AGIWORLD: AN AGENTIC VIDEO WORLD MODEL FOR BUILDING PERSISTENT WORLDS

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent interactive video world models generate visually compelling videos from a single image, following user-provided camera motion and text prompts. However, such low-level control demands substantial user effort to realize a high-level intent, and the generated world struggles to remain persistent as the exploration proceeds. In this work, we present **AgiWorld**, an agentic video world model that explores and builds a persistent world from a single image and a free-form instruction. For high-level planning, our agent **AgiWorld-Agent** designs the world and the journey through it in a world state and directs the exploration. For low-level realization, our video world model **AgiWorld-WM** generates the video chunk by chunk and stores it with its geometry in a spatial memory, from which the agent updates its world state. This closed loop keeps the world faithful to the user's intent and keeps the agent's world state consistent with the generated content throughout the exploration. We co-design the two levels around the shared spatial memory and an interface of metric camera trajectories and structured prompts. In AgiWorld-WM, to handle occlusion and accumulated geometric errors in the spatial memory over long rollouts, we propose occlusion-aware history selection, which retrieves past views by surface visibility, and reliability-based memory maintenance, which prefers reliable geometry and earlier observations. AgiWorld-WM achieves state-of-the-art overall performance on world revisiting and exploration benchmarks, and qualitative results and ablations show that AgiWorld builds persistent worlds from free-form instructions over minute-long journeys. Persistence thus arises not from the video model alone but from its closed loop with the agent.

1 INTRODUCTION

A world is more than what is currently visible. It has places beyond the current view, spatial relationships between them, and content that should remain coherent as one moves through it. Building an explorable world therefore requires two capabilities: *generating* what the camera sees and *reasoning* about the world beyond the camera.

Recent video world models have made remarkable progress on the first capability (Hong et al., 2025; Team et al., 2026; He et al., 2025c). Given an image, a camera trajectory, and a text prompt, they can generate realistic video as the camera moves through a scene. By generating successive video chunks, these models can support interactive exploration over increasingly long horizons. However, they do not decide what world should be explored. The camera trajectory and local content are still specified by the user, while content outside the current view is otherwise left to the model's generative prior. As a result, turning a high-level intent into a coherent journey still requires the user to continuously decide where to go and what should appear next.

We ask a different question: can a world model build and explore a world from a high-level instruction by itself? Doing so requires reasoning at a level above individual video chunks. Before entering an unseen region, the system must decide what should be there and how it connects to what has already been seen. After generating it, the system must treat the generated content as part of the world and reason from it thereafter. World building is therefore inherently a closed-loop process: *plan the world, generate part of it, observe what was generated, and plan again*.

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.



Figure 1: **Persistent worlds built by AgiWorld.** Given an input image and a free-form instruction, AgiWorld-Agent designs the world and the journey through it, and directs AgiWorld-WM with a camera trajectory and a text prompt for each chunk. AgiWorld-WM generates the video autoregressively and uses a point cloud as its spatial memory, which in turn is fed back to AgiWorld-Agent for further exploration. We show the rendered spatial memory with the camera route planned by the agent, and frames sampled from the long rollout videos. The results demonstrate the 3D persistence of the generated worlds as well as semantically meaningful exploration routes of the agent.

We introduce **AgiWorld**, an agentic video world model built around this loop. Given a single image and a free-form instruction, AgiWorld autonomously designs a world, plans a journey through it, and realizes the journey as a long video (Figure 1). It separates world building into two complementary components, an agent for high-level planning and a video world model for low-level realization. **AgiWorld-Agent** reasons about the world at a high level, maintaining a world state that contains both what has already been observed and what it has designed beyond the current view. **AgiWorld-WM** realizes this plan through camera-controlled video generation and maintains a spatial memory of what has actually been generated. After every generated chunk, the agent reads this memory, reconciles its plan with the generated world, and decides what should happen next.

For high-level planning, **AgiWorld-Agent**, built on a multimodal language model, keeps a world state that records the spatial layout of the world and the stages of the journey. The layout contains confirmed elements, which have already appeared in the video, and imagined elements, which the agent has designed but the camera has not yet reached. For each chunk, a director in AgiWorld-Agent turns the next stage of the journey into a camera move and writes it as a metric camera trajectory and a structured text prompt. After each chunk, the agent confirms and corrects the imagined elements that were realized, retires those that the video contradicts, and extends the design ahead.

For low-level realization, we propose **AgiWorld-WM**, a novel camera-conditioned autoregressive video generation model that follows the director's trajectories and prompts accurately and keeps the generated world consistent over long rollouts. We condition the model on metric camera trajectories through a camera adapter and a relative camera positional encoding, PRoPE (Li et al., 2025c). For spatial memory, prior work compresses historical context (Zhang et al., 2025) or retrieves past views using camera overlap (Yu et al., 2025) and spatial indexing (Li et al., 2025d). Reusing the geometry stored in our spatial memory raises two problems over long rollouts. A past view that overlaps the new camera may show surfaces that are occluded from it, and the estimated geometry accumulates errors. We therefore introduce occlusion-aware history selection, which retrieves past views by the surfaces visible from the new camera, and reliability-based memory maintenance, which prefers reliable geometry and earlier observations of each surface.

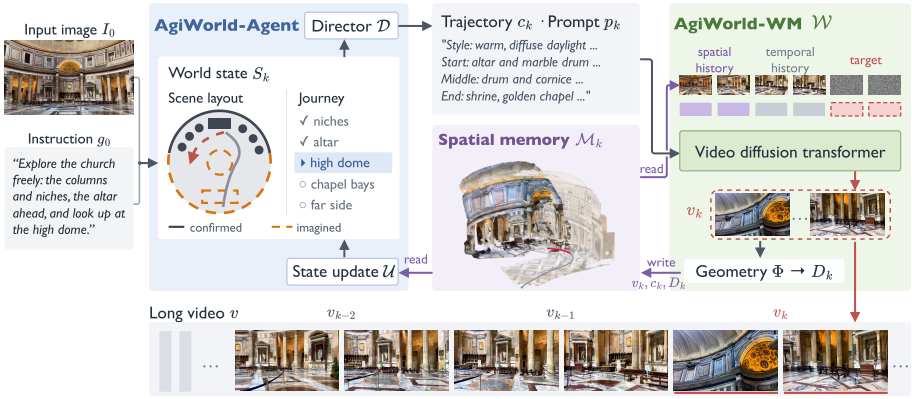

Figure 2: **Overview of AgiWorld.** The director $\mathcal{D}$ turns the world state S_k into c_k and p_k. AgiWorld-WM generates v_k from the spatial history read from its memory (purple, two of five shown) and the temporal history (gray), and writes v_k with its depth D_k into the memory, which the state update $\mathcal{U}$ reads to write S_{k+1}.

Our main contributions are as follows:

- We propose AgiWorld, a novel agentic video world model that couples an agent for high-level planning with a video world model for low-level realization.
- We introduce AgiWorld-Agent, which designs the world and the journey through it, directs the video world model to generate them, and revises this design in a closed loop.
- We design AgiWorld-WM, a camera-conditioned autoregressive video generation model with novel occlusion-aware history selection and reliability-based memory maintenance.
- Experiments show that AgiWorld-WM achieves state-of-the-art overall performance on world revisiting and exploration benchmarks, that its two memory components both improve long-term consistency, and that AgiWorld builds persistent worlds from free-form instructions over minute-long journeys.

2 System Overview

2.1 Problem Formulation

The input to AgiWorld is a single image I_0 and a free-form instruction g_0 written in natural language. The output is a long video $v = (v_1, \ldots, v_K)$ that starts from the image, builds a world around it, and travels through this world as the instruction describes. The video is generated chunk by chunk by a camera-controlled video world model $\mathcal{W}$. For chunk k, the model is conditioned on a camera trajectory $c_k = \{\mathbf{T}_{k,f}\}_{f=1}^{n}$, where $\mathbf{T}_{k,f} \in SE(3)$ is the camera pose of the f-th frame of the n frames in chunk k, a text prompt p_k that describes the content of the chunk, and the video generated so far:

$$v_k = \mathcal{W}(v_{<k},\, c_k,\, p_k),$$

where $v_{<1} = I_0$, and all camera poses are metric and expressed in the coordinates of the first frame I_0. Unlike I_0 and g_0, the trajectories and prompts are not given. They have to be constructed from g_0 as the journey proceeds, because where the camera can go and what it should see depend on what has been generated so far.

We call the generated world *persistent* when two conditions hold. *Visual persistence* concerns the appearance and geometry of the generated frames. Whenever the camera returns to a region that has been generated before, the new frames agree with the earlier ones in appearance and geometry, regardless of how many chunks have been generated in between, and the style of the scene does not drift over the journey. *Semantic persistence* concerns the content of the world and the journey. The world unfolds as the instruction g_0 describes, the scene stays coherent as it grows, and the journey through it is one continuous story rather than a series of unrelated views.

2.2 CLOSED-LOOP GENERATION

We construct the trajectories and prompts with an agent, AgiWorld-Agent, which runs in a closed loop with the world model $\mathcal{W}$, AgiWorld-WM (Figure 2). The world model is responsible for visual persistence through its spatial memory, and the agent is responsible for semantic persistence through its world state. Together, the agent and the world model form AgiWorld, an agentic video world model.

The world model organizes its history $v_{<k}$ into a spatial memory. After a chunk is generated, a geometry estimator Φ estimates the depth $D_k = \Phi(v_k, c_k)$ for its frames. The memory $\mathcal{M}_k = \{(v_j, c_j, D_j)\}_{j<k}$ holds all chunks generated before chunk k, together with their trajectories and depths. The world model and its memory can then be written as

$$v_k = \mathcal{W}(\mathcal{M}_k, c_k, p_k), \quad D_k = \Phi(v_k, c_k), \quad \mathcal{M}_{k+1} = \mathcal{M}_k \cup \{(v_k, c_k, D_k)\},$$

starting from $\mathcal{M}_1 = \{(I_0, \mathbf{T}_0, \Phi(I_0))\}$, where $\mathbf{T}_0$ is the camera pose of I_0.

The agent keeps the world in a high-level world state S_k rather than pixels. The state records the layout of the world in two parts, what has been generated and what the agent has designed but not yet generated, together with the progress of the video journey. The agent has two main functions. A director $\mathcal{D}$ directs the world model: based on the world state and the instruction, it decides the trajectory and the prompt of the next chunk. A state update $\mathcal{U}$ reads the memory of the world model after the chunk has been generated, updates the world state, and designs (or redesigns) the world and the journey ahead:

$$(c_k, p_k) = \mathcal{D}(S_k, g_0), \quad S_{k+1} = \mathcal{U}(S_k, \mathcal{M}_{k+1}, g_0),$$

starting from $S_1 = \mathcal{U}(\varnothing, \mathcal{M}_1, g_0)$, which reconstructs the visible scene from the input image into an initial world state and designs the journey and the world beyond it. The loop runs until the agent judges that the instruction g_0 has been carried out, so the number of chunks K is not fixed in advance.

3 AGIWORLD-AGENT: CLOSED-LOOP WORLD-BUILDING AGENT

The goal of AgiWorld-Agent is a journey through one world that unfolds as the instruction describes and stays coherent as it grows. This goal cannot be reached by choosing the trajectory and the prompt one chunk at a time. The world beyond the frame has to be decided before the camera reaches it, the places that the journey visits have to fit into one spatial layout, and every chunk has to advance the same journey. AgiWorld-Agent therefore keeps a persistent world state that holds the layout of the world and the stages of the journey (Section 3.1), and uses geometric tools for all geometric computation (Section 3.2). A director plans and reviews each chunk from this state (Section 3.3), and after every chunk, a state update revises the state from what was generated and extends it ahead of the camera (Section 3.4). Figure 4 in Appendix A shows one iteration of this loop, and the appendix gives the implementation details.

3.1 HIGH-LEVEL WORLD STATE

The state S_k is a compact, high-level description of the world, separate from the world model's memory $\mathcal{M}_k$ and expressed in the coordinate frame of I_0. It consists of two parts. The first is a scene layout: a set of named elements, including floors, walls, openings, and objects, each with a coarse footprint, a height, and a short appearance, and labeled either *confirmed*, meaning generated and observed, or *imagined*, meaning designed by the agent but not yet reached by the camera. The second describes the journey: a fixed visual style for the whole video, an ordered sequence of stages, each stating what the camera should show and the layout elements it involves, and the progress made through them so far. The state is coarse and semantic, which keeps it easy to confirm and extend as the world is generated, and its imagined elements let the agent design the world beyond the frame before the camera arrives.

3.2 GEOMETRIC GROUNDING

The world state layout shares its metric coordinate frame with the camera poses and the memory of the world model, so a trajectory can be planned on the layout and every element can be checked against the generated geometry. The agent, however, is built on a multimodal language model, which can reason about a scene but cannot compute with point clouds or camera poses by itself. All geometric computation is therefore done by deterministic tools. Before each query to the language model, the tools prepare what it reads: images such as a top-down map of the generated point cloud with the layout and the camera path over it, or the layout projected into the frames of a chunk, together with quantities such as the visibility of each element along a planned trajectory. After each query, the same tools check what the model wrote and return it with its errors when it fails.

3.3 DIRECTOR

The director turns the world state into the trajectory and the prompt of each chunk, $(c_k, p_k) = \mathcal{D}(S_k, g_0)$. A few waypoints are placed on the layout toward what the active stages should show, following camera practices that suit what the world model can generate, and solved into the per-frame metric poses c_k in the coordinate frame of I_0, matching the camera conditioning of Section 4.2. The prompt is then derived from this trajectory. The layout is projected along the planned poses to find the elements in view at the start of the chunk, the elements that enter or grow during it, the elements that leave the view, and the elements in view at its end. The director writes these elements into the structured format of the training captions (Appendix C), a style line fixed for the whole journey followed by the start, the middle, and the end of the chunk. The camera motion and the prompt therefore describe the same content, and consecutive chunks follow one plan. Rather than committing the whole journey at once, the director plans a bounded window ahead and extends it as the rollout proceeds.

After each chunk, the director reviews the generated chunk against the plan. The review checks whether the elements named in the prompt and those that the next stages rely on were generated as planned, and measures the camera on the geometry that the world model estimated, such as its height above the generated ground and the nearest solid ahead. A chunk whose camera fails these measurements, or that shows something else in place of an element named in the instruction, is removed from the memory and generated again from a redesigned trajectory and prompt.

3.4 WORLD DESIGN AND STATE UPDATE

The state update $S_{k+1} = \mathcal{U}(S_k, \mathcal{M}_{k+1}, g_0)$ both designs the world ahead of the camera and reconciles the state with what is generated, closing the loop between the agent and the world model. The agent designs the imagined layout and the journey for the stretch to come, so the plan always extends beyond what has been generated. Stages are closed once their content has been shown, and new stages are appended when the remaining plan runs short. The imagined layout is also kept walkable from the confirmed one, with imagined ground connected to confirmed ground and no imagined element crossing a confirmed wall, so the camera can always be routed into what the agent has designed.

Once a chunk is generated, the agent observes it directly from the world model's memory, reading the geometry the model has already estimated for it, so no separate perception step is needed. The update follows one principle: the generated video overrides the agent's design but not the user's instruction. The agent therefore updates the state, confirming the imagined elements that were realized, correcting their geometry, and retiring those the new content contradicts, so the layout agrees with the generated world. Elements seen in the input image or named in the instruction are the exception. They are not rewritten to match a generated chunk, and when the video shows something else in their place, the agent treats the generation as the error and keeps its prompts and route faithful to the instruction.

4 AGIWORLD-WM: PERSISTENT VIDEO WORLD MODEL

The goal of AgiWorld-WM is to follow the trajectory and the prompt of every chunk accurately and to keep the generated world consistent over the whole rollout, so that a region looks the same

whenever the camera returns to it. Existing camera-controlled world models generate long videos autoregressively, conditioning each iteration on recent frames (Li et al., 2025b), a compressed history (Hong et al., 2025), or past views retrieved by camera overlap (Wang et al., 2026b). Our model follows the same autoregressive scheme, in chunks of five seconds that match the unit at which the agent acts, and keeps the generated frames with their metric geometry as memory (see Section 4.1). Accurate camera following is the foundation of reliable memory. We condition the model on metric cameras through a camera adapter and a relative camera positional encoding (see Section 4.2). The difficulty then lies in keeping the memory consistent over a long rollout, despite occlusion and error accumulation in the estimated geometry. To solve these problems, we propose occlusion-aware history selection, which retrieves and injects only the history visible from the new camera, and reliability-based memory maintenance, which prefers reliable geometry and gives earlier observations precedence (see Section 4.3).

4.1 Autoregressive Video Generation

We generate the video $v = (v_1, \ldots, v_K)$ chunk by chunk with a bidirectional video diffusion model. For chunk k, the history is placed in front of the target latents of v_k as clean context, and the target is denoised conditioned on this history, on the trajectory c_k, and on the prompt p_k. The history consists of two parts, a spatial history and a temporal history. The spatial history consists of frames retrieved from the memory $\mathcal{M}_k = \{(v_j,\, c_j,\, D_j)\}_{j<k}$. Since all poses and depths share the coordinate frame of I_0, the memory forms a point cloud of the generated world, and we retrieve the frames whose points cover what the trajectory c_k is about to see. The temporal history consists of the last frames of v_{k-1}, encoded jointly with the chunk so that the video continues without a seam. In our design, the temporal history is short and is responsible only for the continuity of motion and appearance between consecutive chunks. Long-term persistence relies on the spatial history, which is retrieved from the entire memory by geometry rather than by recency. The history context has a fixed token budget independent of the rollout length, which allows generation to continue indefinitely.

4.2 Accurate Camera Following

The world model has to follow the trajectory c_k accurately. The poses in c_k are metric and in the coordinates of I_0, without the per-clip normalization common in camera-conditioned video models that discards the absolute scale of the motion. The trajectory enters the model through two paths. A camera adapter represents the relative pose between consecutive frames as a per-pixel ray embedding and modulates the features of the generated tokens in every block with a learned scale and shift. However, a generated token also needs to know the relative camera to every history token it attends to, including memory frames from distant viewpoints. We therefore add PRoPE (Li et al., 2025c) to the self-attention, which covers the full sequence and is the only camera signal that history frames receive.

4.3 Spatial Memory for Persistent Generation

Every frame in $\mathcal{M}_k$ is back-projected from its depth into metric world points, and every point is linked to the frame that observed it. Making this memory reliable requires solving three problems: (i) how to retrieve frames without letting occluded surfaces contaminate the spatial history, (ii) how to inject the geometry of the memory into the generation of new views, and (iii) how to keep the memory trustworthy at inference, where depth errors can compound across chunks.

Occlusion-aware history selection. From the point cloud of the entire generated world, we retrieve only a fixed small set of frames as the spatial history, five in our model, the ones most relevant to the trajectory c_k. Previous works may select them by camera overlap (Wang et al., 2026b), but overlap compares only camera poses and can keep frames whose surfaces are hidden from the target views. A more geometric method is to render the point cloud into the target views with a shared z-buffer (Yu et al., 2024b) and keep the frames whose points survive. But a point cloud is points, not surfaces. It has no surface orientation, so a point on the back of an object counts the same as one on its front. Its points also leave gaps, so a nearer surface cannot fully hide what lies behind it. We fix both by giving each point a normal and rendering the points as small surfels. A memory point then counts as visible from the target views along c_k only when its normal faces the camera and no nearer surface

Table 1: Quantitative comparison with camera-controlled world models on DL3DV-Evaluation (world revisiting) and World Explore (world exploration). Imag., Aes. and Text are the VBench imaging quality, aesthetic quality and text consistency scores. Trans., Rot. and Repr. are translation (m), rotation (°) and depth reprojection errors. Best in **bold**, second best underlined.

	DL3DV-Evaluation							World Explore						
Method	FID↓	Imag.↑	Aes.↑	Text↑	Trans.↓	Rot.↓	Repr.↓	FID↓	Imag.↑	Aes.↑	Text↑	Trans.↓	Rot.↓	Repr.↓
SPMem (Wu et al., 2025a)	90.26	49.72	0.420	0.186	2.98	6.39	0.085	137.78	49.96	0.397	0.163	3.91	4.00	0.072
GEN3C (Ren et al., 2025)	41.11	60.07	0.498	0.213	1.07	1.94	0.031	63.39	53.84	0.435	0.187	2.05	3.64	0.052
HY-WorldPlay (Sun et al., 2025)	81.34	56.12	0.395	0.152	3.08	5.36	0.049	103.03	54.50	0.378	0.127	4.01	6.04	0.095
LingBot-World (Team et al., 2026)	69.05	61.90	0.459	0.177	2.74	7.65	0.073	57.95	67.51	0.471	0.181	3.36	4.35	0.075
Matrix-Game 3.0 (Wang et al., 2026b)	46.50	70.09	0.477	0.191	2.09	6.64	0.073	51.73	69.35	0.462	0.177	2.46	2.02	0.065
Lyra 2.0 (Shen et al., 2026)	25.54	73.02	0.486	0.232	0.34	**0.63**	0.021	42.73	72.05	0.461	0.216	1.32	1.42	0.037
SANA-WM (Zhu et al., 2026)	48.32	72.01	**0.547**	0.215	1.28	2.69	0.038	65.31	70.15	0.467	0.195	1.87	1.61	0.045
Ours	**21.74**	**73.54**	0.524	**0.236**	**0.29**	0.99	**0.020**	**35.37**	**72.65**	**0.477**	**0.223**	**1.05**	**1.27**	**0.032**

hides it. Given this occlusion-aware visibility test, the spatial history is built by a greedy set cover (see Appendix B for details).

Geometry correspondence injection. The spatial history selected is placed in the sequence, but this alone leaves the token-level correspondence between the memory latents and the target latents implicit. Conditioning on a frame warped into the target view (Seo et al., 2024) or on a rendered point cloud (Huang et al., 2025a) would make it explicit, but the model then reproduces the depth errors, holes, and stretched regions of these images in the generated frames. We instead inject an index rather than the source appearance, following coordinate-based geometry conditioning (Bai et al., 2026; Shen et al., 2026). Concretely, we use two kinds of sources: the spatial history frames and the last frame of the temporal history. From each source, we warp a per-pixel index map and depth into every generated frame, linking each generated pixel that a source reaches to its source pixel. We use the same occlusion-aware visibility test for retrieval and injection, so the two stay consistent. In each transformer block, an encoder turns this correspondence into a bias on the query and key inputs of the generated tokens, steering each token toward the memory it matches.

Reliability-based memory maintenance. Both mechanisms above rely on a memory whose geometry is accurate and consistent. In training this is guaranteed: the geometry of each training video is estimated offline over its full length and filtered for quality, so the memory is stable. At inference, however, the memory is built incrementally from the model's own outputs. A small drift in camera following leaves a chunk's geometry slightly inconsistent, the next chunk is generated on top of it and drifts further, and the error compounds, accumulating over the rollout. We therefore extend the occlusion-aware visibility test into a memory-maintenance step at inference to decide which pixels the model should trust across the many history frames and the point cloud they form. We introduce two judgments for this, on a pixel's reliability and on its generation time. The reliability judgment tests three conditions for each point: the confidence from the geometry estimator, the angle at which it was observed, and its distance from the observing camera. Reliable points are used first in both retrieval and injection, and less reliable ones fill only the gaps they leave. The generation-time judgment trusts earlier content: when a later point lies where an earlier frame reliably observed empty space, we keep the earlier observation and drop the later, drifted point. Later generation can then extend the world but not overwrite it, which curbs drift (see Appendix B for details).

5 Experiments

5.1 World Model Evaluation

Benchmarks. We evaluate long-term camera-conditioned video generation on two benchmarks. We evaluate *world revisiting* on DL3DV-Evaluation (Ling et al., 2024), whose 60-second camera trajectories circle the scene and return to where they started. Besides, we introduce another benchmark, World Explore, to evaluate the model's capability of *world exploration*, where the camera keeps moving into unseen parts of the scene, so most of the video is generated beyond the input image and has to stay consistent with the frames before it. World Explore consists of 40 real-world outdoor scenes held out from an internal collection and never used in training, each with a 60-second camera trajectory. Appendix D.3 describes the World Explore benchmark (Figure 7), and Appendix D.2 describes the metrics and the baselines.



Figure 3: Camera-controlled long video generation comparison on DL3DV-Evaluation and World Explore. The first column shows the input image with a top view of the target camera trajectory.

Quantitative results. We show the quantitative comparisons in Table 1. Our model achieves the best results on both benchmarks across nearly all metrics. For camera following and geometric consistency, our model has the lowest translation and reprojection errors on both benchmarks, and reduces the translation error on World Explore from 1.32 m of the strongest baseline to 1.05 m. For visual quality, our model has the lowest FID and the highest imaging quality on both benchmarks, and reduces FID from 25.54 to 21.74 on DL3DV-Evaluation and from 42.73 to 35.37 on World Explore. On DL3DV-Evaluation, the generated frames are compared with the real scene along a trajectory that returns to earlier viewpoints, so this gain also reflects the long-term memory of our model. Our model also has the highest text consistency on both benchmarks, a property that the director in Section 3 relies on when prompting the model. (See Figure 3, and Figure 11 in Appendix D.5, for qualitative comparisons.)

Ablation. Table 7 in Appendix D.4 ablates the two memory components. The full model is the best on all metrics in every time window, and removing both components raises FID over the last 10 seconds from 51.84 to 57.74.

5.2 Agentic World Generation

We now evaluate the full AgiWorld, with AgiWorld-Agent directing the world model.

Demonstration. Figure 1 shows three worlds built by AgiWorld. In the 80-second temple video, the camera moves along the main hall, into it, back out and into the right-hand passage, as the instruction asks. Inside the hall, which the input image shows only from outside, AgiWorld generates the gilded statue named in the instruction, and the courtyard reappears with the same lanterns and trees. Full videos are in the supplementary material.

Ablation of AgiWorld-Agent. We ablate different modules of AgiWorld-Agent with the same input and world model (Figure 9 in Appendix D.5): *w/o agent* plans the whole trajectory once from the input image and uses the instruction as every prompt, *w/o world state* plans each chunk from what has been generated so far, *w/o state update* generates the whole video from the initial plan without review, and *w/o geometric grounding* uses neither the point cloud nor the geometric tools. In the temple, without the world state the objects in the hall change between chunks and the camera returns to a facade with two doorways instead of one, and without geometric grounding the camera enters a gallery deeper than the building and never leaves it. In the bookstore, w/o agent ends in the road and w/o state update stays under a wooden structure at the shopfront. The full system carries out both instructions. The world state thus keeps the design consistent across chunks, geometric grounding matches the plan to the generated geometry, and the state update lets the agent correct its plan after each chunk.

Replacing the world model. We also keep AgiWorld-Agent and replace only AgiWorld-WM with LingBot-World (Team et al., 2026), Matrix-Game 3.0 (Wang et al., 2026b) or Lyra 2.0 (Shen et al., 2026) (Figure 10 in Appendix D.5). The agent directs all three models, but only with AgiWorld-WM do the frames stay in the courtyard of the input image and keep its lighting. Visual persistence thus comes from AgiWorld-WM, which is co-designed with the agent.

6 Related Work

Camera-controlled video generation. Camera-controlled video generation conditions a pretrained video diffusion model on a camera trajectory, and existing methods differ mainly in how the trajectory enters the network. Early methods inject the pose parameters directly, either as flattened extrinsics (Wang et al., 2024) or as per-pixel Plücker rays (Sitzmann et al., 2021; Chen et al., 2023) passed through an adapter (He et al., 2025a; Xu et al., 2024; Bahmani et al., 2025a;b; He et al., 2025b), and recent interactive world models (He et al., 2025c; Li et al., 2025b; Mao et al., 2026; Hong et al., 2025) expose camera control as keyboard actions that are mapped into camera space. Geometry-guided methods instead condition on a rendering of the estimated geometry at the target view, a point cloud lifted from the input frames (Yu et al., 2024b; Ren et al., 2025; Huang et al., 2025a; Zhao et al., 2025) or a depth-warped frame (Li et al., 2025a). Their precision depends on the estimated geometry, and warping introduces disocclusion holes and errors at depth boundaries (Seo et al., 2024) that the model tends to reproduce. A third line modifies the positional encoding of the transformer itself (Miyato et al., 2024; Li et al., 2025c; Bai et al., 2025b; 2026). Relative camera encodings such as PRoPE (Li et al., 2025c) require no fixed reference frame and generalize to unseen sequence lengths.

Long video generation. Persistence has been a long-standing difficulty in long video generation. Autoregressive video diffusion models (Chen et al., 2024; Huang et al., 2025b) attend only to a bounded window of recent frames, so a region drifts in content, appearance, and geometry once it leaves the window. Later methods extend the window, either by compressing the history into the context (Zhang et al., 2025; Hong et al., 2025; Wu et al., 2026) or by adaptively selecting a sparse subset of the history as context (Yi et al., 2026; Cai et al., 2025; Xu et al., 2026), but the history is still organized by time and selected by recency or coarse camera correspondence, so a revisited region is not guaranteed to match what was generated before. Another line of methods organizes the history spatially. Retrieval approaches keep past frames and select those that overlap the current view (Yu et al., 2025; Xiao et al., 2025; Sun et al., 2025; Wang et al., 2026b; Li et al., 2025d), and explicit approaches accumulate a 3D representation and render it into the target view (Wu et al., 2025a; Huang et al., 2025a). Occlusion and noise in the retrieved or reconstructed content remain difficult to handle, and geometric errors accumulate over long rollouts.

Agents for world generation. LLM agents can pursue a goal by planning, calling tools, and revising the plan based on the outcomes of their actions (Yao et al., 2023; Park et al., 2023). These abilities have been brought to content generation. In 3D scene generation, agents plan the scene as object layouts that are then populated with assets (Feng et al., 2023; Yang et al., 2024; Hu et al., 2024; Ling et al., 2025), and in video generation, agents expand a prompt into multi-scene plans that a video generator renders scene by scene (Lin et al., 2023; Wu et al., 2025b). Moreover, agents can take control of a visual generation model to automate world generation, deciding what should appear next and steering the generator through text conditions (Yu et al., 2024a; Feng et al., 2025; Gao et al., 2026). WonderJourney (Yu et al., 2024a) generates each scene along a journey as a point cloud from LLM-written descriptions and verifies it with a VLM, and LingBot-World 2.0 (Gao et al., 2026) pairs its autoregressive video generator with an agent that provides the input actions and next-event control signals in place of the user.

7 Conclusion

In this work, we proposed AgiWorld, a novel agentic video world model that builds a persistent world from a single image and a free-form instruction. AgiWorld-Agent designs the world and the journey through it in a high-level world state, directs the world model with metric camera trajectories and structured prompts, and revises its design in a closed loop. AgiWorld-WM keeps its spatial memory consistent over long rollouts through novel occlusion-aware history selection and reliability-based memory maintenance. The two are co-designed around this shared spatial memory and the interface between them. AgiWorld-WM achieves state-of-the-art overall performance on world revisiting and exploration benchmarks, and AgiWorld builds persistent worlds that follow free-form instructions over minute-long journeys. We believe that AgiWorld opens a new direction for video world models, in which an agent plans the world and directs the world model to generate it, instead of a user steering the camera chunk by chunk.

AI USE STATEMENT

Language models are part of the proposed system. AgiWorld-Agent is built on a multimodal language model, GPT-6-Astra in all our experiments (Section 3 and Appendix A), and Qwen3-VL (Bai et al., 2025a) writes the captions of the training and evaluation videos (Appendix C). Beyond these system components, we used LLM-based assistants to edit the text of the paper, to draft figure layouts, to assist in writing implementation code, and to build the HTML pages in the supplementary material that visualize our results. All AI-assisted text, figures, code and pages were reviewed by the authors, and the code was tested before it was used for any reported result. All reported numbers come from our own runs. We take responsibility for the final content of this work, including text, claims and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

Our training data combine public datasets, used under their licenses, with an internal collection of indoor and outdoor scene captures. During data filtering we remove clips with moving people and vehicles (Appendix C), and the generated worlds are static scenes. This work involves no human subjects. Like other video generation models, AgiWorld can produce realistic videos of places that do not exist, or new views of places shown in a real photograph, which could be used to create misleading footage. We intend the system for world building and content creation, and we do not claim that its outputs depict real places faithfully.

REPRODUCIBILITY STATEMENT

Section 3 and Appendix A describe AgiWorld-Agent, including its language-model steps, the world state, the geometric tools, the thresholds of every check, and the order of all calls (Algorithm 1). The full guidelines, schemas, and specification will be released with the code. Section 4 and Appendix B describe AgiWorld-WM and the parameters of its memory mechanisms. Appendix C describes the data pipeline and the training stages. Appendix D.1 gives the training settings, and Appendix D.3 describes the World Explore benchmark.

REFERENCES

Sherwin Bahmani, Ivan Skorokhodov, Guocheng Qian, Aliaksandr Siarohin, Willi Menapace, Andrea Tagliasacchi, David B. Lindell, and Sergey Tulyakov. AC3D: Analyzing and improving 3D camera control in video diffusion transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22875–22889, 2025a.

Sherwin Bahmani, Ivan Skorokhodov, Aliaksandr Siarohin, Willi Menapace, Guocheng Qian, Michael Vasilkovsky, Hsin-Ying Lee, Chaoyang Wang, Jiaxu Zou, Andrea Tagliasacchi, David B. Lindell, and Sergey Tulyakov. VD3D: Taming large video diffusion transformers for 3D camera control. In *The Thirteenth International Conference on Learning Representations*, 2025b.

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*, 2025a.

Yunpeng Bai, Haoxiang Li, and Qixing Huang. Positional encoding field. *arXiv preprint arXiv:2510.20385*, 2025b.

Yunpeng Bai, Haoxiang Li, and Qixing Huang. PE-Field 4D: Video generation models as canvas. *arXiv preprint arXiv:2607.15667*, 2026.

Shengqu Cai, Ceyuan Yang, Lvmin Zhang, Yuwei Guo, Junfei Xiao, Ziyan Yang, Yinghao Xu, Zhenheng Yang, Alan Yuille, Leonidas Guibas, Maneesh Agrawala, Lu Jiang, and Gordon Wetzstein. Mixture of contexts for long video generation. *arXiv preprint arXiv:2508.21058*, 2025.

Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Eric Ming Chen, Sidhanth Holalkere, Ruyu Yan, Kai Zhang, and Abe Davis. Ray conditioning: Trading photo-consistency for photo-realism in multi-view image generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 23185–23194, 2023.

Hao Feng, Zhi Zuo, Jia-Hui Pan, Ka-Hei Hui, Qi Dou, Jingyu Hu, and Zhengzhe Liu. WonderVerse: Extendable 3D scene generation with video generative models. *arXiv preprint arXiv:2503.09160*, 2025.

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

Zelin Gao, Qiuyu Wang, Jiapeng Zhu, Jingye Chen, Zichen Liu, Qingyan Bai, Jiahao Wang, Yufeng Yuan, Hanlin Wang, Yichong Lu, Ka Leong Cheng, Haojie Zhang, Jian Gao, Tianrui Feng, Yuzheng Liu, Yao Yao, Yinghao Xu, Xing Zhu, Yujun Shen, and Hao Ouyang. Infinite worlds with versatile interactions. *arXiv preprint arXiv:2607.07534*, 2026.

Hao He, Yinghao Xu, Yuwei Guo, Gordon Wetzstein, Bo Dai, Hongsheng Li, and Ceyuan Yang. CameraCtrl: Enabling camera control for video diffusion models. In *The Thirteenth International Conference on Learning Representations*, 2025a.

Hao He, Ceyuan Yang, Shanchuan Lin, Yinghao Xu, Meng Wei, Liangke Gui, Qi Zhao, Gordon Wetzstein, Lu Jiang, and Hongsheng Li. CameraCtrl II: Dynamic scene exploration via camera-controlled video diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 13416–13426, 2025b.

Xianglong He, Chunli Peng, Zexiang Liu, Boyang Wang, Yifan Zhang, Qi Cui, Fei Kang, Biao Jiang, Mengyin An, Yangyang Ren, Baixin Xu, Hao-Xiang Guo, Kaixiong Gong, Size Wu, Wei Li, Xuchen Song, Yang Liu, Yangguang Li, and Yahui Zhou. Matrix-game 2.0: An open-source real-time and streaming interactive world model. *arXiv preprint arXiv:2508.13009*, 2025c.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

Yicong Hong, Yiqun Mei, Chongjian Ge, Yiran Xu, Yang Zhou, Sai Bi, Yannick Hold-Geoffroy, Mike Roberts, Matthew Fisher, Eli Shechtman, Kalyan Sunkavalli, Feng Liu, Zhengqi Li, and Hao Tan. RELIC: Interactive video world model with long-horizon memory. *arXiv preprint arXiv:2512.04040*, 2025.

Ziniu Hu, Ahmet Iscen, Aashi Jain, Thomas Kipf, Yisong Yue, David A. Ross, Cordelia Schmid, and Alireza Fathi. SceneCraft: An LLM agent for synthesizing 3D scene as blender code. In *International Conference on Machine Learning*, pp. 19252–19282, 2024.

Tianyu Huang, Wangguandong Zheng, Tengfei Wang, Yuhao Liu, Zhenwei Wang, Junta Wu, Jie Jiang, Hui Li, Rynson W. H. Lau, Wangmeng Zuo, and Chunchao Guo. Voyager: Long-range and world-consistent video diffusion for explorable 3D scene generation. *ACM Transactions on Graphics (TOG)*, 44(6):1–15, 2025a.

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. In *Advances in Neural Information Processing Systems*, volume 38, 2025b.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. Vbench: Comprehensive benchmark suite for video generative models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21807–21818. IEEE, 2024.

Guangyuan Li, Bo Li, Jinwei Chen, Xiaobin Hu, Lei Zhao, and Peng-Tao Jiang. MagicWorld: Towards long-horizon stability for interactive video world exploration. *arXiv preprint arXiv:2511.18886*, 2025a.

Jiaqi Li, Junshu Tang, Zhiyong Xu, Longhuang Wu, Yuan Zhou, Shuai Shao, Tianbao Yu, Zhiguo Cao, and Qinglin Lu. Hunyuan-GameCraft: High-dynamic interactive game video generation with hybrid history condition. *arXiv preprint arXiv:2506.17201*, 2025b.

Ruilong Li, Brent Yi, Junchen Liu, Hang Gao, Yi Ma, and Angjoo Kanazawa. Cameras as relative positional encoding. In *Advances in Neural Information Processing Systems*, volume 38, 2025c.

Runjia Li, Philip Torr, Andrea Vedaldi, and Tomas Jakab. VMem: Consistent interactive video scene generation with surfel-indexed view memory. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 25690–25699, 2025d.

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. *Advances in Neural Information Processing Systems*, 38, 2026.

Han Lin, Abhay Zala, Jaemin Cho, and Mohit Bansal. VideoDirectorGPT: Consistent multi-scene video generation via LLM-Guided planning. *arXiv preprint arXiv:2309.15091*, 2023.

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.

Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22160–22169. IEEE, 2024.

Lu Ling, Chen-Hsuan Lin, Tsung-Yi Lin, Yifan Ding, Yu Zeng, Yichen Sheng, Yunhao Ge, Ming-Yu Liu, Aniket Bera, and Zhaoshuo Li. Scenethesis: A language and vision agentic framework for 3D scene generation. *arXiv preprint arXiv:2505.02836*, 2025.

Xiaofeng Mao, Zhen Li, Chuanhao Li, Xiaojie Xu, Kaining Ying, and Kaipeng Zhang. Yume-1.5: A text-controlled interactive world generation model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7752–7761, 2026.

Takeru Miyato, Bernhard Jaeger, Max Welling, and Andreas Geiger. GTA: A geometry-aware attention mechanism for multi-view transformers. In *The Twelfth International Conference on Learning Representations*, 2024.

Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, pp. 2:1–2:22, 2023.

Manthan Patel, Fan Yang, Yuheng Qiu, Cesar Cadena, Sebastian Scherer, Marco Hutter, and Wenshan Wang. Tartanground: A large-scale dataset for ground robot perception and navigation. *arXiv preprint arXiv:2505.10696*, 2025.

Xuanchi Ren, Tianchang Shen, Jiahui Huang, Huan Ling, Yifan Lu, Merlin Nimier-David, Thomas Müller, Alexander Keller, Sanja Fidler, and Jun Gao. GEN3C: 3D-Informed world-consistent video generation with precise camera control. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6121–6132, 2025.

Junyoung Seo, Kazumi Fukuda, Takashi Shibuya, Takuya Narihira, Naoki Murata, Shoukang Hu, Chieh-Hsin Lai, Seungryong Kim, and Yuki Mitsufuji. GenWarp: Single image to novel views with semantic-preserving generative warping. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Tianchang Shen, Sherwin Bahmani, Kai He, Sangeetha Grama Srinivasan, Tianshi Cao, Jiawei Ren, Ruilong Li, Zian Wang, Nicholas Sharp, Zan Gojcic, Sanja Fidler, Jiahui Huang, Huan Ling, Jun Gao, and Xuanchi Ren. Lyra 2.0: Explorable generative 3D worlds. *arXiv preprint arXiv:2604.13036*, 2026.

Vincent Sitzmann, Semon Rezchikov, William T. Freeman, Joshua B. Tenenbaum, and Frédo Durand. Light field networks: Neural scene representations with single-evaluation rendering. In *Advances in Neural Information Processing Systems*, volume 34, pp. 19313–19325, 2021.

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.

Rawmantic Team. Rawmantic 3d training field: From the real world, for the real world. *Rawmantic*, 2026.

Robbyant Team, Zelin Gao, Qiuyu Wang, Yanhong Zeng, Jiapeng Zhu, Ka Leong Cheng, Yixuan Li, Hanlin Wang, Yinghao Xu, Shuailei Ma, et al. Advancing open-source world models. *arXiv preprint arXiv:2601.20540*, 2026.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 42592–42603, 2026a.

Wenshan Wang, Delong Zhu, Xiangwei Wang, Yaoyu Hu, Yuheng Qiu, Chen Wang, Yafei Hu, Ashish Kapoor, and Sebastian Scherer. Tartanair: A dataset to push the limits of visual slam. 2020.

Zhouxia Wang, Ziyang Yuan, Xintao Wang, Yaowei Li, Tianshui Chen, Menghan Xia, Ping Luo, and Ying Shan. MotionCtrl: A unified and flexible motion controller for video generation. In *ACM SIGGRAPH 2024 Conference Papers*, pp. 1–11, 2024.

Zile Wang, Zexiang Liu, Jiaxing Li, Kaichen Huang, Baixin Xu, Fei Kang, Mengyin An, Peiyu Wang, Biao Jiang, Yichen Wei, Yidan Xietian, Jiangbo Pei, Liang Hu, Boyi Jiang, Hua Xue, Zidong Wang, Haofeng Sun, Wei Li, Wanli Ouyang, Xianglong He, Yang Liu, Yangguang Li, and Yahui Zhou. Matrix-game 3.0: Real-time and streaming interactive world model with long-horizon memory. *arXiv preprint arXiv:2604.08995*, 2026b.

Ruiqi Wu, Xuanhua He, Meng Cheng, Tianyu Yang, Yong Zhang, Zhuoliang Kang, Xunliang Cai, Xiaoming Wei, Chunle Guo, Chongyi Li, and Ming-Ming Cheng. Infinite-world: Scaling interactive world models to 1000-frame horizons via pose-free hierarchical memory. *arXiv preprint arXiv:2602.02393*, 2026.

Tong Wu, Shuai Yang, Ryan Po, Yinghao Xu, Ziwei Liu, Dahua Lin, and Gordon Wetzstein. Video world models with long-term spatial memory. In *Advances in Neural Information Processing Systems*, volume 38, 2025a.

Weijia Wu, Zeyu Zhu, and Mike Zheng Shou. Automated movie generation via multi-agent CoT planning. *arXiv preprint arXiv:2503.07314*, 2025b.

Zeqi Xiao, Yushi Lan, Yifan Zhou, Wenqi Ouyang, Shuai Yang, Yanhong Zeng, and Xingang Pan. WorldMem: Long-term consistent world simulation with memory. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Dejia Xu, Weili Nie, Chao Liu, Sifei Liu, Jan Kautz, Zhangyang Wang, and Arash Vahdat. CamCo: Camera-controllable 3D-Consistent image-to-video generation. *arXiv preprint arXiv:2406.02509*, 2024.

Jiacong Xu, Hanwen Jiang, Zhixin Shu, Kalyan Sunkavalli, Vishal M. Patel, and Yiqun Mei. Wonder: Video world model done better. *arXiv preprint arXiv:2607.26037*, 2026.

Yue Yang, Fan-Yun Sun, Luca Weihs, Eli VanderBilt, Alvaro Herrasti, Winson Han, Jiajun Wu, Nick Haber, Ranjay Krishna, Lingjie Liu, Chris Callison-Burch, Mark Yatskar, Aniruddha Kembhavi, and Christopher Clark. Holodeck: Language guided generation of 3D embodied AI environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16277–16287, 2024.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023.

Jung Yi, Minjae Kim, Paul Hyunbin Cho, Wooseok Jang, Sangdoo Yun, and Seungryong Kim. WorldKV: Efficient world memory with world retrieval and compression. *arXiv preprint arXiv:2605.22718*, 2026.

Hong-Xing Yu, Haoyi Duan, Junhwa Hur, Kyle Sargent, Michael Rubinstein, William T. Freeman, Forrester Cole, Deqing Sun, Noah Snavely, Jiajun Wu, and Charles Herrmann. WonderJourney: Going from anywhere to everywhere. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6658–6667, 2024a.

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 Conference Papers*, pp. 19:1–19:11, 2025.

Wangbo Yu, Jinbo Xing, Li Yuan, Wenbo Hu, Xiaoyu Li, Zhipeng Huang, Xiangjun Gao, Tien-Tsin Wong, Ying Shan, and Yonghong Tian. ViewCrafter: Taming video diffusion models for high-fidelity novel view synthesis. *arXiv preprint arXiv:2409.02048*, 2024b.

Lvmin Zhang, Shengqu Cai, Muyang Li, Gordon Wetzstein, and Maneesh Agrawala. Frame context packing and drift prevention in next-frame-prediction video diffusion models. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Jinjing Zhao, Fangyun Wei, Zhening Liu, Hongyang Zhang, Chang Xu, and Yan Lu. Spatia: Video generation with updatable spatial memory. *arXiv preprint arXiv:2512.15716*, 2025.

Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *arXiv preprint arXiv:1805.09817*, 2018.

Haoyi Zhu, Haozhe Liu, Yuyang Zhao, Tian Ye, Junsong Chen, Jincheng Yu, Tong He, Song Han, and Enze Xie. Sana-wm: Efficient minute-scale world modeling with hybrid linear diffusion transformer. *arXiv preprint arXiv:2605.15178*, 2026.

A AgiWorld-Agent Details

This section gives the implementation details of AgiWorld-Agent (Section 3), in the order of the main text.

A.1 Overview

Section 2.2 defines the two functions of AgiWorld-Agent. The director $\mathcal{D}$ turns the world state into the trajectory and the prompt of each chunk, and the state update $\mathcal{U}$ updates the world state from the memory of AgiWorld-WM and designs the world and the journey ahead. Both functions are carried out by one multimodal language model, GPT-6-Astra in all our experiments, together with the geometric tools. The agent does not depend on this model, and other multimodal language models, such as Claude Opus, Claude Fable, and GPT-5.6, can take its place. The language model is called in separate steps, each with its own guidelines and output schema (Table 2). The tools prepare the inputs of each step and check its output.

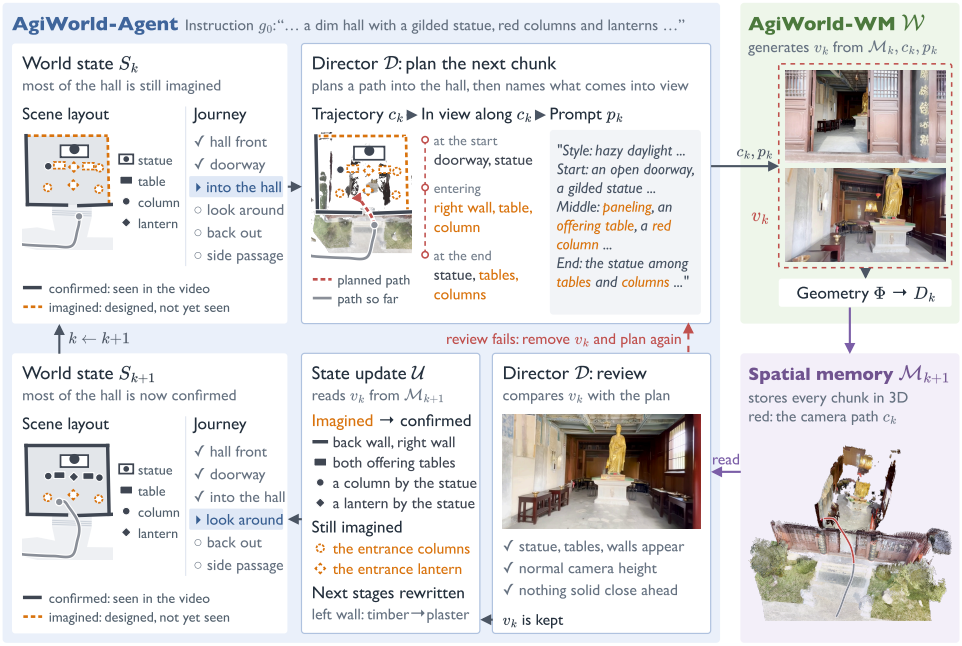

Figure 4: All decisions and state changes of AgiWorld-Agent for one chunk, from the temple example in Figure 1: the director plans the chunk, AgiWorld-WM generates it, the director reviews it, and the state update turns S_k into S_{k+1}.

Director. The director $\mathcal{D}$ consists of four steps. Three of them plan the route. The trajectory step writes waypoints for the next 60 s, and the tools solve them into a camera path. The fix step corrects the waypoints when this path fails a check, and the prompt step writes the prompts of the chunks along the path. These three steps run at initialization and again whenever the route has to change, which we call a replan. The fourth step, the review, checks each generated chunk against the plan and reads the camera measurements that the tools compute on it. A chunk whose camera fails these measurements, or that shows something else in place of an element named in the instruction, is generated again. The small edits of the layout and the stages that the review returns are passed to the state update.

State update. The state update $\mathcal{U}$ has two parts. Reconstruction writes the confirmed layout from the input image and the generated geometry, and design writes the journey and the imagined layout. Each part has one step at initialization and one step that can run after each chunk. At initialization, the state update $S_1 = \mathcal{U}(\varnothing, \mathcal{M}_1, g_0)$ runs the initial reconstruction of the scene visible in I_0, which reads the ground, writes the layout, and verifies it. The initial design of the journey and of the world beyond the image follows. After each chunk, the state update $S_{k+1} = \mathcal{U}(S_k, \mathcal{M}_{k+1}, g_0)$ applies the edits of the review, revises the confirmed layout when the chunk contradicts it, which we call a revision, and rewrites the stages and the imagined layout when they have to change, which we call a refresh.

Queries. Each query contains the guidelines of the step, the images and data that the geometric tools prepared for it, and a JSON output schema. The queries of the initial design, the refresh, and the trajectory step also contain a shared specification of AgiWorld-WM.

Validation. The tools check each output against its schema and against the checks of its step. An output that fails is returned with a list of its errors, and the model answers again. When a revision, a refresh, or a replan still fails, the loop keeps the previous state and continues. A path that still violates a bound after the fix step is the exception and is accepted.

Order of calls. Algorithm 1 gives the order of all steps. The review and the state update run after every chunk. Regeneration, revision, refresh, and replan run only when their triggers fire.

Table 2: The steps of AgiWorld-Agent, grouped by the function they carry out. Each step has its own guidelines and output schema.

Step	Called	Writes
Director $\mathcal{D}$		
Trajectory	At initialization, and whenever the route has to change (replan)	Waypoints of the next 60 s
Fix	After Trajectory, when the solved path fails a check	Corrected waypoints
Prompt	After Trajectory and Fix	Start, middle, and end of the prompt of each chunk
Review	After every chunk	Verdicts, small layout edits, stage updates, decision for the next chunk
State update $\mathcal{U}$: reconstruction		
Initial reconstruction	At initialization: ground reading, layout, verification	Standing surface and other marked surfaces, confirmed layout
Revision	Major mismatch, contradicted target, or ground offset	Confirmed layout, contradicted imagined elements
State update $\mathcal{U}$: design		
Initial design	At initialization	Style, stages, imagined layout
Refresh	Stage status changed, less than 30 s of path left, layout revised, or elements retired	Stages, edits of the imagined layout, finished flag

Algorithm 1 The generation loop of AgiWorld. LM marks calls to the language model, WM marks calls to AgiWorld-WM and its geometry estimator Φ, and tools marks deterministic computation.

Require: image I_0, instruction g_0
1: $\mathcal{M}_1 \leftarrow \{(I_0, \mathbf{T}_0, \Phi(I_0))\}$ ▷ WM
2: Initial reconstruction: read the ground, write and verify the confirmed layout ▷ LM, tools
3: Initial design: write the style, the stages, and the imagined layout ▷ LM, tools
4: $S_1 \leftarrow \mathcal{U}(\varnothing, \mathcal{M}_1, g_0)$, the state written in lines 2–3
5: Director: trajectory, fix, and prompt steps for the first 60 s ▷ LM, tools
6: $k \leftarrow 1$
7: **repeat**
8: $(c_k, p_k) \leftarrow \mathcal{D}(S_k, g_0)$, taken from the planned path and prompts
9: $v_k \leftarrow \mathcal{W}(\mathcal{M}_k, c_k, p_k)$ ▷ WM
10: $D_k \leftarrow \Phi(v_k, c_k)$, $\mathcal{M}_{k+1} \leftarrow \mathcal{M}_k \cup \{(v_k, c_k, D_k)\}$ ▷ WM
11: Measure the camera on the generated geometry ▷ tools
12: Director: review v_k ▷ LM
13: **if** $k > 1$ and a regeneration check fires **then** ▷ tools
14: Restore $\mathcal{M}_k$ and S_k ▷ WM, tools
15: Director: replan from the end of chunk $k-1$, given the first review ▷ LM, tools
16: Repeat lines 8–12 once, with a new seed
17: If the review rewrote the next prompt, replace p_{k+1}
18: $S_{k+1} \leftarrow \mathcal{U}(S_k, \mathcal{M}_{k+1}, g_0)$: apply the edits of the review, then revise and refresh if triggered ▷ LM, tools
19: If the route has to change, the director replans with the trajectory, fix, and prompt steps ▷ LM, tools
20: $k \leftarrow k + 1$
21: **until** a refresh marks the journey as finished

A.2 World State

Table 3 lists the fields of the world state.

Coordinate frame. All positions are metric and lie in one frame. It is the frame of I_0, leveled to gravity by the ground reading at initialization, and $\mathbf{T}_0$ denotes the pose of I_0 in this frame. The first camera lies at $x = z = 0$. The x axis points to the right of the first camera, the z axis points forward, and both are horizontal. The height h is measured upward from the standing surface under the first camera. The camera poses c_k that AgiWorld-WM receives and the geometry of $\mathcal{M}_k$ are expressed in the same frame.

Table 3: Fields of the world state.

Field	Meaning
Layout element	
Name	What the element is, with a fixed feature of the place when several are alike, e.g., "offering table on the passage side". Words that depend on the camera, such as *left* or *ahead*, are not allowed.
Kind	Surface, terrain, structure, object, opening, or zone. Surfaces and terrain form the ground, and structures and objects are solids.
Shape	Polygon, polyline, path, box, circle, or point, with metric coordinates in the (x, z) plane and an optional elevation per vertex for ramps and stairs.
Elevation, height	Height of the base above $h = 0$, and vertical extent.
Appearance	One line that describes what the element looks like.
Label	Confirmed (generated and observed) or imagined (designed by the agent but not yet reached by the camera).
Flags	Walkable, named in the instruction, read from I_0.
Retired element	
Record	Name, kind, position, and when and why it was retired.
Journey	
Summary	A few sentences on what the place is and how the journey proceeds.
Style	A medium, such as photorealistic real-world footage, and a look that names the light, the shadows, and the palette. Fixed for the whole journey.
Stages	Ordered list. Each stage has a goal, what is on screen and from where, a soft time window, the elements it uses, and a status (pending, active, done, or skipped).
Finished	Set by a refresh when the journey is complete. It ends the loop.

Layout. The layout is a set of named elements. The initial reconstruction and each revision write the confirmed elements, and the review may add, confirm, revise, or drop elements in small edits. The initial design and each refresh write the imagined elements. An imagined element becomes confirmed when the review marks it as generated as designed, or when a revision includes it with its observed geometry. An imagined element is retired when a revision reports that the video contradicts it, or when the confirmed layout grows over its place. Retired elements are kept as a record so that they are not proposed again.

Journey. The initial design writes the journey, and each refresh rewrites the stages. The review marks a stage as done once a chunk has shown its content. A stage states what the camera should show and contains no coordinates or speeds, because the route and the pace are decided by the director.

A.3 Geometric Grounding

Table 4 lists what the geometric tools compute and render, and which steps read each output. Figure 5 shows the renders of one iteration.

Sources of geometry. At initialization, the geometry estimator Φ estimates the depth of I_0. During the rollout, the tools read the depth $D_k = \Phi(v_k, c_k)$ stored in $\mathcal{M}_{k+1}$. They accumulate these depths and the depth of I_0 into one point cloud. They keep the points with a confidence of at least 0.35 within 25 m of their camera, and remove the sky.

Ground reading. A plane fitted to the whole image fails to find the ground when little of it is visible. The initial reconstruction therefore reads the ground first, before it writes the layout. To read the ground, the language model draws boxes on I_0 around the standing surface under the camera, other horizontal surfaces, and vertical faces such as walls or columns. The tools fit planes to the depth inside these boxes. The up direction comes from the horizontal surfaces, or from the vertical faces when no horizontal surface gives a usable plane. The floor $h = 0$ comes from the standing surface under the camera. When no marked surface gives a usable plane, or when the standing surface slopes by more than 15° or the ground 1 to 6 m ahead lies less than 0.2 m below the eye, the tools fall back to a plane fitted to the whole image with RANSAC.

Visibility. To find what each chunk will show, the tools project every layout element along the solved path. Each face of an element is rasterized with its depth, so structures and objects hide what

Table 4: Outputs of the geometric tools and the steps that read them.

Tool output	Content	Read by
Depth map of I_0	The depth of I_0 as a color map with a metric scale.	Initial reconstruction
Point cloud and samples of I_0	The leveled point cloud of I_0 from above, and I_0 with a grid of points labeled with their (x, z, h).	Initial reconstruction
Layout check	For each element, whether its position and height agree with the depth and the point cloud of I_0.	Initial reconstruction
Layout projection	The layout drawn as wireframes into I_0 or into the last frame of a chunk.	Initial reconstruction, Revision, Review
Floor plan	The layout over the accumulated point cloud, with the executed path and the planned path.	All steps except Prompt
Trajectory check	Collisions, eye-height and support violations, and violations of the motion bounds on the solved path.	Fix
Visibility timeline	The elements in view along each planned chunk.	Prompt
View previews	The planned view at each chunk boundary, with the element wireframes.	Prompt
Chunk frames	Five evenly spaced frames of the latest chunk.	Revision, Refresh, Trajectory (replan), Review
Camera measurements	The camera measured on the generated geometry.	Review, Revision, Refresh, Trajectory (replan)

lies behind them. A declared opening cuts a hole in the wall that contains it, so an element behind a doorway is visible through the doorway.

Drawing conventions. All renders share the frame of the layout. Confirmed elements have solid outlines colored by kind, and imagined elements have dashed orange outlines. Floor plans, the top-down maps of the main text, show the point cloud from above, colored by height, on a metric grid, with the first camera at the origin, the executed path in gray, and the planned path in red. Projections into an image draw each element as a wireframe from its base to its top.

A.4 Director: Trajectory

Waypoints. The trajectory step writes sparse waypoints on the floor plan. Each waypoint has a time, a position (x, z), a camera height h, a look-at target, the stage it serves, and a one-line purpose. The look-at target is a layout element, a point, or a heading with an optional pitch.

From waypoints to c_k. The tools turn each waypoint into a look-at pose with zero roll. They interpolate the poses, starting from the current pose, into a C^1 curve that starts with the velocity of the executed path, so a new path continues the executed one without a jump. The curve is sampled at 16 fps, and we call it the path. Each chunk takes 80 consecutive poses of the path, which cover 5 s and form its trajectory $c_k = \{\mathbf{T}_{k,f}\}_{f=1}^{n}$ with $n = 80$. Each request to AgiWorld-WM contains these metric poses, the camera intrinsics that Φ estimated for I_0, which stay fixed for the whole rollout, and the prompt p_k.

Planning window. The director designs the path for the next 60 s. When less than 30 s of the path remain, a refresh extends the stages, and the director replans the path for the next 60 s. The director replans from the current pose whenever the route has to change.

World-model specification. The initial design, the refresh, and the trajectory step share one specification of AgiWorld-WM in their guidelines. It states what the model does well and what it cannot do, together with the camera and prompt practices that follow from each (Table 5). We wrote it from the behavior of the model observed in generated videos. The specification also asks for a walking pace of about 0.6 to 0.9 m/s and for an eye height above the walked surface equal to that of I_0. The view tilts up or down only in a chunk that shows something above or below that the instruction names.

Checks. Each time the trajectory step writes waypoints, the tools solve them and check every chunk of the path against the bounds of Table 6. These bounds are looser than the practices of the specification. The estimated depth often places a surface across a doorway, so the point cloud can close an opening that the camera can pass. The collision check therefore ignores the point cloud

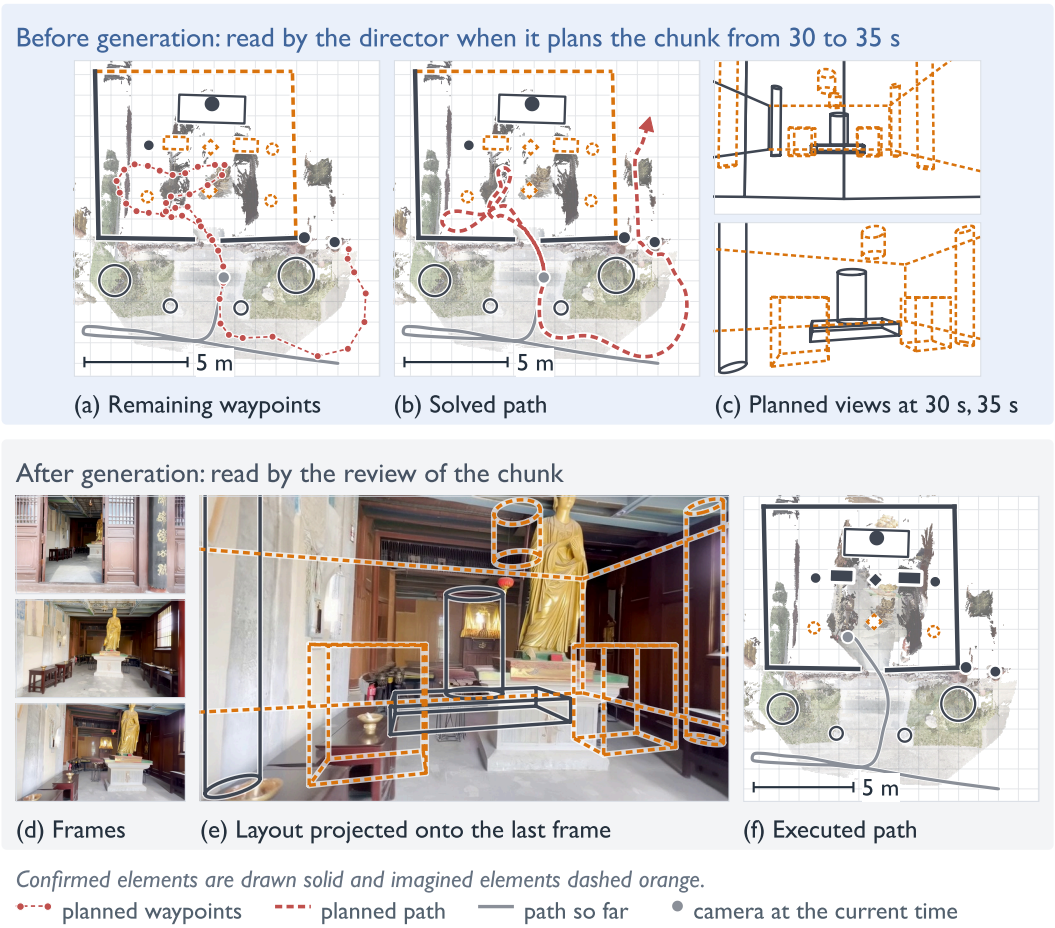

Figure 5: Renders prepared by the geometric tools in one iteration, redrawn from the intermediate results of the temple example in Figure 1, with the point cloud in its image colors and all confirmed elements in one color. The iteration generates the chunk from 30 to 35 s, in which the camera walks into the temple hall, and its path comes from a replan after the previous chunk. Top, before the chunk is generated: (a) the floor plan of the layout with the remaining planned waypoints, read by the trajectory step, (b) the solved path, read by the fix step, and (c) the planned views at 30 s and 35 s, read by the prompt step. Bottom, after the chunk is generated: (d) frames of the chunk, (e) the layout projected onto its last frame, and (f) the floor plan of the accumulated point cloud with the executed path, all read by the review. Confirmed elements are drawn solid and imagined elements dashed.

inside declared openings. A solved path that violates a bound returns to the director in the fix step. After at most three rounds of correction, the last path is accepted even if a bound is still violated.

A.5 Director: Prompt

Visibility along the path. The prompt step derives the prompt p_k from the solved path. After the path is accepted, it writes the prompts of all chunks in the planning window at once. For each chunk, the tools give it four lists: the elements visible in the first frame, the elements that enter the view or grow, the elements that leave the view, and the elements visible in the last frame. Each list is ranked by the largest share of the frame that an element covers during the chunk. Each imagined element is also marked with whether it may be named under the rules below. The prompt step also receives a preview of the planned view at each chunk boundary.

Format. The prompt uses the structured format of the training captions (Appendix C): a style line followed by the start, the middle, and the end of the chunk. The director writes the three sections, and the tools put the style line of the journey in front of them. The start describes the first frame, and the end describes the last frame. The middle describes what becomes visible, grows, or leaves

Table 5: The specification of AgiWorld-WM shared by the initial design, the refresh, and the trajectory step. Each practice follows from the behavior on its left.

Behavior of AgiWorld-WM	Camera and prompt practice
It extends what is already in the middle of the frame while the camera moves smoothly and most of what the previous seconds showed stays in view.	The camera turns gently while it walks, so that most of what the last five seconds showed stays in view. It walks in the direction it looks, and walks backward only for a few seconds.
A space that it has never shown is hard to generate at the moment the camera enters it. Walking through a doorway or around a corner onto unseen ground can produce a jump.	A new space is kept in the middle of the frame from the first chunk in which it is visible, and the camera walks toward it without stopping. The prompt of each of these chunks states what is there. The chunk that crosses an opening ends well beyond it.
Its memory holds only what it has generated. Parts of a room that were never in view have nothing to stay consistent with.	In a room, the first chunk turns the view across the room, about a quarter turn while walking slowly, so that its walls are in the memory before the camera leaves them. To look back, the camera turns around while walking.
It keeps a large space most consistent when the space is shown whole from a distance. Narrow views that each cover a small patch of a room give it little to keep consistent.	Large scenes are shown from far. An element that the instruction names is held in the middle of the frame, from far enough back that the space around it is also in view.
A wall or a shelf that fills the frame from close range gives it nothing to continue.	The camera looks where the space ahead opens, down a lane or across a room, rather than at the wall it passes. It moves sideways only while the view stays open.
It generates what the prompt names where the frame gives it room, and what it generates can differ from the layout.	The prompt names only what the frames of the chunk will certainly show. The next path is designed on the generated geometry.

Table 6: The main bounds on the solved path. A violation is returned to the director.

Check	Bound
Collision	A sphere of radius 0.25 m around the camera stays clear of the confirmed solids and of the point cloud.
Eye height	The eye stays within 0.7 to 1.4 times its starting height above the walked surface of the layout.
Support	The camera does not pass over an area that the layout marks as not walkable.
Speed	At most 2.5 m/s, with a change of pace of at most 0.6 m/s^2.
Turning	At most 90°/s, and at most 25°/s on average over a chunk.
Chunk boundary	Between consecutive planned chunks, the velocity changes by at most 1 m/s and the turn rate by at most 45°/s.
Travel	At least 1.5 m in every chunk.

the view over the five seconds, and what stays in view. The director aims at 27, 73, and 38 words for the three sections, which are the median lengths of the training captions.

Naming rules. AgiWorld-WM generates what the prompt names wherever the frame gives it room. An imagined element may therefore be named only in a chunk that sees it through a confirmed opening, or that holds it in the middle third of the frame for at least 2 s within 10 m. Every imagined element that the instruction names and that enters the view under this rule must be named in the middle section of the prompt. A confirmed element that covers less than 2% of the frame is not described unless the instruction names it or an earlier chunk showed it clearly, because a description makes the model generate it larger than it is. AgiWorld-WM also generates what the prompt negates. The prompt therefore contains no negations, except "no longer visible" and its variants, which the training captions use for an element that leaves the view. The scene is described as static, and camera words are not allowed. An element named in the instruction is described as the instruction describes it, even after a chunk has generated it differently.

Validation. The tools check the word ranges of the three sections (18 to 40, 50 to 115, and 22 to 60 words), negations, camera words, and the naming rules, also for a prompt that the review rewrites.

A.6 DIRECTOR: REVIEW AND REGENERATION

AgiWorld-WM follows the planned camera poses, but the world that it generates around them can depart from the design. A camera that keeps its planned height above the layout can still sink toward the generated ground, and a path that clears a designed wall can still pass close to the wall that was actually generated. The tools therefore measure the camera on the geometry that AgiWorld-WM estimated for the generated frames, and pass the measurements to the review.

Measurements. The tools measure on the point cloud accumulated from I_0 and all chunks generated so far. They evaluate eight poses of the chunk and report the median of each quantity over these poses.

1. The ground plane is fitted to the points within 8 m of the camera, measured horizontally, that lie at most 0.35 m above the 25th percentile of their heights.
2. The eye height is the height of the camera above this plane.
3. The ground beside the path is measured on the points 1 to 6 m ahead that lie within 0.3 m of the plane. Its width to the left and to the right of the line of travel is given by the 2nd and 98th percentiles of their lateral offsets.
4. The nearest solid is the 5th percentile of the horizontal distance of the points in a cone of $\pm 30^\circ$ around the view direction, between 0.3 and 2.5 m above the plane.
5. The tilt is the angle between the view direction and the ground plane.

Inputs and outputs. The director reviews the chunk from one composite image that shows I_0, five frames of the chunk, the layout projected onto its last frame, and a floor plan of the accumulated point cloud with the executed path and the next planned segment. It also reads the stages, the prompt of the chunk, the elements named in the instruction, the plan of the next chunk, the measurements, and the targets, which are the imagined elements that the next stages use. For each element named in the prompt, the review reports whether it was generated as described, is absent, or is different, meaning that something else stands in its place. For each target, it reports whether the target is established, not yet visible, or contradicted, for example by a wall or a closed door in its place. The review rates the overall content match as ok, minor, or major. A major mismatch means that the layout around the camera is wrong, or that the camera entered an imagined space that looks different from its design. The review also returns small layout edits (confirm, revise, drop, or add), stage updates, and a decision for the next chunk: keep it, rewrite its prompt, or ask for a new route.

Regeneration. A chunk fails the review, and is generated again, when one of the following checks fires. The eye-height checks compare with a reference eye height measured on the first chunk.

- The eye height is below 0.55 times or above 1.4 times the reference.
- The nearest solid ahead is closer than 0.7 m.
- Less than 0.25 m of generated ground lies on one side of the path.
- The view points more than 10° below the ground plane.
- The review reports an element named in the instruction as different or contradicted.

We set the thresholds on chunks of earlier rollouts, so that no chunk accepted on inspection crosses them. The overall content match alone does not trigger a regeneration, and the other disagreements are resolved by the state update.

When a check fires, the state is restored to its value before the review. AgiWorld-WM restores its memory to $\mathcal{M}_k$ and discards the frames, latents, and geometry of v_k. The director replans from the end of the previous chunk, given the reasons and the first review, and AgiWorld-WM generates the chunk again with a new seed. The new chunk is measured and reviewed like any other.

A new sample often clears a fault that occurred by chance, such as a view that sinks toward the ground, but it rarely changes how the model generates a place. We therefore bound regeneration. The first chunk is never regenerated, because it defines the reference eye height. Each chunk is regenerated at most once, a rollout regenerates at most four chunks, and each element named in the instruction triggers at most one regeneration. A fault that persists after a regeneration no longer triggers a regeneration for the rest of the rollout and is only reported.

A.7 State Update

After every chunk, the state update first applies the small layout edits and stage updates of the review. The revision and the refresh then run only when their triggers fire, and the director replans afterwards when the route has to change.

Revision. The confirmed layout is revised when the review reports a major mismatch or a contradicted target. It is also revised when the generated ground and the surface of the layout differ by more than 0.3 m under the camera for two consecutive chunks. The revision returns the complete confirmed layout. It keeps the elements that still hold, corrects those that the frames contradict, and adds what the frames newly show. Imagined elements that were generated are promoted to confirmed with their observed geometry. Imagined elements that the video contradicted are listed and retired.

Refresh. A refresh runs when a stage changes status, when less than 30 s of the path remain, when the confirmed layout was revised, or when imagined elements were retired. A refresh returns the stages with their status, new stages where the journey runs short, and edits of the imagined layout (add, revise, or drop). The refreshed stages have to cover the next 60 s. A stage is skipped only when the video contradicted its space. The imagined layout of the initial design and of every refresh has to pass four consistency checks against the confirmed layout, and a failed check is returned with its errors.

1. Imagined ground connects to confirmed ground, directly, through an opening, or through other imagined ground.
2. Imagined ground, openings, and zones do not cross a confirmed wall outside a confirmed opening, and do not lie inside a confirmed solid.
3. Imagined objects do not overlap a confirmed solid at the same height.
4. Imagined ground does not cover confirmed ground at the same level.

Retirement. After the edits of the review and after a revision, the confirmed layout can grow over imagined elements. The tools then retire every imagined element that fails the second or the fourth check, and the retirement triggers a refresh.

Replan. The director replans the route from the current pose when the review asks for a new route, when the layout was revised, when a refresh changed the stages that the route follows or the elements they use, when the remaining path collides with the updated layout, or when less than 30 s of the path remain. A replan runs the trajectory, fix, and prompt steps again for the next 60 s.

Instruction and input image. The generated video overrides the design of the agent but not the instruction. Elements read from I_0 are not revised or dropped to match a generated chunk. A confirmed element named in the instruction is not dropped or changed in kind. An imagined element named in the instruction is not dropped before a fair attempt. A fair attempt is a chunk in which the camera held the element in the middle third of the frame for at least 2 s within 10 m, the middle section of the prompt named it, and the review reported it as absent or different. This bound keeps the journey from stopping at an element that AgiWorld-WM does not generate. When the video shows something else in place of a named element, the chunk is regenerated, and until a fair attempt has been made, the director keeps approaching the element and describing it as the instruction does.

Termination. The loop ends when a refresh marks the journey as finished, which it does once the instruction has been carried out.

B AgiWorld-WM Details

History context and memory. The backbone encodes a video with a causal VAE that maps the first frame to one latent frame and every following group of four frames to one latent frame at the resolution of 832×480. A chunk v_k has $n = 80$ frames, five seconds at 16 fps. It is encoded jointly with its temporal history, the last 17 frames of v_{k-1}, into 25 latent frames, of which the first 5 hold the temporal history and the other 20 hold the chunk. Each of the five spatial-history frames is encoded separately into a single latent frame. The sequence therefore consists of 5 spatial, 5 temporal, and 20 generated latent frames. The history latents are clean, with diffusion timestep

0, and are marked as observed in the image-conditioning mask of the backbone, whereas all 20 generated latent frames are denoised from noise, with no observed frame. A spatial slot that the selection leaves empty is filled with zeros and masked. For the first chunk, the input image I_0 repeated 17 times forms a static temporal history, and all spatial slots are empty. At inference, the previous chunk is decoded, and its last 17 frames are re-encoded as the next temporal history.

The memory stores every fourth generated frame, one per latent frame, with its camera and depth. At inference, the geometry estimator Φ is Depth Anything 3 (Lin et al., 2025) conditioned on the requested cameras, so the memory keeps the poses of c_k, and only the depth, its confidence, and a sky mask are estimated.

Occlusion-aware history selection. Each memory point $\mathbf{x}$ carries a unit normal $\mathbf{n}_\mathbf{x}$ estimated from its depth map. In a target view r with camera center $\mathbf{o}_r$, it is front-facing when its normal points back toward the camera,

$$f_r(\mathbf{x}) = \big[\, \mathbf{n}_\mathbf{x} \cdot (\mathbf{o}_r - \mathbf{x}) > 0 \,\big]. \tag{1}$$

Rasterizing the front-facing points as small surfels, the quads spanned by four neighboring pixels of the depth map, fills the gaps between points and gives, at each pixel p, the depth of the nearest front-facing surface,

$$d_r(p) = \min \big\{\, z_r(\mathbf{x}) \;:\; f_r(\mathbf{x}) = 1,\; \pi_r(\mathbf{x}) = p \,\big\}. \tag{2}$$

A point is unoccluded when its own depth reaches this surface rather than lying behind it, up to a tolerance δ that keeps a surface from occluding itself,

$$g_r(\mathbf{x}) = \big[\, z_r(\mathbf{x}) \le (1+\delta)\, d_r(\pi_r(\mathbf{x})) \,\big]. \tag{3}$$

A memory point is visible from view r when it is both front-facing and unoccluded,

$$\mathrm{vis}_r(\mathbf{x}) = f_r(\mathbf{x})\, g_r(\mathbf{x}). \tag{4}$$

Here $[\cdot]$ equals 1 when the enclosed condition holds and 0 otherwise, π_r and z_r are the projection and the depth in view r, and a point counts as visible along c_k if it is visible from any of six target views, spaced evenly along c_k and ending at its last frame.

Given this test, we build the spatial history by a greedy set-cover over the target pixels of c_k. The anchor, the last frame of v_{k-1}, is already in the sequence as part of the temporal history, so we start from the target pixels its visible points cover and repeatedly add the memory frame whose visible points cover the most target pixels not yet covered, until five frames are selected. Recent frames are excluded because the temporal history already carries them, so the selected frames complement one another and the anchor instead of repeating one viewpoint. The same visibility test is applied when the correspondence is built, so retrieval and injection agree on what the camera can see. During training, the selection runs on the annotated geometry of the training video.

Geometry correspondence injection. We follow Lyra 2.0 (Shen et al., 2026) for geometry correspondence injection. The correspondence has six sources, the five spatial-history frames and the anchor. For each source, we build a map that stores at every pixel its normalized image coordinates in $[-1, 1]^2$ and a constant code in $[-1, 1]$ that identifies the source. This map is forward-warped with the depth of the source into the camera of each of the 20 generated latent frames at the full resolution of 832×480, together with the depth of each warped point in the target camera. The warped depth is normalized to $[-1, 1]$ jointly over all sources and frames. Source pixels at depth discontinuities, where the depth varies by more than 5% within a 5×5 window, are not warped, and target pixels that no source point reaches, or whose source point fails the visibility test, are set to zero.

Reliability-based memory maintenance. The maintenance step runs only at inference and makes two judgments, both built on the occlusion-aware visibility test of equation 4: a reliability judgment on each memory point, and a generation-time judgment that keeps earlier content when two sources conflict.

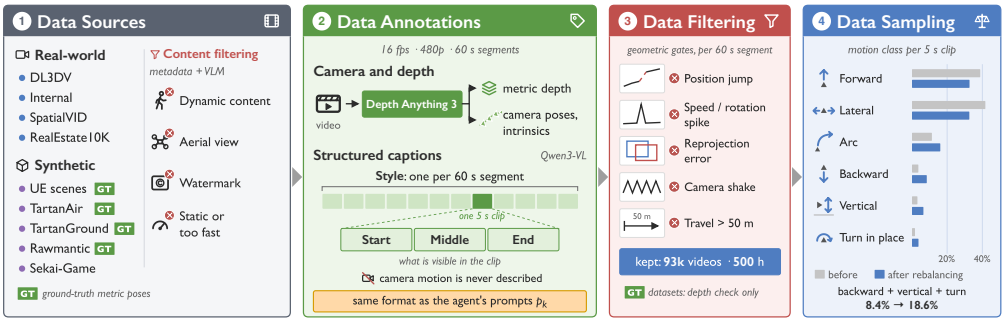

Figure 6: Data pipeline of AgiWorld-WM. (1) Data sources: four real-world and five synthetic datasets. Real-world videos are filtered with metadata and a VLM for dynamic content, aerial views, and near-static or overly fast camera motion. GT marks the datasets with ground-truth metric poses. (2) Data annotations: Depth Anything 3 (Lin et al., 2025) estimates metric depth for every segment, together with camera poses and intrinsics for the datasets without ground-truth poses, and Qwen3-VL (Bai et al., 2025a) writes a structured caption for every 5-second clip, a style line for the segment followed by what is visible at the start, in the middle, and at the end of the clip. The captions never describe the camera motion and have the same format as the prompts p_k of the director. (3) Data filtering: geometric gates on every segment discard unreliable pose and depth annotations. Datasets with ground-truth poses receive only a depth check. (4) Data sampling: every clip is labeled with a camera motion class, and rare classes are sampled more often. The bars show the share of each class before and after rebalancing.

The reliability judgment decides whether a memory point can be trusted. Let $\mathbf{o}$ be the center of the camera that observed a memory point $\mathbf{x}$. The point is reliable when three conditions hold: (i) its confidence from the geometry estimator is at least the 20th percentile of the confidence over the memory; (ii) it was observed close to head-on, $\mathbf{n}_\mathbf{x} \cdot (\mathbf{o} - \mathbf{x}) > \|\mathbf{o} - \mathbf{x}\| \cos\theta_{\max}$, i.e. its incidence angle is below $\theta_{\max}$ (the front-facing test of equation 1 with its threshold raised from 0); (iii) it was observed from nearby, $\|\mathbf{o} - \mathbf{x}\| < z_{\max}$. Reliable points are used first for retrieval and correspondence, and the remaining front-facing points fill only the holes they leave. An unreliable point is deprioritized but never discarded. It therefore still contributes to the occluding surface d_r of equation 2, and an untrusted surface still occludes what lies behind it.

The generation-time judgment keeps the earlier-generated source when two conflict. A later point $\mathbf{x}$ conflicts with an earlier frame u when it falls where that frame saw empty space, i.e. when it lies clearly in front of the surface u observed, $z_u(\mathbf{x}) < (1 - \delta)\, d_u(\pi_u(\mathbf{x}))$, using the same tolerance δ as the occlusion test of equation 3. This free-space test mirrors the occlusion test of equation 3: occlusion asks whether a point is hidden behind the nearest surface, free space whether it floats clearly in front of one. When $\mathbf{x}$ conflicts with an earlier frame that observed its location reliably, we drop $\mathbf{x}$ from the memory, removing it from retrieval, the correspondence, and the occluding surface. A conflicting point therefore occludes nothing, whereas a merely unreliable point is kept and still occludes.

In our implementation, we set $\theta_{\max} = 70°$, $z_{\max} = 30$ m, and $\delta = 0.05$.

C Training Data and Training Strategy

C.1 Training Data

AgiWorld-WM needs to follow metric camera trajectories accurately, keep a memory of everything it has generated, and faithfully generate the scene that the structured prompt describes. To support this, the training data needs metric camera poses with diverse camera motion, long videos with per-frame depth that return to earlier viewpoints, and scene diversity. We build our data pipeline (Figure 6) for training data that meet these requirements.

Data sources. We combine nine datasets. DL3DV (Ling et al., 2024) and an internal collection of indoor and outdoor scenes provide long-term real-world captures in which the camera moves around or explores a scene. SpatialVID (Wang et al., 2026a) and RealEstate10K (Zhou et al., 2018) provide real-world walking, driving, and indoor walkthrough videos that mostly move forward and contribute diverse in-the-wild scenes. The remaining five datasets are synthetic. Videos rendered from

Unreal Engine scenes have the most balanced camera motion, including turns and vertical movement. TartanAir (Wang et al., 2020) and TartanGround (Patel et al., 2025) provide fast drone flights and ground traversal through large simulated environments. Rawmantic (Team, 2026) provides rendered interiors, and Sekai-Game (Li et al., 2026) provides walking videos recorded in a game engine. All synthetic datasets except Sekai-Game come with ground-truth metric camera poses.

Data annotations. All videos are resampled to 16 fps at 480p resolution and split into segments of at most 60 seconds, and every segment receives camera and depth annotations and captions. Depth Anything 3 (Lin et al., 2025) is used to jointly estimate camera poses, intrinsics, and metric depth for each segment. For the synthetic datasets with ground-truth poses, we keep the poses and run the same model conditioned on them. Every 5-second clip is captioned by Qwen3-VL (Bai et al., 2025a) in a structured format: a style description shared by the whole segment, followed by descriptions of what is visible at the start, in the middle and at the end of the clip. The captions describe the scene but never the camera motion, which enters the model only through the camera condition. The structure makes the instruction to the model explicit, so the generated video follows the caption more faithfully. It also gives the director a fixed format to write its prompts in (Section 3.3), which makes generation easier to control.

Data filtering. We filter the data in two stages for high-quality source videos and reliable geometry annotations. We first remove footage that is unsuitable for training: clips with moving people, vehicles or flowing water, aerial views, near-static or overly fast camera motion, and large watermarks. These are detected from dataset metadata and by a local VLM (Bai et al., 2025a).

After annotating the camera parameters and depth, we apply geometric gates to every segment and discard those with unreliable geometry annotations, since a single erroneous pose corrupts the coordinate frame of the entire segment. The gates evaluate the estimated trajectory and depth for (i) sudden jumps of the camera position, (ii) sudden changes in speed or rotation, (iii) low depth reprojection consistency between nearby frames, and (iv) shaking of the camera. Their thresholds are calibrated separately for each dataset, and datasets with ground-truth poses pass only a depth quality check. We also discard training samples whose camera moves more than 50 meters, because the relative camera transforms in PRoPE (Li et al., 2025c) grow with translation and such samples produce exploding gradients. After filtering, about 93k source videos and 500 hours of video remain.

Data sampling. Accurate camera control requires following arbitrary camera trajectories, in any direction and with any motion pattern, so the training data has to cover camera motion of all kinds. In the collected videos, however, most clips move straight forward or sideways, and turning, backward, and vertical motion are rare. We therefore label every 5-second clip with a camera motion class computed from its trajectory, namely forward, backward, lateral, vertical, turning in place, or moving along an arc. During training, we sample clips of the rare motion classes more often, so that the different camera motions are more evenly represented.

C.2 Staged Training

We build our model on Wan2.2-I2V-A14B (Wan et al., 2025), initialized from the LingBot-World checkpoint (Team et al., 2026). The model is trained in two stages, each building on the previous.

Stage 1: Metric camera adapter. We fine-tune only the camera adapter, freezing the rest of the backbone, so the model follows metric camera trajectories. Camera-conditioned models are usually trained with per-clip-normalized poses, which discard the absolute scale of the motion. We instead condition on the metric relative poses between consecutive frames and train the adapter to interpret their absolute scale. We train bidirectionally on single-chunk clips with the base model's flow-matching objective, isolating accurate camera following before any memory is introduced.

Stage 2: Memory-conditioned autoregressive training. The second stage trains the model to generate autoregressively. As at inference, the spatial and temporal history is placed in front of the target chunk as clean context, and the loss is applied only to the generated frames. We add the relative camera positional encoding and the correspondence-injection encoder, both zero-initialized, and train them together with the self-attention, while the camera adapter from Stage 1 and the rest of the backbone stay frozen. The occlusion-aware history selection and the warped correspondences are parameter-free geometry, computed from the annotated geometry at training and from the estimated geometry of the generated frames at inference, so only the modules that inject them into attention are

learned. Since the history at inference comes from the model's own drifting output, we add random noise and saturation to the clean context during training to keep the model robust to that drift.

D Additional Experiments

D.1 Implementation Details

We train AgiWorld-WM on 64 A100 GPUs with a global batch size of 32, using AdamW with a learning rate of 2×10^{-5}. Stage 1 trains the camera adapter for 5,000 steps, and Stage 2 trains for 7,000 steps.

D.2 Metrics and Baselines

Metrics. We evaluate visual quality and text following by FID (Heusel et al., 2017) and by the imaging quality, aesthetic quality, and text consistency scores of VBench (Huang et al., 2024). We evaluate camera following and geometric consistency by estimating camera poses and depth from the generated video with Depth Anything 3 (Lin et al., 2025), and report the mean translation and rotation errors of the estimated camera trajectory against the target trajectory (Trans., Rot.) and the depth reprojection error between frames one second apart (Repr.).

Baselines. We compare with recent state-of-the-art camera-controlled long video generation models: SPMem (Wu et al., 2025a), GEN3C (Ren et al., 2025), HY-WorldPlay (Sun et al., 2025), LingBot-World (Team et al., 2026), Matrix-Game 3.0 (Wang et al., 2026b), Lyra 2.0 (Shen et al., 2026), and SANA-WM (Zhu et al., 2026). For a fair comparison, SPMem (Wu et al., 2025a), GEN3C (Ren et al., 2025), and Lyra 2.0 (Shen et al., 2026), which rely on estimated geometry, use the same Depth Anything 3 (Lin et al., 2025) as our model. All models generate the 60-second video autoregressively with their full denoising schedules, except SANA-WM (Zhu et al., 2026), which generates the whole video bidirectionally.

D.3 World Explore benchmark

World Explore evaluates world exploration, where the camera keeps moving into parts of the scene that the input image does not show. Its 40 scenes are all videos of a hold-out split of the outdoor part of the internal collection (Appendix C.1), and this split was drawn at random from the videos of that part. They are never used in training. We chose the outdoor hold-out over the indoor one because its camera trajectories move away from the start instead of circling the scene, and did not select individual scenes by their camera motion. The videos are mostly handheld or gimbal walks along paths, streets, lakeshores, playgrounds and building exteriors, several of them in snow (Figure 7, left).

The trajectories of the two benchmarks have similar lengths, with a median of 28.9 m on DL3DV-Evaluation and 28.4 m on World Explore, but they cover the scene differently. On DL3DV-Evaluation, the camera stays within a median of 10.0 m of its start and returns to earlier viewpoints. For World Explore, the camera moves up to a median of 18.1 m away from its start. The right panel of the figure shows the distance from the start along the trajectories.

D.4 Ablation of the Memory Components

Table 7: Ablation on DL3DV-Evaluation. Metrics are computed over the full 60 s sequence and over only the last 30, 20 and 10 s, so the later columns measure long-term consistency. Trans., Rot. and Repr. are translation (m), rotation (°) and depth reprojection errors. Occ.: occlusion-aware history selection. Rel.: reliability-based memory maintenance. Best in **bold**.

		Full				Last 30 s				Last 20 s				Last 10 s			
Occ.	Rel.	FID↓	Trans.↓	Rot.↓	Repr.↓	FID↓	Trans.↓	Rot.↓	Repr.↓	FID↓	Trans.↓	Rot.↓	Repr.↓	FID↓	Trans.↓	Rot.↓	Repr.↓
		22.32	0.35	1.18	0.023	34.94	0.36	1.45	0.031	41.91	0.40	1.63	0.040	57.74	0.46	2.18	0.049
✓		22.66	0.39	1.06	0.022	35.41	0.42	1.22	0.028	41.70	0.47	1.23	0.033	56.94	0.56	1.40	0.040
✓	✓	**21.74**	**0.29**	**0.99**	**0.020**	**32.21**	**0.30**	**1.11**	**0.024**	**37.42**	**0.33**	**1.18**	**0.027**	**51.84**	**0.37**	**1.38**	**0.031**



Figure 7: World Explore benchmark. Left: the first frames of 20 of the 40 scenes. Right: distance of the camera from its starting position along the target trajectories of DL3DV-Evaluation and World Explore. DL3DV-Evaluation trajectories move away and come back to a median distance of 1.6 m from the start, whereas World Explore trajectories keep moving away and end at a median distance of 18.1 m.



Figure 8: Qualitative ablation of the two memory components. Each row shows one scene: the input image, the frames generated at one later time step by the variant without both components (w/o both), the variant without reliability-based memory maintenance (w/o Rel.) and the full model (Full), and the ground-truth frame at the same camera pose (GT).

Table 7 reports FID and the camera and geometry errors over the full 60 seconds and over the last 30, 20, and 10 seconds, since both components address occlusion and error accumulation over long rollouts, whose effect concentrates in the later part of the sequence. The full model is the best on all four metrics in every window. Removing both components raises FID from 21.74 to 22.32 over the full sequence but from 51.84 to 57.74 over the last 10 seconds. The camera and geometry errors show the same trend. Over the last 10 seconds, removing both components raises the translation, rotation, and reprojection errors from 0.37 m, 1.38°, and 0.031 to 0.46 m, 2.18°, and 0.049. Reliability-based memory maintenance accounts for most of the FID gap (56.94 with occlusion-aware history selection alone), and the occlusion-aware history selection on its own lowers FID only over the last 20 and 10 seconds, since occlusion reasoning depends on the estimated geometry of the history, including its unreliable parts when reliability-based memory maintenance is off. With occlusion-aware history selection alone, the rotation and reprojection errors already drop in every window, but the translation error rises, and adding reliability-based memory maintenance brings it from 0.56 down to 0.37 m over the last 10 seconds. (See Figure 8 in Appendix D.5 for qualitative ablation comparisons.)

D.5 Qualitative Comparison

Figure 11 compares six of the seven baselines of Table 1 on two further scenes, at two time steps each. Figure 8 shows the ablation variants of the two memory components. Figure 9 shows the ablation of AgiWorld-Agent, and Figure 10 shows AgiWorld-Agent directing different world models. Full videos are provided in the supplementary material.

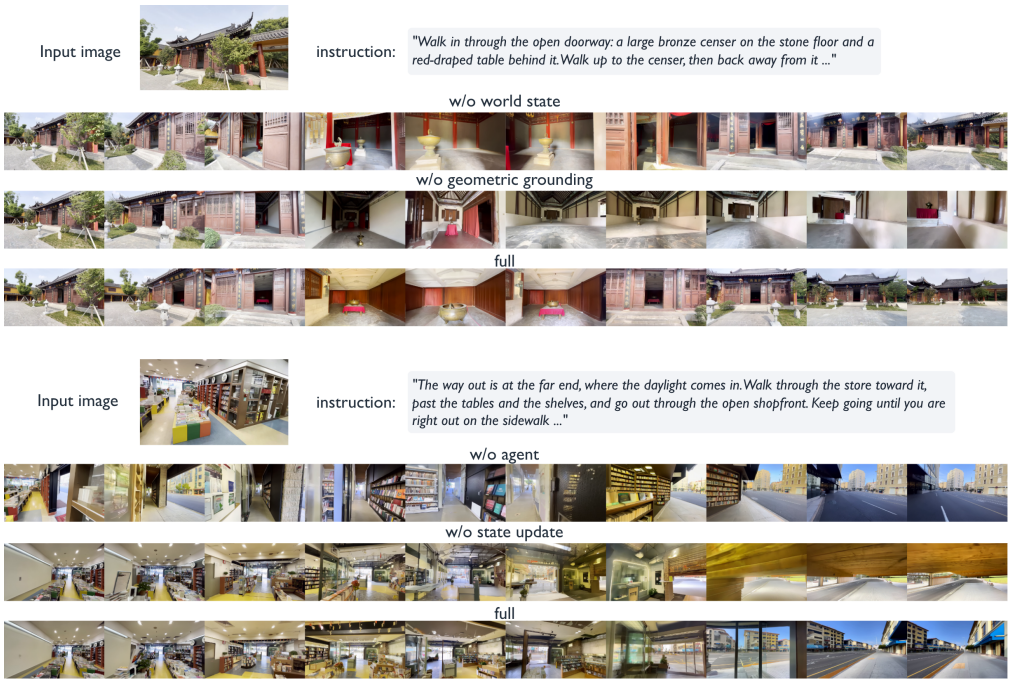

Figure 9: Qualitative ablation of AgiWorld-Agent on two examples, each shown with its input image and abbreviated instruction. Each row shows frames sampled in order along the video of one variant.



Figure 10: Replacing the world model of AgiWorld. All rows use the same AgiWorld-Agent, input image, instruction and initial world state, and differ only in the world model: LingBot-World, Matrix-Game 3.0, Lyra 2.0 and AgiWorld-WM from top to bottom.

E Limitations and Future Directions

AgiWorld currently builds static worlds, and its world state describes places rather than events. Extending the world state to moving objects and events, so that the agent directs what happens in the world as well as where the camera goes, is a natural next step. We use camera trajectories and text prompts as the interface between the agent and the world model, which keeps the agent compatible



Figure 11: Qualitative comparison of camera-controlled long video generation.

with camera-controlled video models in general. A text prompt, however, is a coarser condition than a pixel-level one, and the world model does not always generate everything the prompt names. A finer-grained condition, such as a pixel-level semantic layout of each frame, is a promising direction. AgiWorld-Agent already supports it, since its geometric tools project the world state into every planned view, but the world model would need to be retrained to take such a condition.