VIDEOCOS: When Do Executable World States Help Video Reasoning and Generation?

Shoubin Yu* Daeun Lee* Zun Wang Han Lin Yue Zhang Mohit Bansal

UNC Chapel Hill

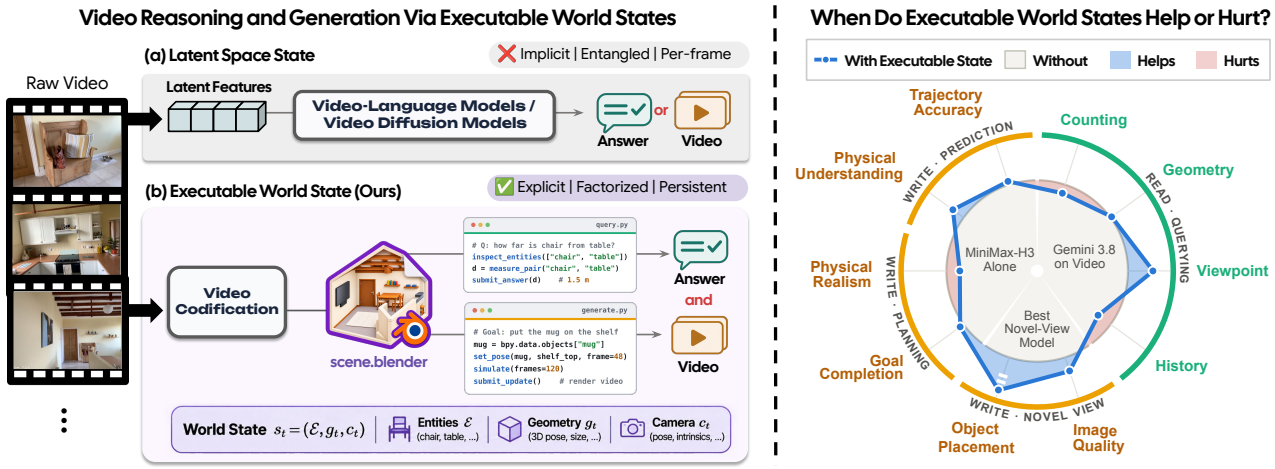

Figure 1. **Executable world state for unified video reasoning and generation. Left:** Unlike VLMs and video diffusion models with implicit, entangled latents, VIDEOCOS codifies a video once into an executable program (`scene.blender`)—an explicit, factorized, and persistent state that a reasoning agent queries (`query.py`) to answer questions or edits and simulates (`generate.py`) to render new videos. **Right:** Performance across **Read** and **Write** axes relative to base models without executable state (gray). Blue shows gains (e.g., viewpoint, object placement, goal completion, physical understanding). Pink shows bottlenecks (history, physical realism).

Abstract

Video reasoning and generation often depend on the underlying world state, e.g. scene entities, their spatial layouts, and object and camera motion. We investigate whether and when representing this state as an executable program *improves video reasoning, generation, and embodied planning. We introduce a unified read–write formulation that casts reasoning as state queries and generation as state edits, alongside* VIDEO REASONING AND GENERATION VIA CODE STATES *(*VIDEOCOS*), a decoupled agentic framework operating via Blender code. A perception-grounded codification agent first reconstructs a reusable scene program from raw video. A separate reasoning agent then works from this program alone, isolating the effect of the state, and reads it to answer questions or writes to it via edits, simulation, and rendering. Across seven read and write task families , three findings emerge. **(1)** Executable states help when a task reduces to explicit geometric operations on 3D structure, such as imagined-viewpoint reasoning, novel-view rendering, and rigid object placement. **(2)** They fall short when answers combine several noisy state estimates (e.g., object order) or depend on physical realism, such as hand interaction, fluids, and photorealism. **(3)** They can outperform raw video despite state errors up to 25% in distance or 90° in direction, but incorrect states can hurt more than missing ones, particularly for viewpoint and order. We hope our study motivates* hybrid world models *that pair executable code for persistent structure and planning with neural models for photorealistic and non-rigid synthesis.*

*Equal contribution.

1. Introduction

Understanding and generating videos requires more than what any single frame shows. Frames capture observa-

tions, whereas the **world state** describes what persists behind them, namely which entities exist, where they are, and how they and the camera move over time. Consider the indoor scene in Fig. 2, counting the cups across a walkthrough requires maintaining persistent object identities across occlusions (T1), and rendering the room from a new angle requires updating the camera trajectory while keeping the scene layout fixed (T5). We call such tasks *state-centric*. As shown in Fig. 1, current video models keep this state implicit: video-language models (VLMs) predict spatial counts and directions directly from visual features [24, 53], while video diffusion models (VidDMs) entangle object identity, geometry, and viewpoint inside high-dimensional latents [3, 20, 36, 44, 48, 49]. Lacking explicit persistence and factorized representation, these models struggle to understand and stay consistent across large viewpoint shifts, cannot execute exact geometric transformations, and frequently corrupt untargeted objects or backgrounds when editing or inspecting a single scene factor.

A growing line of work makes this state explicit as **executable code**, using coding agents to reconstruct a video into a Blender program [15, 42, 43]. Built once from video, such a program provides a persistent, factorized, and simulatable world state that can be queried to answer questions or edited and re-rendered to generate new videos. However, two gaps remain: (1) existing methods rely solely on a coding agent's visual perception without geometric or tracking priors, yielding imprecise, interpenetrating scenes that break downstream tasks; and (2) prior work evaluates how visually similar renders are to the input video, leaving open whether a state reconstructed from raw video can reliably support downstream reasoning and generation. We therefore ask *whether and when an executable world state helps video reasoning and generation, and where its limitations lie*.

In this paper, we answer this question through a unified read–write formulation, a diagnostic task suite, and a decoupled agentic framework. As illustrated in Fig. 1 (bottom-left), we first formalize the world state as decoupled components (entities, 3D geometry, and camera trajectory) instantiated as an *executable scene program*, and unify video reasoning as **state reads** (querying the state without changing it) and video generation as **state writes** (editing or simulating targeted components while keeping the rest fixed). Next, we organize existing benchmarks into seven task families probing distinct state components (Fig. 2): four read tasks (object counting, metric geometry, viewpoint, and appearance order) and three write tasks (novel-view synthesis, goal planning, and future prediction). Because benchmarks lack ground-truth scene programs, every state must be inferred directly from raw video, which allows us to test whether noisy reconstructions still benefit downstream tasks.

Inferring and evaluating such states, however, exposes two limitations of existing coding/tool agents: (1) they lack

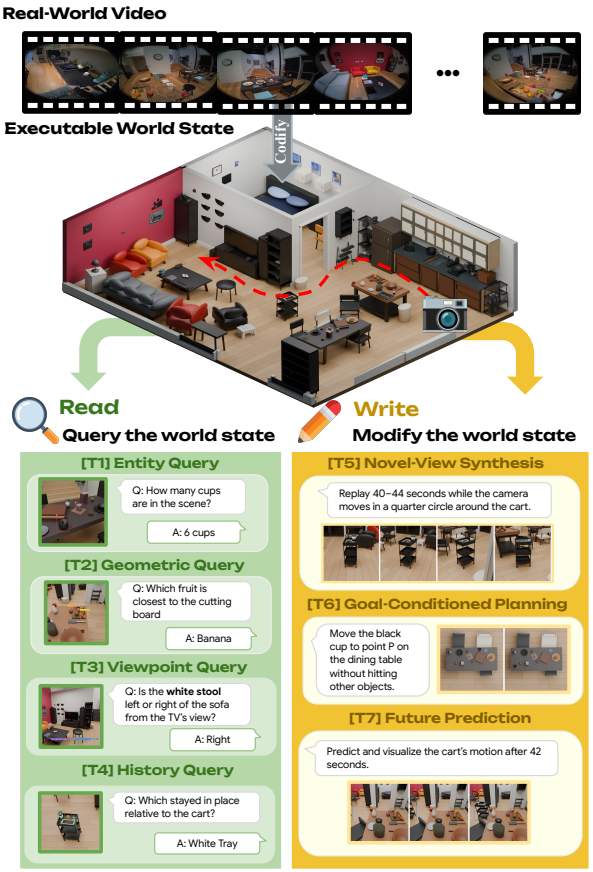

Figure 2. **Overview of our seven state-centric video tasks.** Read tasks (T1–T4) query entity identities, geometry, observer-relative coordinates, and visibility history from the scene state. Write tasks (T5–T7) modify targeted state components under specific drivers and render the resulting video.

geometric priors, yielding imprecise scenes, and (2) they rebuild the state or tool chain for every query, so nothing is reused. To overcome both, we propose VIDEO REASONING AND GENERATION VIA CODE STATES (VIDEOCOS), which grounds reconstruction in perception priors and decouples scene construction from downstream execution via a reusable Blender program (Fig. 1). First, we extract diverse perception cues (such as depth, camera poses, and object tracks) with off-the-shelf perception models [5, 8, 30, 39], and then a **codification agent** iteratively writes and refines a Blender program using this evidence. This produces a persistent, executable world representation that is shared and reused across downstream queries and edits. Second, a **reasoning agent** loads this saved program and operates solely on this representation without accessing raw frames or renders, which amortizes one reconstruction across tasks while isolating performance to the state itself.

As shown in the right of Fig. 1, our experiments across all seven task families yield three findings: **(1) Executable**

states help with explicit geometric operations. Putting objects and cameras in one 3D space boosts imagined-viewpoint reasoning by up to 21.9 points, keeps novel views consistent across 128° camera rotations where video diffusion models break down, and gives direct control over rigid object placement and motion. **(2) They fall short on compositional queries and physical realism.** *Appearance order* (which object appeared first) degrades as errors compound in estimated camera paths and object locations, and rendered videos lack physical realism for human hands, non-rigid phenomena like fluids, and photorealistic appearance. **(3) They tolerate state errors, depending on the task.** Code input beats video until distances are off by over 25% or directions by over 90°, but one miscount or swapped pair puts it behind. A wrong state can also hurt more than a missing one, particularly for viewpoint and order. Together, these findings motivate **hybrid world models** that use executable code for persistent 3D structure and planning, and neural models for photorealistic and non-rigid synthesis.

- A **unified** formulation of state-centric video reasoning and generation as programmatic state reads and writes across seven public task families.
- **VIDEOCOS**, a decoupled agentic framework that codifies video into a simulatable Blender scene and executes downstream reads and writes without accessing raw frames.
- A **diagnostic study** showing where executable world states help and where they fall short, providing empirical design principles for hybrid code–neural world models.

2. Related Work

2.1. Implicit World State in Video Models

Video-language models have improved at spatial and temporal reasoning through reinforcement learning and explicit perception mechanisms [9, 12, 24, 28, 33, 50, 52, 59, 60], with benchmarks such as VSI-Bench [2, 53] testing whether they recover counts, distances, and directions from room videos. Diffusion models are increasingly used to reason by generating frames [3, 19, 20, 29, 44, 48, 51, 57, 58], an ability that recent work analyzes and improves through diffusion dynamics, reward modeling, and inference-time search [31, 36, 49]. Both families hold the scene implicitly in latents, whereas we write the world state out as code and use it for both question answering and video generation.

2.2. Programs and Tools for Visual Reasoning

Visual programming methods generate code that decomposes a query into calls to vision models and tools [18, 21, 41], and have been extended to temporal grounding and reasoning over long videos [14, 25]. Recent agents target 3D spatial reasoning with tools that reconstruct the scene and compute over it [11, 32], and SpatialClaw [13] further keeps a persistent Python environment across reasoning steps. In these methods the program describes how to answer one question, whereas in VIDEOCOS it describes the scene itself.

2.3. Scenes as Code

Code also serves as an intermediate representation for generation, where programs give controllable plans for image synthesis [26, 54] and executable 3D scenes act as a proxy for video generation [6, 10, 27, 47], and for reconstruction, where inverse-graphics methods recover editable programs for articulated objects [46, 62] and interactive environments for reasoning [4, 22, 55]. Closest to our codification agent, BVB [43], Video2World [42], 3DCodeVerse [15], and 4DCodeBench [40] evaluate agents that reconstruct dynamic videos as executable graphics programs. These works measure how faithfully code reproduces a scene, while we ask what the codified scene is good for, namely which downstream tasks it serves better than raw video and how its errors carry into reasoning and generation.

3. VIDEOCOS: Executable World States for Video Reasoning and Generation

To examine when and how an explicit world state benefits video reasoning and generation, we first formalize the state behind a video as persistent attributes like entities, 3D geometry, and camera motion, instantiated as an *executable* scene program (Sec. 3.1)[1] We then cast video reasoning and generation as programmatic *read* and *write* operations over this state across seven task families (Sec. 3.2). Finally, we describe how VIDEOCOS constructs this state from raw video and operates on it (Sec. 3.3).

3.1. Executable World State Representation

Every frame I_t of a video is a 2D observation rendered from an underlying scene state at a specific viewpoint (e.g., the room walkthrough in Fig. 2 top). We decompose the world state at frame t into three components, $s_t = (\mathcal{E},\ g_t,\ c_t)$, which together suffice to re-render the video via $V = \mathcal{R}(s_{1:T})$:

- **Entities ($\mathcal{E}$):** The discrete set of objects and structural elements (e.g., sofas, tables, cups, walls), each with a persistent identity and time-invariant attributes (semantic category, 3D mesh, surface appearance, and basic rigid-body properties). $\mathcal{E}$ is therefore not indexed by t.
- **Geometry (g_t):** The 3D position, orientation, and metric scale of every entity at t in a shared world coordinate system, with $g_{1:T}$ capturing motion and articulation (e.g., the moving cup or cart in Fig. 2 bottom-right).
- **Camera (c_t):** The camera pose and intrinsics at t, with $c_{1:T}$ recording its trajectory (red path in Fig. 2 top).

[1] Our state covers mainly rigid objects and scene structure, and less emphasis on humans/non-rigid, as they are hard to recover as code simulation.

Table 1. **Seven state-centric tasks.** The *State Operation* lists the components of $s_t = (\mathcal{E}, g_t, c_t)$ that each task queries or modifies.

	Task Family	State Operation	Task Details	Data Source	n
Read	Entity Querying (**EQ**)	$\mathcal{E}$	count distinct objects across frames	VSI-Bench-Debiased [2, 53]	251
	Geometric Querying (**GQ**)	g_t	object/room size and absolute/relative distance		1,297
	Viewpoint Querying (**VQ**)	g_t in a query-defined reference frame	relative direction and route planning		496
	History Querying (**HQ**)	$\mathcal{E}, g_{1:T}, c_{1:T}$	order in which objects first become visible		318
Write	Novel-View Synthesis (**NVS**)	$c_{1:T} \to c'_{1:T}$	observed video → novel-view video	Kubric-4D [17, 45]	300
	Goal-Conditioned Planning (**GCP**)	$g_1 \to g'_{1:T'}$	observed frame + goal → video	Ego2Act [23]	110
	Future Prediction (**FP**)	$g_{1:T} \to g'_{T+1:T'}$	observed video → future video	Physics-IQ [35]	66

Rather than a static annotation, this state is **executable**. It is realized as a Python program P (`scene.py`) whose execution instantiates a live Blender scene (`scene.blender` in Figs. 1 and 3) at any frame t, where g_t and c_t jointly determine object visibility and occlusion during rendering. This representation provides three properties that the latent states of VLMs and video diffusion models do not expose:

1. **Persistence:** Entities keep their identities and coordinates across occlusions and viewpoint changes (e.g., all six cups in Fig. 2), so every query sees one consistent scene.
2. **Factorization:** Identity ($\mathcal{E}$), layout and motion (g_t), and viewpoint (c_t) are decoupled, so one component can be edited, such as orbiting the camera or moving a cup, while the rest stay fixed.
3. **Executability:** As runnable code (`query.py` and `generate.py` in Fig. 1 left), the state supports deterministic geometric queries, rigid-body simulation, and counterfactual rendering via $\mathcal{R}(\cdot)$.

Since the true state is unobserved, we infer P from the video V (Sec. 3.3), and executing it yields the estimated state.

3.2. Reading and Writing World States

To locate where an executable state helps and where it falls short, we organize existing benchmarks into seven task families, each probing specific components of $s_{1:T}$ through either a *read* or a *write* (Fig. 2 bottom and Tab. 1).

State Reading (Video Reasoning). A read operation $\mathcal{O}_{\text{read}}(s_{1:T}, q) \to y$ runs a programmatic query q over the state and returns an answer y (Fig. 2, purple). **Entity Querying (EQ)** counts distinct instances in $\mathcal{E}$, such as the six cups seen across a room walkthrough. **Geometric Querying (GQ)** measures metric sizes and distances from g_t. **Viewpoint Querying (VQ)** re-expresses g_t in an observer-relative frame to infer directions and routes, such as whether the white stool is left or right of the sofa *from the TV's view*. **History Querying (HQ)** combines $\mathcal{E}$, $g_{1:T}$, and $c_{1:T}$ to recover the order in which objects first become visible.

State Writing (Video Generation). A write operation $\mathcal{O}_{\text{write}}(s_{1:T}, u) \to s'_{1:T'}$ edits or extrapolates the targeted components given conditioning u, holds the others invariant, and renders $V' = \mathcal{R}(s'_{1:T'})$ (Fig. 2, tan). **Novel-View Synthesis (NVS)** overwrites only the camera trajectory ($c_{1:T} \to c'_{1:T}$), as in orbiting around the cart. **Goal-Conditioned Planning (GCP)** synthesizes a collision-free, goal-reaching object trajectory from the first frame ($g_1 \to g'_{1:T'}$) under a fixed camera, as in moving the black cup to a marked target point. **Future Prediction (FP)** extrapolates the observed motion via simulation ($g_{1:T} \to g'_{T+1:T'}$).

3.3. Constructing and Operating on Executable World States

Existing benchmarks provide no ground-truth scene programs, so studying reads and writes first requires reconstructing P from raw video. Existing code-based reconstruction [43] is often geometrically inaccurate, which makes it hard to tell whether a downstream failure comes from the representation or from a poor reconstruction. Our VIDEOCOS therefore uses two decoupled agents (Fig. 3). A **codification agent** builds P once per video with perception evidence, and a **reasoning agent** then performs all reads and writes on P.

Codifying Video into an Executable State. Prior work [43] relies on the coding agent alone to infer 3D structure from visual appearance, which frequently leads to inaccurate room dimensions, object scales, positions, and geometries (see Fig. 8). To ground the reconstruction in metric measurements, we first run off-the-shelf perception models [5, 8, 30] to extract read-only reference evidence (top-left of Fig. 3), including camera poses, depth maps, a canonical 3D world frame, 2D segmentation masks, and 3D meshes. Conditioned on the raw video and this evidence (top-right of Fig. 3), the agent writes an initial program and refines it inside a live Blender environment with a *build–check–fix* loop:

1. **Build & Render:** The agent executes P and renders frames corresponding to the input video.
2. **Check:** Diagnostics and visual comparisons against the real frames, flag invalid syntax, inter-object collisions, implausible trajectory, and missing or floating entities.
3. **Fix:** The agent edits P to resolve the flagged errors and repeats the cycle until all checks pass or the time and iteration budget is exhausted.

Passing these checks does not guarantee an accurate reconstruction, so we evaluate state accuracy separately in Fig. 7.

Operating on the Executable State. Once codification finishes, P is loaded into a separate, persistent Blender process

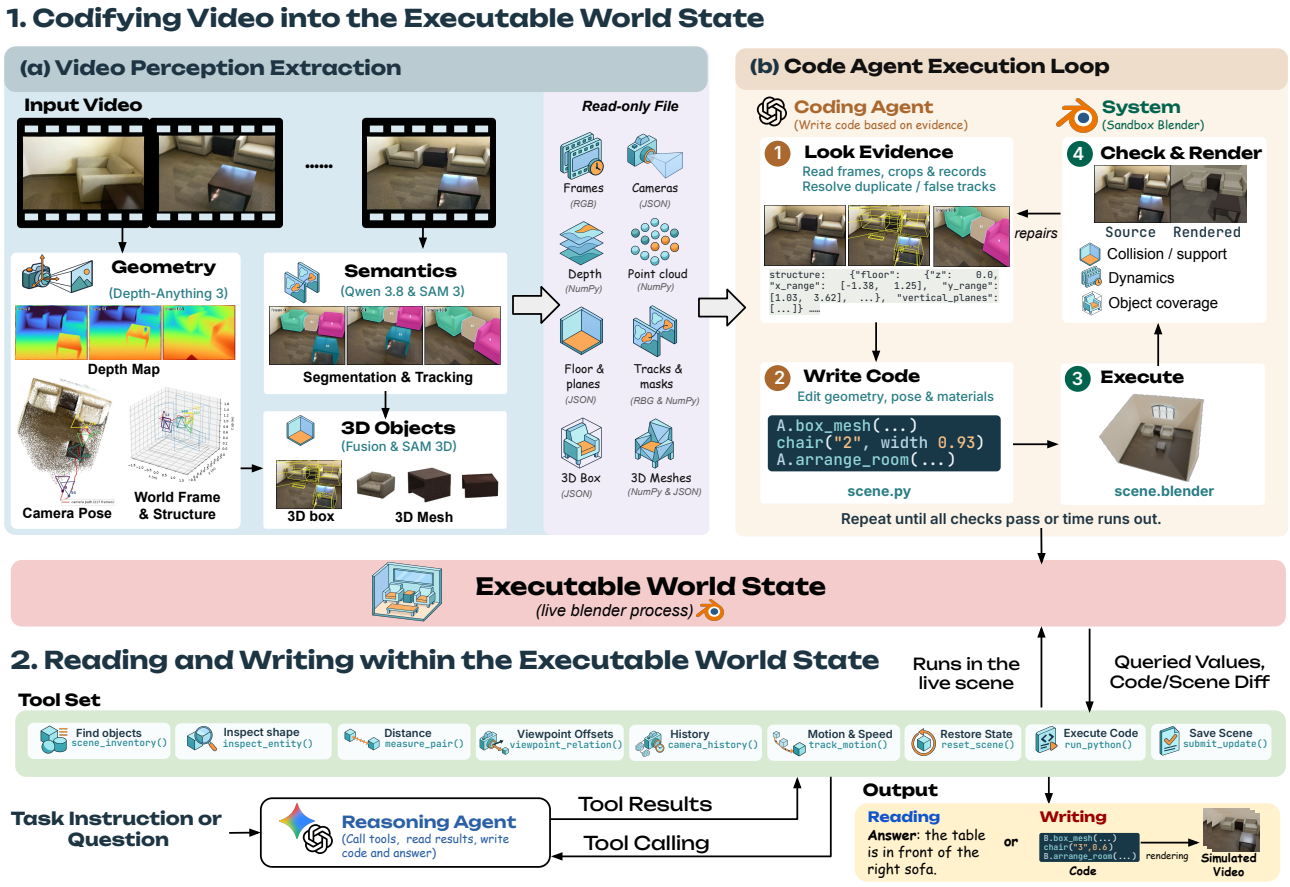

Figure 3. **Overview of VIDEOCOS.** The codification agent turns perception evidence into an executable scene program, which gives the estimated state $\hat{s}_t = (\hat{\mathcal{E}},\ \hat{g}_t,\ \hat{c}_t)$ (red in the middle), through a loop of building, checking, and rendering. The reasoning agent then reads this scene to answer questions, or edits and simulates it to generate videos. It only receives text feedback from tools and never sees the frames.

(middle of Fig. 3). Given a task instruction, the reasoning agent iteratively invokes programmatic tools and updates its plan based on their text outputs. It never accesses the raw video, so all downstream actions depend on $\hat{s}_{1:T}$.

For *reads*, the agent calls read-only inspection tools and composes their outputs into a text answer:

- `scene_inventory` and `inspect_entities` return object identities and 3D bounding geometry;
- `measure_pair` returns metric distances between specified object pairs;
- `viewpoint_relation` projects coordinates into an observer-centric frame defined by a reference object;
- `camera_history` logs the camera trajectory alongside the frame at which each object first becomes visible.

For *writes*, the agent edits the scene by executing Python scripts via `run_python`, supported by helper functions for poses, keyframes, articulated joints, cameras, and predefined physical rules. Each execution returns a text diff summarizing the modified entities and their updated trajectories. To enforce factorization, an edit is accepted only if it changes the components its task targets and keeps other parts unchanged. For NVS, the camera must match the target poses $\hat{c}'_{1:T}$ with all entities and geometry unchanged. For GCP and FP, the camera and the observed geometry ($\hat{g}_1$ or $\hat{g}_{1:T}$) must remain fixed. This check enforces *invariance* rather than *task success*. It prevents unintended edits to preserved components but does not verify that the goal is achieved. The accepted state is then rendered into the output video V'. More details are in Appendix Sec. A.

4. Experiments

After the setup (Sec. 4.1), we ask where executable world states help (Sec. 4.2), where they fall short (Sec. 4.3), and how a state should be built and used (Sec. 4.4). More details and results are in the Appendix Secs. B to D.

4.1. Setup

Benchmarks and Evaluation. We evaluate the seven task families of Tab. 1. Read tasks use VSI-Bench-Debiased [2, 53] with its official scoring, regrouping its eight question

Table 2. **State querying results on VSI-Bench-Debiased.** Results are grouped into 4 tasks. Green/red indicate improvements/declines relative to the video baseline. Best in **bold**, second-best underlined. Full table in Appendix Sec. C.1.

Reasoner	State	Entity	Geom.	View.	Hist.	Avg.
Qwen3.8-27B	Video	51.7	**56.1**	59.0	**74.5**	**58.6**
+ VIDEOCOS	Code (BVB)	**58.1**	52.1	**64.8**	64.4	54.1
+ VIDEOCOS	Code (Ours)	54.1	**56.1**	60.2	62.0	57.6
GPT-5.6 Luna	Video	49.4	50.3	40.6	**68.6**	50.0
+ VIDEOCOS	Code (BVB)	**59.0**	53.9	**68.9**	63.8	59.5
+ VIDEOCOS	Code (Ours)	54.0	**58.1**	62.5	64.5	**59.6**
Gemini-3.8 Flash	Video	**59.6**	60.2	63.2	**77.7**	63.1
+ VIDEOCOS	Code (BVB)	58.5	56.0	79.3	69.2	63.8
+ VIDEOCOS	Code (Ours)	53.7	**61.1**	**80.7**	64.5	**65.5**

types into our four read tasks. Each write task has a video-level and a state-level metric. NVS uses Kubric-4D [17, 45] with PSNR and the mask IoU (mIoU) of SAM3 object tracks against the ground truth. GCP uses Ego2Act [23], whose MLLM judge gives a physical score (P) and a task score (T). FP uses Physics-IQ [35] with its official score and the average displacement error (ADE) of SAM3 object tracks. To analyze codified scenes, we measure *visual similarity* with BVB's [43] Dual VQA (DV) and Latent Similarity (LS), and *state accuracy* against benchmark annotations of object counts, sizes, distances, and room area.

Baselines and Implementation Details. Codification uses Codex CLI with GPT-6.1 Terra [38]. The reasoning agents are Gemini 3.8 Flash [16], GPT-5.6 Luna [37], and Qwen3.8-27B [39], and they never see the raw video. Write tasks always use Gemini 3.8 Flash and are compared with diffusion-based video generators. For GCP and FP, a hybrid variant re-renders the Blender video with MiniMax-H3 [34] (Ref2VA with Turbo LoRA) via a depth ControlNet [61].

4.2. Where Do Executable World States Help?

> **Finding 1:** Executable states help when a task is a direct geometric operation on 3D structure, such as changing the viewpoint or moving rigid objects.

They help under viewpoint changes, in both reading and writing. On reads (Tab. 2), viewpoint accuracy (View.) improves for all three reasoners, by +21.9 and +17.5 points for GPT and Gemini. A video reasoner must infer the imagined viewpoint or route implicitly, whereas the executable scene keeps objects and observers in one 3D coordinate system, where a viewpoint change is a simple transform. Writes show the same benefit. In NVS (Tab. 3), a camera write changes only $\hat{c}$ and re-renders the same scene. VIDEOCOS leads on both metrics at all camera shifts, and its object mIoU stays at 0.224–0.254 from 42° to 128° while diffusion baselines struggle for distorted generation (Fig. 4).

Table 3. **Novel-view Synthetics results on Kubric-4D.** The full table is included in the Appendix Sec. D.1.

Method	Low (42°)		Medium (97°)		High (128°)	
	PSNR↑	mIoU↑	PSNR↑	mIoU↑	PSNR↑	mIoU↑
Ground truth	–	0.604	–	0.619	–	0.618
Diffusion-based novel-view synthesis						
TrajectoryCrafter [56]	10.88	0.020	10.45	0.004	10.19	0.004
ReCamMaster [1]	11.68	0.016	10.47	0.003	9.34	0.001
CogNVS [7]	11.49	0.008	11.04	0.001	10.69	0.001
VIDEOCOS *camera edit in executable state, rendered by*						
Blender	**12.87**	**0.254**	**12.98**	**0.230**	**12.67**	**0.224**



Figure 4. **Qualitative comparison on NVS.** VIDEOCOS renders more state-consistent novel views and matches the GT view better than the strong diffusion baseline. More in Appendix Sec. D.1.

Table 4. **Goal-conditioned Planning results on Ego2Act.** Tasks are split by the median time humans take to complete them into Short ($\leq$ 15 s) and Long ($>$ 15 s). T and P are task completion and physical plausibility, and Final averages the per-video $\sqrt{TP}$.

Method	Short		Long		Total		
	T	P	T	P	T	P	Final
Video generators (plan and render jointly)							
Seedance-2.0	**87.8**	**92.2**	**74.0**	**87.8**	**77.3**	**88.9**	**81.5**
Kling-v3-Pro	62.4	88.6	57.6	87.7	58.8	87.9	69.5
Grok-1.5	71.0	81.2	61.1	87.5	63.4	86.0	71.5
Wan-2.7	72.3	85.0	61.6	83.3	64.1	83.7	70.6
Cosmos-3	19.6	52.0	13.1	63.1	14.6	60.2	19.2
MiniMax-H3	67.2	88.7	52.0	76.4	55.6	79.0	62.3
VIDEOCOS *plan in executable state, rendered by*							
Blender	81.3	53.5	64.2	48.9	68.2	50.0	52.5
Blender+MiniMax-H3	67.9	78.8	55.3	64.2	58.3	67.3	56.5

They help control and plan rigid object motion. On GCP (Tab. 4), VIDEOCOS rendered in Blender has the second-highest task score after SoTA Seedance-2.0 on both short and long tasks (81.3 and 64.2, Fig. 5). On FP (Tab. 5), its rigid

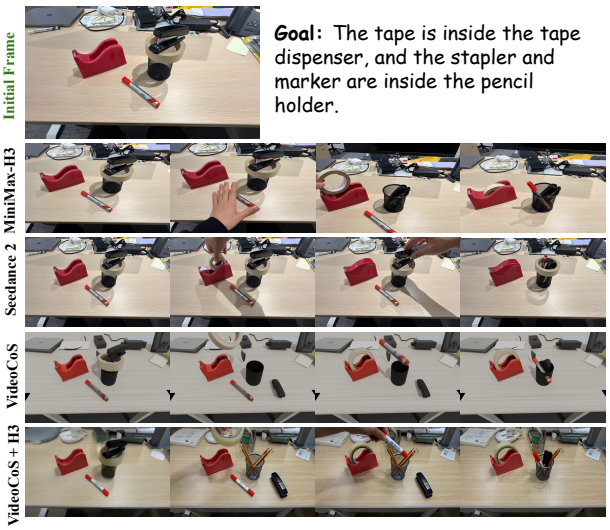

Figure 5. **Qualitative comparison on GCP.** Video diffusion models often handle the objects in the wrong order or hallucinate new ones. VIDEOCOS plans the correct object movements, and rendering its plan with H3 makes the video more realistic and closer in appearance to the initial frame. More in Appendix Sec. D.2.

Table 5. **Future Prediction results on Physics-IQ.** *Rigid* covers rolling, falling, colliding, and magnet-driven motion. *Non-rigid* covers deformation, fracture, and liquids. Others (lighting and mirror effects, and combustion) count only toward *All*.

Method	Physics-IQ ↑			ADE ↓	
	All	Rigid	Non-rigid	Rigid	Non-rigid
Video generators (plan and render jointly)					
Cosmos3-Super	**59.3**	**56.8**	**64.2**	**10.3**	5.4
Cosmos3-Nano	56.3	53.5	61.1	10.5	5.5
Physical Registry	48.8	46.4	49.8	11.9	**5.3**
MiniMax-H3	34.5	37.5	31.1	16.1	7.9
VIDEOCOS *plan in executable state, rendered by*					
Blender	37.5	45.1	31.1	11.1	9.9
Blender+MiniMax-H3	39.6	42.6	36.4	15.5	7.8

trajectory error is 11.1, close to Cosmos3 (10.3 and 10.5) and well below MiniMax-H3 (16.1, Fig. 6). These plans also help a diffusion model. Conditioning MiniMax-H3 on depth maps from the edited scene raises its GCP task score from 55.6 to 58.3 and its Physics-IQ score from 34.5 to 39.6.

4.3. Where Do They Fall Short?

> **Finding 2:** Executable states fall short when an answer combines several noisy state estimates or requires physical realism (hands, non-rigid dynamics, photorealism).

They fall short when answers combine several state estimates. History accuracy (Hist. in Tab. 2) drops for every

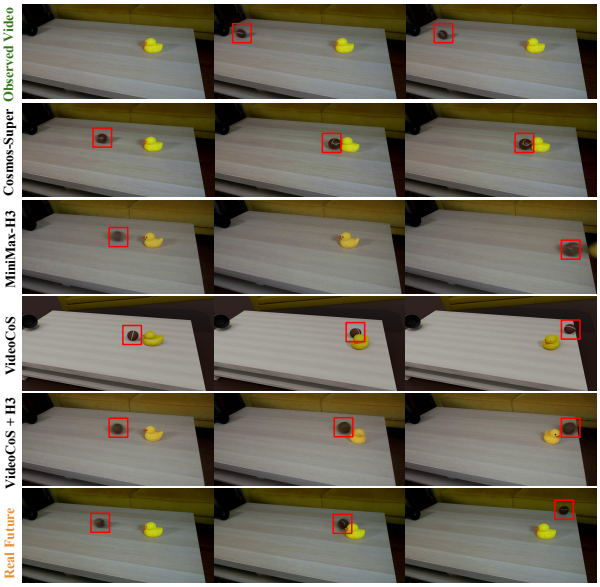

Figure 6. **Qualitative comparison on FP.** VIDEOCOS predicts object motions closer to the GT future video, while diffusion baselines can generate implausible motions and collisions. Additional examples are provided in the Appendix Sec. D.3.

reasoner. An appearance-order answer combines four first-appearance estimates, one per object, each computed from the estimated objects and camera path. We score the sub-orders of k objects inside each Gemini answer (Fig. 7, top left). As k grows, code accuracy falls more steeply than video accuracy (20.1 vs. 13.5 points from k=2 to k=4). The gap to video input therefore widens from -6.6 points at k=2 to -13.2 at k=4, while the code error stays 1.6–1.75× the video error. The more objects an answer must order, the further code input falls behind video input.

They fall short on physical and visual realism. On GCP (Tab. 4), VIDEOCOS has the lowest physical score (50.0), since hands are not captured in the state and objects move on their own (Fig. 5). On FP (Tab. 5), its Physics-IQ score (37.5) trails Cosmos3-Super (59.3) and its non-rigid trajectory error is the highest (9.9 vs. 5.3–7.9), since rigid-body simulation cannot capture fluid or electromagnetism. Diffusion does not close this gap, as conditioning H3 on the Blender plan lowers its GCP physical score from 79.0 to 67.3. This trade-off between control and realism motivates hybrid world models with targeted neural conditioning.

4.4. How Should a State Be Built and Used?

> **Finding 3:** Executable states can beat video despite imperfect reconstruction, but error tolerance varies by task, and a wrong state hurts more than a missing one.

Build for state accuracy, not visual similarity. Across the three codified scenes of each room (Fig. 8), QA accuracy

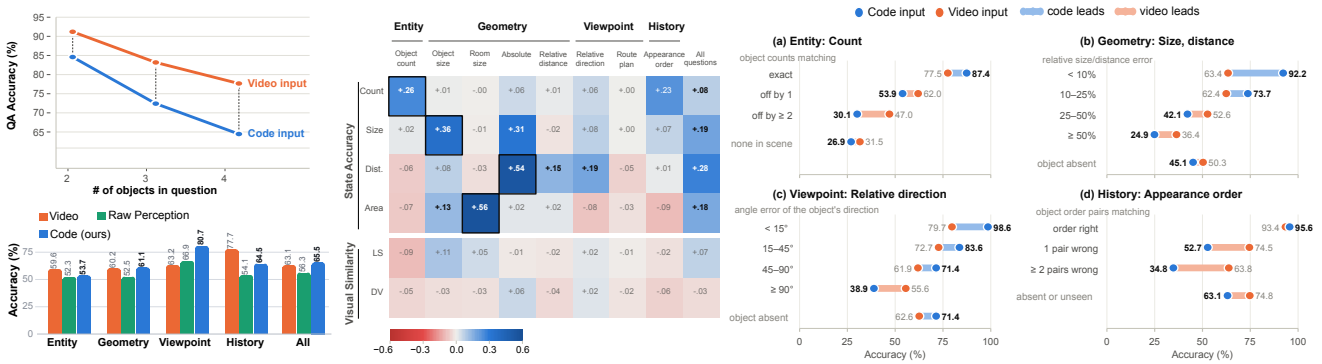

Figure 7. **Analysis of reads from executable states**. **Top left:** appearance-order acc. against k objects. State errors compound across the estimates an order combines. **Bottom left:** Gemini given the video, raw perception outputs, or our codified scene. Perception trails video when raw and leads once codified. **Middle:** Spearman correlation between scene metrics and QA acc. across the 3 codified scenes of each room. State accuracy predicts the matching questions (boxed), while visual similarity stays near zero (pale cells). **Right:** code and video accuracy by the error of our scene on the queried value. Code input tolerates 25% error in geometry and 90° in direction.

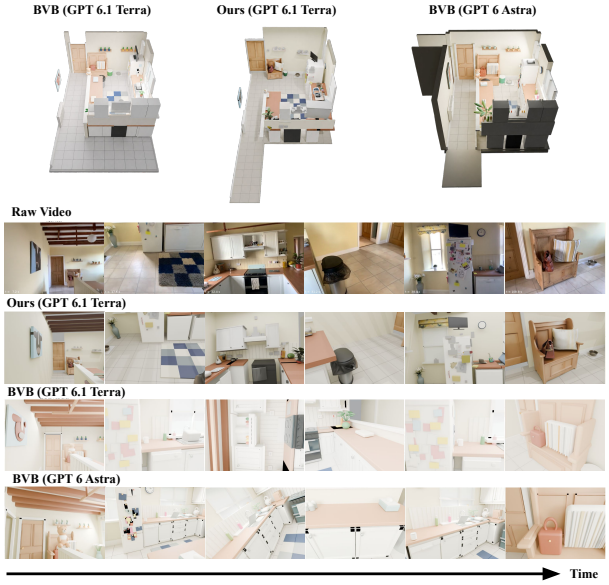

Figure 8. **Top**: Scene reconstructions from different codification methods/agents. **Bottom**: Input video frames and corresponding rendered frames. VIDEOCOS more accurately reproduces camera poses, visual details, and object states. More in Appendix Sec. E.1.

tracks the accuracy of the queried attribute (ρ of 0.26–0.56; Fig. 7, middle) but not visual similarity (+0.07 for LS and −0.03 for DV). Our scenes beat BVB's on both (57.9 vs. 52.2 and 73.9 vs. 71.7; see Sec. E.1).

Downstream tasks tolerate moderate state errors. As no reconstructed scene is exact, we check how wrong it can be before video is the better input (Fig. 7, right). For sizes and distances, code input beats video input by 28.8 points when the scene is within 10% errors of the GT and by 11.3 within 25% errors, and loses beyond that. For relative direction, it

Table 6. **Build and per-question cost**, averaged over three VSI-Debiased scenes (90 questions) with Gemini-3.8 Flash answering.

	Build (per scene)		Answer (per question)			
Method	**API ($)**	**GPU (h)**	**Calls**	**In / Out tok. (K)**	**API ($)**	**GPU (min)**
Gemini-3.8 Flash (video)	–	–	1	8.0 / 1.3	0.011	0
+ SpatialClaw [13]	–	–	13.5	660.8 / 14.5	0.180	≤1.2
+ VIDEOCOS	1.52	2.17	5.4	45.9 / 2.5	0.033	0

wins until the direction is off by more than 90°. For counts and orders, it wins only when the scene is exactly right. One miscounted object puts it 8.1 points behind video, and one swapped pair 21.8. A rough direction still picks the right side (left or right), and a close size or distance still earns partial credit, but one object off is a large error for a small count, and one swapped pair makes the whole order wrong.

Codify the perception evidence, do not pass it raw. As shown in the bottom left of Fig. 7, the same perception as raw JSON gives Gemini 56.3, below video (63.1). Per-frame depth, poses, and masks are fragments. They cannot be fused by simple reading, while our method uses tools to organize and look up this evidence, outperforming the video input.

Reuse the state across queries and generation. As listed in Tab. 6, a scene is built once for $1.52 and then answers each question for $0.033, against $0.18 for SpatialClaw [13], which restarts from the video. The build pays off after about 10 questions at similar accuracy (65.3 vs. 66.4), and on scenes with 18–48 questions our total API cost is 36–65% of SpatialClaw's. Also, the same state can be edited and re-rendered for generation.

5. Conclusion

We studied when executable world states help video reasoning and generation with VIDEOCOS, which codifies a video into a Blender program that an agent reads or writes. Such states help with geometric operations like viewpoint

changes and rigid object control, but fall short when answers combine noisy estimates or depend on unmodeled physics and appearance. They should be built for state accuracy, not visual similarity, tolerate moderate errors, and pay off across queries and generation. We hope this motivates hybrid world models pairing executable states with neural synthesis.

LLM Usage Statement

Large language models were used to assist with manuscript editing, language refinement, and code development. All scientific contributions, experimental results, and conclusions were developed and verified by the authors, who take full responsibility for the content of this paper.

References

[1] Jianhong Bai, Menghan Xia, Xiao Fu, Xintao Wang, Lianrui Mu, Jinwen Cao, Zuozhu Liu, Haoji Hu, Xiang Bai, Pengfei Wan, et al. Recammaster: Camera-controlled generative rendering from a single video. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14834–14844. IEEE, 2025. 6

[2] Ellis Brown, Jihan Yang, Shusheng Yang, Rob Fergus, and Saining Xie. Benchmark designers should" train on the test set" to expose exploitable non-visual shortcuts. *arXiv preprint arXiv:2511.04655*, 2025. 3, 4, 5

[3] Zefan Cai, Haoyi Qiu, Tianyi Ma, Haozhe Zhao, Gengze Zhou, Kung-Hsiang Huang, Parisa Kordjamshidi, Minjia Zhang, Wen Xiao, Jiuxiang Gu, et al. Mmgr: Multi-modal generative reasoning. *arXiv preprint arXiv:2512.14691*, 2025. 2, 3

[4] Chentao Cao, Zhanke Zhou, Sangni Duan, Bo Han, and Hang Li. Spatial harness: Grounding spatial reasoning in executable world states with physics engines. *arXiv preprint arXiv:2611.11344*, 2026. 3

[5] Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris Coll-Vinent, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, et al. Sam 3: Segment anything with concepts. In *International conference on learning representations*, pages 138846–138923, 2026. 2, 4

[6] Harold Haodong Chen, Rui Peng, Lingteng Qiu, Kejie Qiu, Chao Xu, and Zilong Dong. Physopsd: Code as privilege for physically plausible video generation. *arXiv preprint arXiv:2611.12411*, 2026. 3

[7] Kaihua Chen, Tarasha Khurana, and Deva Ramanan. Reconstruct, inpaint, test-time finetune: Dynamic novel-view synthesis from monocular videos. *Advances in Neural Information Processing Systems*, 38:47723–47753, 2026. 6

[8] Xingyu Chen, Fu-Jen Chu, Pierre Gleize, Kevin J Liang, Alexander Sax, Hao Tang, Weiyao Wang, Michelle Guo, Thibaut Hardin, Xiang Li, et al. Sam 3d: 3dfy anything in images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7220–7232, 2026. 2, 4

[9] Yukang Chen, Wei Huang, Baifeng Shi, Qinghao Hu, Hanrong Ye, Ligeng Zhu, Zhijian Liu, Pavlo Molchanov, Jan Kautz, Xiaojuan Qi, et al. Scaling rl to long videos. *Advances in Neural Information Processing Systems*, 38:172842–172870, 2026. 3

[10] Yiwen Chen, Guosheng Lin, and Chi Zhang. Code world model: Coding agent as world brain. *arXiv preprint arXiv:2608.25927*, 2026. 3

[11] Zhangquan Chen, Manyuan Zhang, Xinlei Yu, Xufang Luo, Mingze Sun, Zihao Pan, Xiang An, Yan Feng, Peng Pei, Xunliang Cai, et al. Think with 3d: Geometric imagination grounded spatial reasoning from limited views. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2613–2624, 2026. 3

[12] Zixu Cheng, Da Li, Jian Hu, Ziquan Liu, Wei Li, and Shaogang Gong. Graphthinker: Reinforcing video reasoning with event graph thinking. *arXiv preprint arXiv:2602.17555*, 2026. 3

[13] Seokju Cho, Ryo Hachiuma, Abhishek Badki, Hang Su, Byung-Kwan Lee, Chan Hee Song, Sifei Liu, Subhashree Radhakrishnan, Seungryong Kim, Yu-Chiang Frank Wang, et al. Spatialclaw: Rethinking action interface for agentic spatial reasoning. *arXiv preprint arXiv:2606.13673*, 2026. 3, 8

[14] Rohan Choudhury, Koichiro Niinuma, Kris M Kitani, and László A Jeni. Zero-shot video question answering with procedural programs. *arXiv preprint arXiv:2312.00937*, 2023. 3

[15] Yipeng Gao, Ziyao Zeng, Jinfa Huang, Yijiang Li, Haolin Xiong, Wang Qin, Xinlei Yu, Youheng Yao, Jie Zhang, Kun-Yu Lin, et al. 3dcodeverse: Exploring 3d worlds as code. *arXiv preprint arXiv:2610.22341*, 2026. 2, 3

[16] Google DeepMind. Gemini 3.8 Flash model card, 2026. Accessed: 2026-11-14. 6

[17] Klaus Greff, Francois Belletti, Lucas Beyer, Carl Doersch, Yilun Du, Daniel Duckworth, David J Fleet, Dan Gnanapragasam, Florian Golemo, Charles Herrmann, Thomas Kipf, Abhijit Kundu, Dmitry Lagun, Issam Laradji, Hsueh-Ti (Derek) Liu, Henning Meyer, Yishu Miao, Derek Nowrouzezahrai, Cengiz Oztireli, Etienne Pot, Noha Radwan, Daniel Rebain, Sara Sabour, Mehdi S. M. Sajjadi, Matan Sela, Vincent Sitzmann, Austin Stone, Deqing Sun, Suhani Vora, Ziyu Wang, Tianhao Wu, Kwang Moo Yi, Fangcheng Zhong, and Andrea Tagliasacchi. Kubric: a scalable dataset generator. 2022. 4, 6

[18] Tanmay Gupta and Aniruddha Kembhavi. Visual programming: Compositional visual reasoning without training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 3

[19] Qiushi Han, Keya Hu, Linlu Qiu, Cathy Wu, and Kaiming He. VISTA: A visual harness for reasoning in an interactive world, 2026. 3

[20] Zefeng He, Xiaoye Qu, Yafu Li, Tong Zhu, Siyuan Huang, and Yu Cheng. Diffthinker: Towards generative multimodal reasoning with diffusion models. *arXiv preprint arXiv:2512.24165*, 2025. 2, 3

[21] Yushi Hu, Otilia Stretcu, Chun-Ta Lu, Krishnamurthy Viswanathan, Kenji Hata, Enming Luo, Ranjay Krishna, and

Ariel Fuxman. Visual program distillation: Distilling tools and programmatic reasoning into vision-language models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9590–9601. IEEE, 2024. 3

[22] Zhening Huang, Yueyan Li, Johnathan Chiu, Xiaoyang Lyu, Matt Zhou, Yuxin Yao, Joan Lasenby, and Shangzhe Wu. Litereality-agent: An agentic system for interactable 3d indoor scene reconstruction. *arXiv preprint arXiv:2610.01863*, 2026. 3

[23] Patrick Amadeus Irawan, Iskandar Muda Rizky Parlambang, Rava Maulana, Qinrong Cui, Erland Hilman Fuadi, Zayd MK Zuhri, Nanda Ryaas Absar, Ahmed Elshabrawy, Wilfried Ariel Mulyawan, Shoubin Yu, et al. Ego2act: Evaluating goal-directed manipulation in egocentric video generation. *arXiv preprint arXiv:2610.01092*, 2026. 4, 6

[24] Daeun Lee, Shoubin Yu, Yue Zhang, and Mohit Bansal. Visioncoach: Reinforcing grounded video reasoning via visual-perception prompting. *arXiv preprint arXiv:2603.14659*, 2026. 2, 3

[25] Chenglin Li, Feng Han, Yikun Wang, Ruilin Li, Shuai Dong, Haowen Hou, Haitao Li, Qianglong Chen, Feng Tao, Jingqi Tong, et al. Videopro: Adaptive program reasoning for long video understanding. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7497–7513, 2026. 3

[26] Haodong Li, Chunmei Qing, Huanyu Zhang, Dongzhi Jiang, Yihang Zou, Hongbo Peng, Dingming Li, Yuhong Dai, ZePeng Lin, Juanxi Tian, et al. Coco: Code as cot for text-to-image preview and rare concept generation. In *European Conference on Computer Vision*, pages 57–75. Springer, 2026. 3

[27] Haodong Li, Tianfei Ren, Xiaoxiao Ma, Chunmei Qing, Zhen Fang, Sipeng He, Ziyu Guo, Haoyu Wu, Juanxi Tian, Yihang Zou, et al. Videococo: Code-as-cot for physically-consistent video generation via an agentic dual-engine system. *arXiv preprint arXiv:2607.27380*, 2026. 3

[28] Xinhao Li, Ziang Yan, Desen Meng, Lu Dong, Xiangyu Zeng, Yinan He, Yali Wang, Yu Qiao, Yi Wang, and Limin Wang. Videochat-r1: Enhancing spatio-temporal perception via reinforcement fine-tuning. *arXiv preprint arXiv:2504.06958*, 2025. 3

[29] Yifan Li, Yukai Gu, Yingqian Min, Zikang Liu, Yifan Du, Kun Zhou, Min Yang, Wayne Xin Zhao, and Minghui Qiu. Viper: Process-aware evaluation for generative video reasoning. *arXiv preprint arXiv:2512.24952*, 2025. 3

[30] Haotong Lin, Sili Chen, Junhao Liew, Donny Y Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. *arXiv preprint arXiv:2511.10647*, 2025. 2, 4

[31] Ming Liu, Yunbei Zhang, Shilong Liu, Liwen Wang, and Wensheng Zhang. Wan-r1: Verifiable-reinforcement learning for video reasoning. *arXiv preprint arXiv:2603.27866*, 2026. 3

[32] Zhanpeng Luo, Ce Zhang, Silong Yong, Cunxi Dai, Qianwei Wang, Haoxi Ran, Guanya Shi, Katia Sycara, and Yaqi Xie. pyspatial: Generating 3d visual programs for zero-shot spatial reasoning. *arXiv preprint arXiv:2603.00905*, 2026. 3

[33] Jiahao Meng, Xiangtai Li, Haochen Wang, Yue Tan, Tao Zhang, Lingdong Kong, Yunhai Tong, Anran Wang, Zhiyang Teng, Yujing Wang, et al. Open-o3 video: Grounded video reasoning with explicit spatio-temporal evidence. *arXiv preprint arXiv:2510.20579*, 2025. 3

[34] MiniMax. MiniMax H3: An open model breaking the boundaries between tasks and modalities. https://github.com/MiniMax-AI/MiniMax-H3, 2026. Open-weight omni-modal audio-video generation model. 6

[35] Saman Motamed, Laura Culp, Kevin Swersky, Priyank Jaini, and Robert Geirhos. Do generative video models understand physical principles? In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2026. 4, 6

[36] Kaleb Newman, Tyler Zhu, and Olga Russakovsky. Video models reason early: Exploiting plan commitment for maze solving. *arXiv preprint arXiv:2603.30043*, 2026. 2, 3

[37] OpenAI. GPT-5.6 Luna, 2026. Accessed: 2026-11-14. 6

[38] OpenAI. Gpt-6.1 terra model card, 2026. Accessed: 2026-11-14. 6

[39] Qwen Team. Qwen 27b model card. https://huggingface.co/Qwen, 2026. 2, 6

[40] Ruihong Shen, Žiga Kovačič, Peter Kulits, Xingrui Wang, Zizhang Li, Joshua B. Tenenbaum, Alan Yuille, Jieneng Chen, and Jiajun Wu. 4DCodeBench: Benchmarking agents on inverse graphics of dynamic scenes. Project page, 2026. 3

[41] Dídac Surís, Sachit Menon, and Carl Vondrick. ViperGPT: Visual inference via Python execution for reasoning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023. 3

[42] Jinzhou Tang, Zijun Zhang, Jing Yang, Yuchen Yan, Kun Zhou, Lingjun Mao, Ruobing Han, Jinglin Cao, Wenpeng Xu, Lukun He, et al. Video2world: Benchmarking coding agents for interactive world modeling from embodied videos. *arXiv preprint arXiv:2610.04432*, 2026. 2, 3

[43] Yolo Y. Tang, Daiki Shimada, Jiayue Meng, Jing Bi, Pinxin Liu, Yicheng Wang, Yunzhong Xiao, Zhangyun Tan, Zeliang Zhang, Chao Huang, Susan Liang, Qianxiang Shen, Luchuan Song, Ali Vosoughi, Mingqian Feng, Melika Filvantorkaman, and Chenliang Xu. Bvb: Benchmarking agentic video understanding via programmatic reconstruction in blender, 2026. 2, 3, 4, 6

[44] Jingqi Tong, Yurong Mou, Hangcheng Li, Mingzhe Li, Yongzhuo Yang, Ming Zhang, Qiguang Chen, Tianyi Liang, Xiaomeng Hu, Yining Zheng, et al. Thinking with video: Video generation as a promising multimodal reasoning paradigm. *arXiv preprint arXiv:2511.04570*, 2025. 2, 3

[45] Basile Van Hoorick, Rundi Wu, Ege Ozguroglu, Kyle Sargent, Ruoshi Liu, Pavel Tokmakov, Achal Dave, Changxi Zheng, and Carl Vondrick. Generative camera dolly: Extreme monocular dynamic novel view synthesis. In *European Conference on Computer Vision*, pages 313–331. Springer, 2024. 4, 6

[46] Binjie Wang, Xinran Zhao, Shoubin Yu, Xiang Li, Silong Yong, Chenyang Yang, Tianyu Chen, Ruofei Du, He He, Chen Zhao, et al. 3dcodeedit: Benchmarking instruction-guided 3d program editing. *arXiv preprint arXiv:2610.38231*, 2026. 3

[47] Hanyang Wang, Yimo Cai, Weiliang Chen, Jiawei Chi, Haowen Sun, Qiyu Dai, Yi-Hsin Hung, Xingzhuo Guo, Jinshan Ren, Runmao Yao, et al. Code as worlds: Agentic discovery of executable world representations for physical reasoning. *arXiv preprint arXiv:2608.27549*, 2026. 3

[48] Maijunxian Wang, Ruisi Wang, Juyi Lin, Ran Ji, Thaddäus Wiedemer, Qingying Gao, Dezhi Luo, Yaoyao Qian, Lianyu Huang, Zelong Hong, Jiahui Ge, Qianli Ma, Hang He, Yifan Zhou, Lingzi Guo, Lantao Mei, Jiachen Li, Hanwen Xing, Tianqi Zhao, Fengyuan Yu, Weihang Xiao, Yizheng Jiao, Jianheng Hou, Danyang Zhang, Pengcheng Xu, Boyang Zhong, Zehong Zhao, Gaoyun Fang, John Kitaoka, Yile Xu, Hua Xu, Kenton Blacutt, Tin Nguyen, Siyuan Song, Haoran Sun, Shaoyue Wen, Linyang He, Runming Wang, Yanzhi Wang, Mengyue Yang, Ziqiao Ma, Raphaël Millière, Freda Shi, Nuno Vasconcelos, Daniel Khashabi, Alan Yuille, Yilun Du, Ziming Liu, Bo Li, Dahua Lin, Ziwei Liu, Vikash Kumar, Yijiang Li, Lei Yang, Zhongang Cai, and Hokin Deng. A very big video reasoning suite, 2026. 2, 3

[49] Ruisi Wang, Zhongang Cai, Fanyi Pu, Junxiang Xu, Wanqi Yin, Maijunxian Wang, Ran Ji, Chenyang Gu, Bo Li, Ziqi Huang, et al. Demystifying video reasoning. *arXiv preprint arXiv:2603.16870*, 2026. 2, 3

[50] Ziyang Wang, Jaehong Yoon, Shoubin Yu, Md Mohaiminul Islam, Gedas Bertasius, and Mohit Bansal. Video-rts: Rethinking reinforcement learning and test-time scaling for efficient and enhanced video reasoning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 28114–28128, 2025. 3

[51] Thaddäus Wiedemer, Yuxuan Li, Paul Vicol, Shixiang Shane Gu, Nick Matarese, Kevin Swersky, Been Kim, Priyank Jaini, and Robert Geirhos. Video models are zero-shot learners and reasoners. *arXiv preprint arXiv:2509.20328*, 2025. 3

[52] Ziang Yan, Xinhao Li, Yinan He, Zhengrong Yue, Xiangyu Zeng, Yali Wang, Yu Qiao, Limin Wang, and Yi Wang. Videochat-r1.5: Visual test-time scaling to reinforce multimodal reasoning by iterative perception. *arXiv preprint arXiv:2509.21100*, 2025. 3

[53] Jihan Yang, Shusheng Yang, Anjali W Gupta, Rilyn Han, Li Fei-Fei, and Saining Xie. Thinking in space: How multimodal large language models see, remember, and recall spaces. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10632–10643. IEEE, 2025. 2, 3, 4, 5

[54] Junyan Ye, Jun He, Zilong Huang, Dongzhi Jiang, Xuan Yang, Rui Chen, and Weijia Li. Genclaw: Code-driven agentic image generation. *arXiv preprint arXiv:2605.30248*, 2026. 3

[55] Shaofeng Yin, Jiaxin Ge, Zora Zhiruo Wang, Chenyang Wang, Xiuyu Li, Michael J Black, Trevor Darrell, Angjoo Kanazawa, and Haiwen Feng. Vision-as-inverse-graphics agent via interleaved multimodal reasoning. In *European Conference on Computer Vision*, pages 557–575. Springer, 2026. 3

[56] Mark Yu, Wenbo Hu, Jinbo Xing, and Ying Shan. Trajectorycrafter: Redirecting camera trajectory for monocular videos via diffusion models. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 100–111. IEEE, 2025. 6

[57] Shoubin Yu, Difan Liu, Ziqiao Ma, Yicong Hong, Yang Zhou, Hao Tan, Joyce Chai, and Mohit Bansal. Veggie: Instructional editing and reasoning video concepts with grounded generation. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15147–15158. IEEE, 2025. 3

[58] Shoubin Yu, Yue Zhang, Zun Wang, Jaehong Yoon, Huaxiu Yao, Mingyu Ding, and Mohit Bansal. When and how much to imagine: Adaptive test-time scaling with world models for visual spatial reasoning. *arXiv preprint arXiv:2602.08236*, 2026. 3

[59] Xiangyu Zeng, Zhiqiu Zhang, Yuhan Zhu, Xinhao Li, Zikang Wang, Changlian Ma, Qingyu Zhang, Zizheng Huang, Kun Ouyang, Tianxiang Jiang, et al. Video-o3: Native interleaved clue seeking for long video multi-hop reasoning. *arXiv preprint arXiv:2601.23224*, 2026. 3

[60] Haoji Zhang, Xin Gu, Jiawen Li, Chixiang Ma, Sule Bai, Chubin Zhang, Bowen Zhang, Zhichao Zhou, Dongliang He, and Yansong Tang. Thinking with videos: Multimodal tool-augmented reinforcement learning for long video reasoning. *arXiv preprint arXiv:2508.04416*, 2025. 3

[61] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3813–3824. IEEE, 2023. 6

[62] Mandi Zhao, Yijia Weng, Dominik Bauer, and Shuran Song. Real2code: Reconstruct articulated objects via code generation. In *International Conference on Learning Representations*, pages 668–686, 2025. 3