. Fact.	Phys. Quant.	Causal	Counterf.	Plan.
TiViBench (Chen et al., 2025)	✓	×	×	×	×	×	✓
WorldModelBench (Li et al., 2026a)	×	×	✓	×	×	×	×
Physics-IQ (Motamed et al., 2026)	×	×	✓	×	✓	×	×
What-If World (Cai et al., 2026)	✓	×	✓	×	✓	✓	×
UI2V-Bench (Zhang et al., 2025)	✓	×	✓	×	✓	×	×
TC-Bench (Feng et al., 2024)	✓	×	×	×	×	×	×
VBench++ (Huang et al., 2025)	✓	×	×	×	×	×	×
PhyWorldBench (Gu et al., 2026)	×	×	✓	×	×	✓	×
Morpheus (Tragoudaras et al., 2025)	×	×	✓	×	×	×	×
WorldScore (Duan et al., 2025)	✓	×	×	×	×	×	×
MechVerse (Jain et al., 2026)	✓	×	✓	×	✓	×	×
API World Bench (Ours)	✓	✓	✓	✓	✓	✓	✓

A.2 API World Bench Details

A.2.1 Task Examples

API World Bench is designed to evaluate whether a world model can correctly execute and evolve a world according to instructions, rather than merely generate visually plausible videos. Each task starts from an initial executable world and evaluates whether the generated rollout satisfies the specified state transitions, interactions, and physical constraints.

The benchmark covers seven dimensions of world behavior:

Spatial Understanding. These tasks evaluate whether the model can correctly maintain spatial relationships between entities. For example, the instruction may require an object to move to a specified location, maintain a relative distance to another object, or satisfy a geometric constraint.

Temporal Control. These tasks test whether the model follows temporal requirements, including event ordering, delays, and duration constraints. For example, an object may be required to start moving after a certain delay or remain stationary for a specified period before an interaction.

Physical Factuality. These tasks evaluate whether the generated world follows physical rules. Examples include gravity-driven motion, collision responses, object support relationships, and parameter-dependent dynamics.

Physical Quantities Control. These tasks further examine whether the model responds consistently to changes in physical quantities. For example, modifying the mass or applied force of an object should result in corresponding changes in its motion.

Causal Reasoning. These tasks evaluate whether the model captures causal consequences of interventions. Instead of only predicting observed trajectories, the model should generate the correct outcome after an explicit modification to the world state.

Counterfactual Prediction. These tasks evaluate whether the model can simulate alternative outcomes under changed conditions. The evaluation compares the generated evolution with the expected consequence under the modified world configuration.

Planning. These tasks measure whether the model can complete multi-step objectives requiring sequential interactions. The generated rollout must satisfy both intermediate requirements and the final goal state.

Table 2 compares our benchmark with existing benchmarks for evaluating world modeling and video generation. We categorize the evaluation dimensions into seven aspects: spatial reasoning, temporal reasoning, physical factuality, physical quantities, causal reasoning, counterfactual reasoning, and planning. While existing benchmarks typically focus on individual aspects of these capabilities, our benchmark provides a more comprehensive evaluation that jointly considers physical dynamics, causal interactions, counterfactual evolution, and goal-directed planning. In particular, it includes dedicated tasks for evaluating physical quantities and counterfactual scenarios, which are less explicitly covered by existing benchmarks. Particularly, none of existing benchmarks explicitly evaluates a model's ability to control and respond to specified physical quantities. Our benchmark introduces dedicated tasks for physical quantity control, requiring models to generate and predict outcomes under explicitly specified values of physical parameters, such as velocity, acceleration, force, or duration. This evaluation provides a direct test of whether a model can translate quantitative physical specifications into corresponding world dynamics, going beyond qualitative judgments of physical plausibility. This broader coverage enables a more systematic assessment of whether video models can not only generate visually plausible content, but also reason about and predict the evolution of a dynamic world. We hope this benchmark can encourage future research toward more precise and controllable world modeling.

A.2.2 Evaluation Protocol

For each benchmark episode, the evaluator receives the initial world condition, the instruction, and the generated video rollout. The evaluation combines semantic verification and quantitative measurements.

Semantic Verification. We first evaluate whether the required events and interactions occur in the generated video. A vision-language evaluator analyzes the rollout and determines whether the specified objects, actions, and state changes are correctly executed. This evaluation focuses on high-level task completion, such as whether an object is moved, whether an interaction occurs, and whether the final state matches the instruction.

Quantitative Measurement. For tasks with measurable physical or spatial constraints, we additionally compute quantitative metrics from the generated videos. Object trajectories, positions, and temporal events are extracted using visual perception models and compared with the expected constraints defined by the executable world.

For each task, the final score combines semantic correctness and quantitative consistency. A generated video receives a high score only when it both satisfies the semantic requirement and preserves the underlying spatial, temporal, and physical constraints. The category score is computed by averaging all tasks within the corresponding category, and the overall benchmark score is obtained by averaging across all categories.

A.3 Limitations and Future Work

Although API World Model enables executable and controllable world evolution, several limitations remain. First, the current framework relies on coding agents and virtual engines, whose generation capability and simulation fidelity may limit the complexity of supported worlds. Improving agent-based world construction and scaling to more diverse environments are important future directions. Second, the visual renderer still inherits limitations from current video generation models, and maintaining consistent visual quality during longer and more complex interactions remains an open challenge.