CAMCANVAS: ABSOLUTE CAMERA PLACEMENT IN VIDEO WORLD MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Interactive video world models generate long sequences under camera commands and sustain extended exploration of a generated world. Their camera interface comes from relative pose encodings: a command specifies motion with respect to the current context window. As a result, when the camera is commanded back to a place visited long ago, these models fail to return to the commanded viewpoint. We present CamCanvas, which stores the model's own generated content in a panoramic canvas, retaining the first observation of each location. When the camera returns, the model is conditioned on the rendering of this canvas at the commanded pose. The model thus receives the target view directly, so it has neither to infer its absolute pose from relative motion nor to recover that view from a limited context window. The return test assesses absolute camera placement by commanding the camera away and back and estimating, from the frames, the viewing-angle difference between the return and the model's own first visit, with no external alignment. Return success rises from 0% to 91% on 90° returns and from 17% to 99% on real video. CamCanvas also generalizes to a video world model with a different architecture.

1 INTRODUCTION

Video world models are becoming interactive simulators for users and agents (Ha & Schmidhuber, 2018; Alonso et al., 2024; Valevski et al., 2025; Bruce et al., 2024; Sun et al., 2025; He et al., 2025c; Gao et al., 2026). Each camera command is a query: "put me at this pose and show me what is there." For policy training, robotics, and games, the camera must reach the commanded pose and show the content already generated there, even after the camera has explored elsewhere. Current models already follow camera commands smoothly and produce plausible long explorations.

Despite this progress, when the camera is commanded back to a place it visited earlier, almost every existing model fails. Figure 1 shows a 110° return with 77% of the target view outside the model's attention window, the finite set of recent frames it can see. We observe that Wan2.2-5B ends 54° from the view the model generated when the camera was first at that pose. HY-WorldPlay, which has retrieval-based memory, produces a plausible view at the wrong pose (Figure 5) and succeeds on 0% of these returns. The missing capability is *absolute camera placement*: the commanded pose, not the path taken to it, determines which part of the generated world is on screen.

We trace this failure to how camera commands are encoded: relative commands specify how to move from the current view (Li et al., 2025a; Xiang et al., 2026; Xu et al., 2026b), but the model sees only a finite window of recent frames. In Theorem 1, we show that a generator conditioned only on window frames and relative commands cannot recover an earlier visit beyond what the window retains (Section 3.2), so more training cannot restore the missing information. While retrieval-based memory brings back views taken at other poses (Xiao et al., 2025; Li et al., 2025b; Wu et al., 2025; Yu et al., 2025a), it still leaves the model to work out the requested view.

In this paper, we propose CamCanvas, which supplies the view that the model cannot recover from its recent frames (Figure 2). Specifically, we store generated content in a panoramic canvas, writing each location only the first time it is seen. When the camera returns, we render the canvas at the commanded pose and feed this view to the model as a noise-free conditioning input aligned with the frames to be generated. The rendered view and the frames to be generated share the same com-

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.

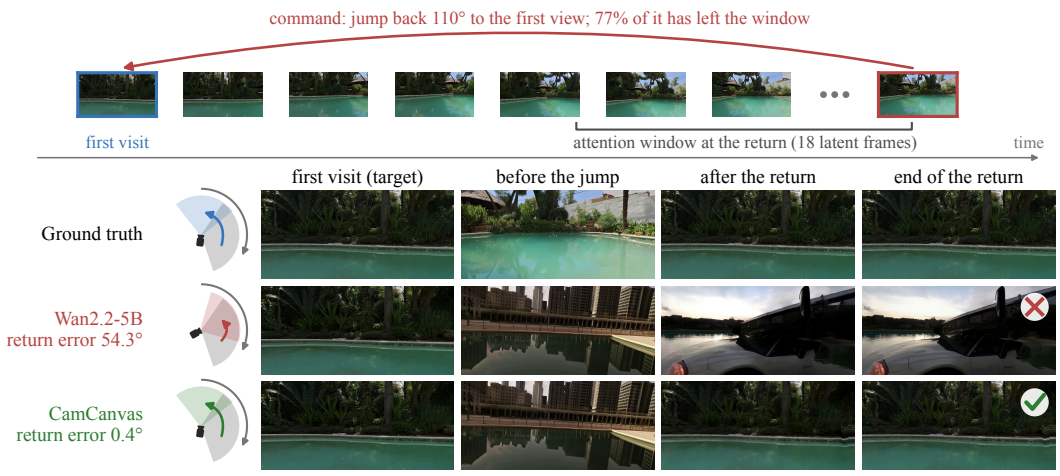

Figure 1: **A 110° return with 77% of the target view outside the attention window.** Top: the ground-truth camera sweep, with the first visit (blue) and the return (red). Columns show the first visit, the frame before the jump, and two frames after the return; each row's top-down schematic shows the attention window (grey) and the view shown in the final frame. Wan2.2-5B ends $54.3°$ from its first visit, while CamCanvas returns within $0.4°$.

manded pose, so their content is aligned in image coordinates. This enables the model to produce the target by following the condition instead of inferring the camera from relative motion (Section 3.4).

To measure this capability, we introduce the *return test*, which commands the camera away and back and scores the return against the model's own first-visit frame, without ground-truth alignment or camera calibration (Section 4.1). Extensive experiments show that CamCanvas raises $90°$ return success from 0% to 91% and lowers median error from $70°$ to $1.1°$; in Figure 1, it returns within $0.4°$. Real-video return success rises from 17% to 99%. Moreover, the same rendered condition remains effective on SANA-WM (Zhu et al., 2026), a model with a different architecture.

In summary, our contributions are as follows:

- We identify the missing capability of absolute camera placement and establish its cause: window frames and relative commands cannot recover earlier content beyond the information retained in the window (Theorem 1).
- We introduce CamCanvas, which substantially raises return success on panorama and real-video returns and works on a second architecture, as measured by the return test.

2 Related Work

Interactive video world models and camera control. Streaming video models generate successive chunks and respond to camera commands chosen during exploration (Chen et al., 2024; Yin et al., 2025; Huang et al., 2025c; Cui et al., 2026; Liu et al., 2026; Yang et al., 2026; He et al., 2025c; Sun et al., 2025; Gao et al., 2026; Huang et al., 2026). Camera conditioning represents motion through Plücker rays, relative projective encodings, and discrete actions (Wang et al., 2024; He et al., 2025a; Bahmani et al., 2025; He et al., 2025b; Li et al., 2025a; He et al., 2025c; Xiang et al., 2026; Xu et al., 2026b). These representations support camera control, but relative encodings specify movement from visible context without locating the camera among previously generated scene content (Li et al., 2025a; Huang et al., 2026). CamCanvas retains this camera conditioning and adds a rendered view that ties the commanded pose to content in the generated world.

Long-horizon consistency and memory. Interactive world models report preserving the appearance of places revisited during exploration (Wang et al., 2026a; Gao et al., 2026; Xu et al., 2026a; DreamX Team et al., 2026). Memory modules store past views and retrieve them by time or spatial location (Xiao et al., 2025; Yu et al., 2025a; Li et al., 2025b; Wu et al., 2025; Huang et al.,

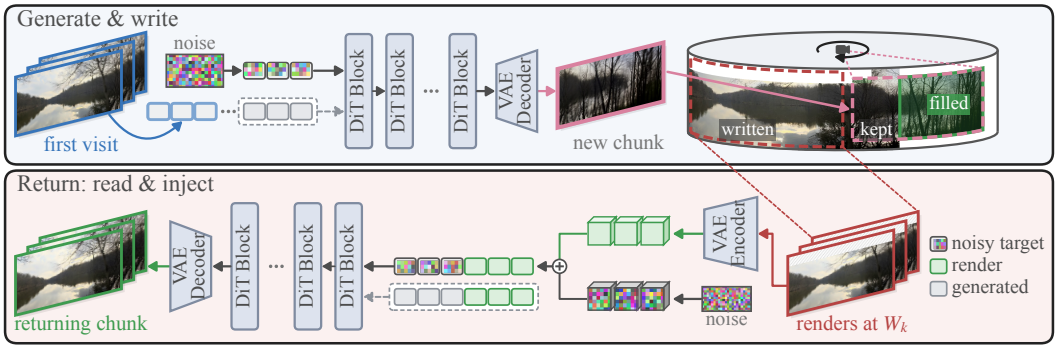

Figure 2: **Overview of CamCanvas.** Top: each generated chunk writes only blank canvas regions, so the first observation of every location is kept. Bottom: on a returning chunk, the canvas is rendered at the commanded pose, encoded, and interleaved with the noisy targets as a noise-free condition; the generator completes the regions the canvas does not cover.

2025a; Yu et al., 2026; Wang et al., 2026b; Garcin et al., 2026; Sun et al., 2025; Ma et al., 2026). These methods extend access to earlier content beyond the recent attention window. However, a retrieved view was observed at another pose, leaving the model to infer the requested view. A return frame can therefore preserve scene appearance without placing the camera at the commanded pose. CamCanvas supplies the commanded view by rendering stored content at that pose.

Rendered conditions and geometric caches. GEN3C and related methods render stored geometry at target views, providing direct visual evidence that improves camera control and geometric consistency (Ren et al., 2025; Yu et al., 2025c;b; Cao et al., 2025; Huang et al., 2025b). GEN3C and Voyager also extend their caches with generated frames (Ren et al., 2025; Huang et al., 2025b). These methods generate offline along trajectories specified in advance, conditioning every frame on rendered content. Interactive generation instead receives commands one at a time, without knowing the future trajectory. Recent streaming methods bring rendered geometry online. GeoStream rebuilds its cache from its own outputs at each refresh (Zhao et al., 2026), while WorldWarp warps cached content into new views as a partially noised condition (Kong et al., 2025). Our canvas preserves each location's first observation instead of refreshing it, so a return reads what the first visit generated. CamCanvas supplies the rendered view as a noise-free condition (Section 3.4).

Evaluating camera control and returns. Camera benchmarks align generated and reference trajectories to remove scale or reference-frame differences (Lin et al., 2025; Duan et al., 2025; Zheng et al., 2024), while revisit protocols compare return frames with earlier observations (Li et al., 2025b; Sun et al., 2025; Li et al., 2026) and long-horizon benchmarks assess content preservation and drift (Xu et al., 2026c; Ying et al., 2026). Revisit protocols compare return frames through image similarity, which mixes viewpoint error with content change (Xu et al., 2026c). Our return test instead uses the pose estimator VGGT (Wang et al., 2025a) to estimate the viewing-angle difference from the return frame and the model's own first-visit frame, without aligning trajectories.

3 CamCanvas

Our key idea is to supply the view at the commanded pose so the model need not infer where the camera is from relative motion. We first establish why relative commands fail at returns (Section 3.2), then describe how the canvas retains generated content and renders it at the commanded pose (Section 3.3). The rendered view conditions generation (Section 3.4) through an attention adapter trained as described in Section 3.5. Figure 2 gives an overview, and Algorithm 1 summarizes generation.

3.1 Preliminaries

We build on block-causal Wan2.2-5B (Team Wan et al., 2025; Huang et al., 2026), which denoises three latent frames z_c per chunk in four steps and decodes them into pixel frames x_c. Its video variational autoencoder (VAE) maps $1 + 4n$ pixel frames to $1 + n$ latent frames and encodes the

first frame y_0 separately. The key–value (KV) window $\mathcal{W}_c$ holds up to 18 latent frames without an attention sink (Xiao et al., 2024), so the first frame is evicted like any other frame. Camera commands enter through PRoPE (Li et al., 2025a; Huang et al., 2026): the tokens of latent frame k carry $P_k = \mathrm{lift}(K)W_k$, where $W_k \in \mathrm{SE}(3)$ is the commanded world-to-camera pose and K holds fixed normalized intrinsics lifted to homogeneous coordinates. Attention therefore depends on poses only through relative products:

$$\ell_{ij} = \mathbf{q}_i^\top (P_i P_j^{-1}) \mathbf{k}_j, \qquad \mathbf{o}_i = \sum_j a_{ij} (P_i P_j^{-1}) \mathbf{v}_j, \tag{1}$$

where i, j index frames, suppressing token indices; $\mathbf{q}_i, \mathbf{k}_j, \mathbf{v}_j$ are query, key, value vectors, ℓ_{ij} the logit, a_{ij} its softmax weight, and $\mathbf{o}_i$ the output. For chunk indices $\mathcal{C}_c$, let $u_c = \{P_i P_j^{-1} : i \in \mathcal{C}_c,\ j \in \mathcal{W}_c\}$. Each chunk is then sampled from

$$p_\theta(z_c \mid z_{\mathcal{W}_c}, u_c), \tag{2}$$

where θ denotes the model parameters and $z_{\mathcal{W}_c}$ the clean context latents in the KV window.

3.2 Returns and Return Blindness

Definition 1 (absolute camera placement). Absolute camera placement is the ability to put the camera at a commanded pose in the generated world, so that the pose alone determines which previously generated content is visible.

Definition 2 (return). A chunk returns when its commanded pose is within 10° and $0.15\bar{D}$ of a pose visited at least two chunks earlier, after departure by at least 25° or $0.4\bar{D}$. Here $\bar{D}$ is the largest distance between a commanded camera center issued so far and the first camera center; the criterion depends only on the commands issued so far.

We call an instantaneous change of the commanded pose between chunks a *camera jump*.

To measure placement, let y_j, y_c denote the model's first-visit and return frames, with y_c the frame of the last latent in the returning chunk x_c. Using the relative-rotation estimator $E(y, y') \in \mathrm{SO}(3)$ and geodesic angle $\angle$ from identity, define

$$e = \angle E(y_j, y_c).$$

The *return error* e is the orientation difference between the return and the model's own first visit, estimated from the frames without alignment; a return succeeds when $e < 10^\circ$. The return error is invariant to rigid motions of the world frame, as Lemma 5 in Appendix A.6 shows.

The following theorem characterizes access to first-visit content outside the attention window.

Theorem 1 (return blindness). *Conditioning only on window latents $z_{\mathcal{W}_c}$ and relative products u_c, an out-of-window first visit $j \notin \mathcal{W}_c$ satisfies $x_c \perp y_j \mid (z_{\mathcal{W}_c}, u_c)$ and, for fixed commands, $I(x_c; y_j) \le I(z_{\mathcal{W}_c}; y_j)$, where I is mutual information.*

The proof appears in Appendix A.2. Theorem 1 establishes that information about an earlier visit is limited to what the window retains, motivating a canvas that supplies first-visit content at the commanded pose.

3.3 Writing and Reading the Canvas

We store generated content in a canvas outside the attention window: adding every stored frame to attention would grow its cost with the number of frames, while a fixed window would still bound access to earlier content (Theorem 1). The canvas state before chunk c is $\mathcal{S}_c = (C_c, M_c, \mathcal{D})$, with panoramic content C_c, a binary observed mask M_c, and depth $\mathcal{D}$. The canvas is initialized from the first frame and extended with decoded outputs; monocular depth estimated on the first frame with MoGe (Wang et al., 2025b) converts commanded translations into image-space displacement.

Writing. We reproject each decoded chunk x_c onto the canvas to obtain content $\widetilde{x}_c$ and an in-view indicator B_c, and update pixelwise:

$$U_c = B_c(1 - M_c), \quad C_{c+1} = (1 - U_c)C_c + U_c\widetilde{x}_c, \quad M_{c+1} = \max(M_c, U_c). \tag{3}$$

The update mask U_c admits only previously unobserved locations, so written content is never overwritten and each return reads content whose error is incurred once, when it is written and read, as Lemma 1 in Appendix A.4 shows.

Reading. For each pixel frame p of a returning chunk, we render $r_p = \Pi(\mathcal{S}_c, W_p^{\text{pix}})$, where Π is the renderer and W_p^{pix} is the commanded pose of that pixel frame, obtained from the sequence W_k. Rotations use spherical resampling, which is locally the homography $H = KRK^{-1}$ for relative rotation R (Hartley & Zisserman, 2004); translations reproject stored content with the depth $\mathcal{D}$ and occlusion checks. The rendered view covers a fraction κ of the target, and the generator completes the remaining $1 - \kappa$. At inference, the canvas is stored at 8192×16384 pixels (height by width).

3.4 Injecting the Commanded View

The rendered view and the frames to be generated share the commanded pose, so $P_i P_{\text{canvas}}^{-1} = \text{Id}$ in PRoPE (equation 1), where Id is the identity matrix. This alignment lets attention reproduce the requested view directly (Proposition 2). Appendix A.5 proves Propositions 1–4.

On six clips at the $90°$ level, offsetting the rendered condition by $20°$ or $40°$ with commands unchanged moves the returned viewpoint by a median $19.2°$ or $40.6°$, against $0.5°$ at the commanded pose (Figure 3). This shows that the rendered view sets where the camera returns.

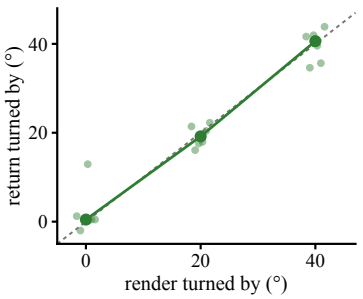

Figure 3: **Returned viewpoint versus render offset.** Six $90°$ returns with fixed commands. Dots: clips; line: medians; dashed: returned offset equal to render offset.

Rather than adding rendered views as extra context frames, we encode the rendered sequence with the target's video VAE and interleave condition and target tokens on the same token grid, each condition token carrying its target's pose. The condition tokens stay at timestep zero, since noise would remove information needed for reproduction (Proposition 1). Unlike pose coordinates without content, which specify where the camera is but supply no appearance, the rendered view provides the content the model must show. We ablate these choices in Table 3.

The following theorem combines canvas writing, reading, and conditioning to bound the return error of CamCanvas.

Theorem 2 (return-error bound). *Under A0, A2, and A3 in Appendix A.1, the return error obeys*

$$e \leq \epsilon_E + L_E(\delta + \epsilon_I) + L_E(1 - \kappa)\Delta. \tag{4}$$

The proof appears in Appendix A.3. Here ϵ_E is the estimator floor, δ the render error, ϵ_I the reproduction error, and κ the covered fraction of the target view. The estimator sensitivity L_E converts image error to angular error, and Δ bounds completion error on uncovered pixels. Appendix A.7 reports the measured terms. Theorem 2 establishes a bound with no rotation angle, elapsed chunk count, or window length on its right-hand side, holding at every return with the same constants (Proposition 4). This motivates training the adapter to reduce the reproduction error ϵ_I.

3.5 Training

We keep the block-causal Wan2.2-5B world model frozen and adapt only its self-attention (Team Wan et al., 2025; Huang et al., 2026). One rank-one update per head suffices to let attention use the aligned condition (Proposition 3), so we attach rank-32 low-rank adapters (Hu et al., 2022) to the query, key, value, and output projections of all 30 self-attention blocks (23.6M trainable parameters); cross-attention and feed-forward layers stay frozen.

We train the adapter with teacher-forced flow matching (Lipman et al., 2023) on 21-latent-frame clips. Each three-frame chunk samples its own diffusion timestep t and is interleaved at timestep zero with its rendered condition r_c, which keeps only content observed in the first frame or in earlier chunks, rendered at the chunk's commanded poses. With $z_c^t = (1 - t)z_c + t\epsilon$ and $\epsilon \sim \mathcal{N}(0, I)$, the

Algorithm 1 Streaming generation with CamCanvas

Require: First frame y_0, commanded poses $\{W_k\}$, frozen generator θ with adapter ψ
Initialize the canvas $\mathcal{S}_0$ from y_0 and its MoGe depth
for $c = 0, 1, \ldots$ **do**
  if chunk c returns (Definition 2) **then**
    $r_p \leftarrow \Pi(\mathcal{S}_c, W_p^{\text{pix}})$ for each pixel frame p of chunk c ▷ render at the commanded pose
    $z_c \sim p_{\theta,\psi}(z_c \mid z_{\mathcal{W}_c}, u_c, r_c)$ ▷ condition at timestep zero
  else
    $z_c \sim p_\theta(z_c \mid z_{\mathcal{W}_c}, u_c)$ ▷ backbone unchanged
  end if
  Decode x_c and update $\mathcal{S}_{c+1}$ by equation 3
  Append z_c to the KV window, evicting frames beyond 18 latents
end for

loss supervises the flow velocity on noisy target tokens only:

$$\mathcal{L}(\psi) = \mathbb{E}_{c,t,\epsilon}\Big[w(t)\,\big\|v_{\theta,\psi}(z_c^t, t \mid z_{<c}, u_c, r_c) - (\epsilon - z_c)\big\|_2^2\Big], \tag{5}$$

where ψ denotes the adapter parameters, $w(t)$ the scheduler weight, and $z_{<c}$ the ground-truth context. Condition tokens and the clean first frame are excluded from the loss, and the first frame and the complete rendered condition are each dropped with probability 0.1. Training runs for 3000 steps (Appendix C). Rotation clips come from approximately 900 CC0 Poly Haven panoramas at 2048×4096, and translation clips from TartanAir and Sekai with MoGe depth (Poly Haven, n.d.; Wang et al., 2020; Li et al., 2025c; Wang et al., 2025b).

SANA-WM uses linear attention (Katharopoulos et al., 2020) and $8\times$ temporal compression (Zhu et al., 2026). Its adapter has rank 32 and its canvas is 2048×4096. Its adapter is trained with rendered views as conditions (Appendix C). For both models, commands identify returning chunks by Definition 2, on which the rendered condition and adapter are active.

4 Experiments

4.1 Setup

Protocols. We evaluate whether models reach the commanded viewpoint and preserve content by commanding the camera away and back, using the protocols detailed in Appendix B. The panorama protocol tests four jump sizes ($51°$–$110°$) and a within-window return on 91 panoramas, with 249 frames per trajectory (Figure 1). Two protocols use 300 clips of the RealEstate10K test set (Zhou et al., 2018): a jump return to an earlier real frame and an out-and-back trajectory. Rotation and translation cycles, in which the camera moves out and retraces its path back, use the same 91 panoramas. We use jumps because they separate reaching the commanded viewpoint from smooth tracking and occur in practice as gaze shifts, snap turns, cuts, and navigation goals (Land, 2004; Farmani & Teather, 2018; Cutting et al., 2010; Savva et al., 2019) (Appendix E).

Metrics. The return test measures the viewing-angle difference e between the return and the model's own first visit, estimated by VGGT without alignment (Wang et al., 2025a); Section 3.2 defines the success threshold. The out-of-window fraction ϕ is the share of the target view outside the attention window. PSNR and LPIPS (Zhang et al., 2018) compare returns with the real frames on RealEstate10K, which tests agreement with the observed scene, and with the paired first-visit frames on cycles. We also report MEt3R (Asim et al., 2025) for content consistency and mean return error $\bar{e}$ on cycles and out-and-back trajectories.

Methods. We compare CamCanvas with the Wan2.2-5B block-causal base model (Team Wan et al., 2025; Huang et al., 2026), MiniMax-H3-33B (Huang et al., 2026), LingBot-World (Gao et al., 2026), HY-WorldPlay with retrieval-based memory (Sun et al., 2025), GEN3C, a bidirectional video model that renders a 3D cache into every frame (Ren et al., 2025), and CamCanvas w/o canvas. All models receive the same commands and first frames through their own inference code. CamCanvas w/o canvas uses the same adapter, data and schedule, trained and run without the canvas condition. Wan2.2-5B, CamCanvas w/o canvas, and CamCanvas share the backbone, prompts and seeds, pro-

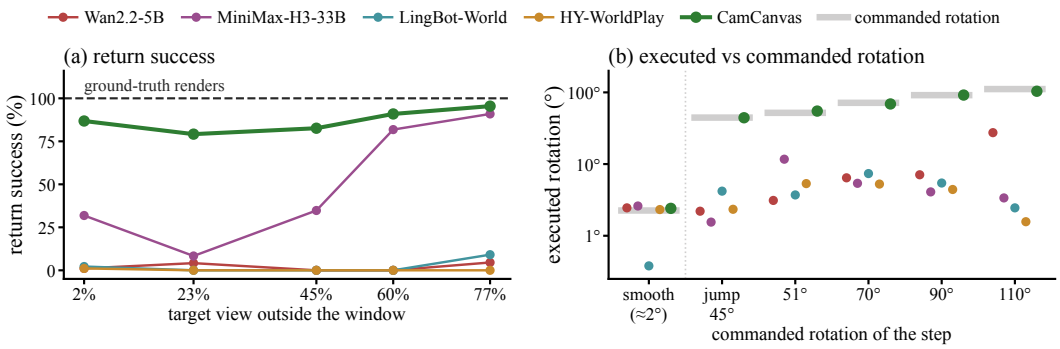

Figure 4: **Return success and executed rotation.** (a) Return success ($e < 10°$) versus the out-of-window fraction ϕ; ground-truth renders reach 100%. (b) Median executed rotation per step, grouped by commanded rotation (log scale); grey bars mark the commands. Existing models follow smooth steps but rotate far less than commanded on jumps, while CamCanvas turns by about the commanded amount.

ducing identical frames before the first return (Appendix A.4). All results are averaged over three random seeds.

4.2 Main Results

Existing models. We observe that existing models follow small smooth commands but rotate far less than commanded on jumps (Figure 4(b)). Wan2.2-5B, MiniMax-H3-33B and HY-WorldPlay follow commands of about $2°$ per step, while LingBot-World turns $0.4°$. Their median jump rotations are 1.6–$11.7°$, except for Wan2.2-5B's $27.4°$ at $110°$, while CamCanvas turns nearly as commanded. Across four backbones from three families, the return test exposes failures to reach the commanded viewpoint (Figure 4(a) and Table 9). Wan2.2-5B succeeds on at most 5% per level, with median errors 54–$70°$; within-window returns fail as often as out-of-window returns. This indicates that camera-following metrics averaged over trajectories made mostly of small steps can look good while returns fail, even when the target remains in the window.

Panorama jump returns. As shown in Table 1, CamCanvas outperforms the compared methods in return success across all levels, achieving 79–95% against 0–9% for Wan2.2-5B, LingBot-World, HY-WorldPlay and CamCanvas w/o canvas (Figure 4(a)). Those models end 39–$124°$ away from their own first visit, and HY-WorldPlay produces a plausible view at the wrong pose (Figure 5). MiniMax-H3-33B keeps its first frame in attention, so its success rises only as the target moves closer to that frame, not because the camera reaches the commanded pose. GEN3C, which receives the whole trajectory in advance and conditions every frame, succeeds on 41–84% of returns, with 41% success at the $90°$ level. These results highlight the benefit of supplying the commanded view.

Real video. On real video, CamCanvas succeeds on 99% of jump returns to the first frame against 17% for the base model (Table 2), with PSNR 20.3 against 9.8 dB and LPIPS 0.205 against 0.712 (Figure 5). The streaming and retrieval-based baselines achieve 40–44% success on the same jump return. On out-and-back trajectories, CamCanvas reduces $\bar{e}$ to $2.6°$ from the base model's $4.9°$, compared with 12.9–$23.7°$ for those baselines. Figure 7 in Appendix D shows content consistency along return paths. This indicates that CamCanvas reaches the commanded viewpoint and reproduces the real scene more closely.

Rotation and translation cycles. Table 10 (Appendix D) shows that CamCanvas achieves the lowest mean return error on both cycles and improves paired-frame agreement over Wan2.2-5B, with rotation-cycle PSNR 18.9 against 15.1 dB. HY-WorldPlay retrieves stored views, yet its rotation-cycle $\bar{e}$ is $27.5°$ against CamCanvas's $3.2°$. Although LingBot-World achieves rotation-cycle PSNR 23.4 dB, LPIPS 0.164 and MEt3R 0.070, its camera turns only $1.7°$ of the commanded $40°$, leaving paired return and outbound frames at nearly the same viewpoint. Content measures alone therefore cannot distinguish preserved content from a camera that barely moves, so we report them alongside return error and executed rotation.

Table 1: **Panorama jump returns.** S: success (%, $e < 10°$); e: median orientation error (°) against the model's own first visit; ϕ: out-of-window fraction. Entries are means over three seeds, and ground-truth renders give the estimator floor. Best results are in bold.

	within window $\phi = 2\%$		51° $\phi = 23\%$		70° $\phi = 45\%$		90° $\phi = 60\%$		110° $\phi = 77\%$	
Method	S↑	$e\downarrow$	S↑	$e\downarrow$	S↑	$e\downarrow$	S↑	$e\downarrow$	S↑	$e\downarrow$
Ground-truth renders	*100*	*0.2*	*100*	*0.2*	*100*	*0.1*	*100*	*0.1*	*100*	*0.1*
Wan2.2-5B	1	53.8	4	60.4	0	70.0	0	70.0	5	53.5
MiniMax-H3-33B	32	19.0	8	24.1	35	18.9	82	3.5	91	2.0
LingBot-World	2	64.6	0	44.3	0	62.8	0	74.2	9	58.1
HY-WorldPlay	1	39.3	0	42.7	0	63.5	0	81.4	0	124.1
GEN3C	84	**0.6**	75	**0.4**	65	**1.2**	41	36.8	64	2.6
CamCanvas w/o canvas	1	52.6	4	63.1	0	72.2	0	48.3	9	72.3
CamCanvas	**87**	4.6	**79**	3.8	**83**	3.3	**91**	**1.1**	**95**	**0.4**

Table 2: **Returns on RealEstate10K test clips.** Jump returns to the first frame cover 150 clips, and out and back covers 300. S: success (%, $e < 10°$); $e/\bar{e}$: median/mean orientation error (°) against the model's own first visit. PSNR (dB) and LPIPS compare returns with the real frames. Best results are in bold.

	Jump return to the first frame				Out and back			
Method	S↑	$e\downarrow$	PSNR↑	LPIPS↓	$\bar{e}\downarrow$	PSNR↑	LPIPS↓	MEt3R↓
Wan2.2-5B	17	21.5	9.8	0.712	4.9	10.6	0.647	0.337
HY-WorldPlay	44	12.5	10.0	0.701	12.9	9.8	0.719	0.304
LingBot-World	40	11.5	12.7	0.566	23.7	11.0	0.708	0.414
CamCanvas w/o canvas	16	21.7	10.0	0.712	4.8	10.9	0.637	0.336
CamCanvas	**99**	**0.1**	**20.3**	**0.205**	**2.6**	**13.1**	**0.519**	**0.250**

Transfer to another architecture. CamCanvas also improves the rotation cycle on SANA-WM (Zhu et al., 2026), raising PSNR from 15.2 to 19.4 dB and reducing $\bar{e}$ from 7.5° to 3.3° (Table 11). The same rendered condition and the same rule that selects returning chunks from commands carry over to linear attention and $8\times$ temporal compression (Appendix C).

4.3 Ablations

To validate each component, we build CamCanvas up from the base model and then alter its canvas, presented in Table 3; all variants use the same returning chunks on the 91 panoramas (Table 5).

Building up from the base model. Supplying rendered views as extra context frames already enables returns whose target stays in the window (82% against 1%), but gives 0% at every out-of-window level: as context frames, the render is not aligned with the target tokens, so attention does not reproduce it (Section 3.4). In contrast, replacing the context frames with the aligned noise-free condition markedly improves out-of-window success to 50–77%, even without adaptation. Adding the attention adapter further improves success at every level, from 50% to 91% at 90° and from 77% to 95% at 110°.

Changing the canvas. Zeroing the condition at inference reduces success to 0% at 90° and 110°, showing that returns depend on the supplied content. Replacing the content with coordinates, or keeping the canvas for only one chunk, degrades success as ϕ grows and reaches 0% at 90° and 110°: coordinates carry no appearance, and these returns need content older than one chunk.

5 Conclusion

In this paper, we introduced CamCanvas, which enables absolute camera placement: a camera commanded back to an earlier pose shows the content generated there. We store the model's own gen-



Figure 5: **Qualitative comparison of first visits and returns.** Rows show ground truth, Wan2.2-5B, HY-WorldPlay, LingBot-World, and CamCanvas on 90° and 110° panorama jumps and on RealEstate10K jumps back to the first frame. Blue and red frames mark first visits and returns; corner numbers give each method's orientation error from its own first visit. CamCanvas returns to the first-visit viewpoint.

Table 3: **Ablations on panorama jump returns.** Top: components added to the base model, each row adding to the one above (the aligned condition replaces the context frames). Bottom: changes to the canvas of CamCanvas. S, e, and ϕ as in Table 1; entries are means over three seeds. Best results are in bold.

	within window $\phi = 2\%$		51° $\phi = 23\%$		70° $\phi = 45\%$		90° $\phi = 60\%$		110° $\phi = 77\%$	
Method	S↑	e ↓	S↑	e ↓	S↑	e ↓	S↑	e ↓	S↑	e ↓
Adding components to the base model										
Wan2.2-5B	1	53.8	4	60.4	0	70.0	0	70.0	5	53.5
+ render as context frames	82	4.7	0	25.9	0	46.3	0	66.6	0	82.4
+ aligned noise-free condition	79	**4.6**	67	5.9	57	7.4	50	9.7	77	2.7
+ attention adapter (CamCanvas)	**87**	**4.6**	**79**	**3.8**	**83**	**3.3**	**91**	**1.1**	**95**	**0.4**
Changing the canvas of CamCanvas										
Condition zeroed at inference	20	18.8	17	24.4	4	34.1	0	39.1	0	49.5
Coordinates without content	41	12.2	29	22.5	13	32.4	0	40.6	0	68.7
Canvas kept for one chunk	55	7.7	21	36.0	4	57.4	0	76.8	0	89.8

erated content in a panoramic canvas that preserves each location's first observation. We render it at the commanded pose and supply the rendered view as a noise-free condition on returning chunks. Relative pose encodings leave an out-of-window first visit inaccessible beyond what the window retains (Theorem 1). The rendered view shares the commanded pose with the frames to be generated, so the model follows the condition instead of inferring pose from relative motion. Our experiments show substantially higher return success on panorama and real-video returns, with continued effectiveness on a different architecture, SANA-WM. Our return test measures this capability through the viewing-angle difference between the return and the model's own first visit, estimated from frames without alignment.

Limitations and future work. CamCanvas assumes static scenes and is evaluated on trajectories of up to 249 frames; extending it to dynamic scenes and longer, larger explorations is a natural next step.

AI USE STATEMENT

In this work, we used generative AI tools only to polish the writing, including correcting grammar, improving sentence fluency, and suggesting word choices. The tools were not used to generate research ideas, design or run experiments, implement methods, analyze results, or write proofs. All content was written, checked, and approved by the authors, who take full responsibility for the paper.

ETHICS STATEMENT

This work complies with the ICLR Code of Ethics. It involves no human subjects, human evaluation, or collection of personal data. Rotation training uses Poly Haven panoramas released under CC0 (Poly Haven, n.d.); RealEstate10K, TartanAir, and Sekai are publicly available datasets used under their respective licenses (Zhou et al., 2018; Wang et al., 2020; Li et al., 2025c), and RealEstate10K is released under a Creative Commons Attribution 4.0 license. As with other video generation models, our method could be misused to produce misleading synthetic video, and safeguards such as watermarking and output filtering should accompany its deployment.

REPRODUCIBILITY STATEMENT

Section 3 and Algorithm 1 describe the method. Appendix A states the assumptions and gives complete proofs of all theoretical results. Appendix B specifies the evaluation protocols, the rule that selects returning chunks, and the scoring. Appendix C details the architecture, training objective, data composition, and optimization settings (Table 7), and Appendix F lists the identifiers of all RealEstate10K evaluation clips. All results are averaged over three random seeds.

REFERENCES

Eloi Alonso, Adam Jelley, Vincent Micheli, Anssi Kanervisto, Amos Storkey, Tim Pearce, and François Fleuret. Diffusion for world modeling: Visual details matter in Atari. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Mohammad Asim, Christopher Wewer, Thomas Wimmer, Bernt Schiele, and Jan Eric Lenssen. MEt3R: Measuring multi-view consistency in generated images. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Jianhong Bai, Menghan Xia, Xiao Fu, Xintao Wang, Lianrui Mu, Jinwen Cao, Zuozhu Liu, Haoji Hu, Xiang Bai, Pengfei Wan, and Di Zhang. Recammaster: Camera-controlled generative rendering from a single video. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

Jake Bruce, Michael Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, Yusuf Aytar, Sarah Bechtle, et al. Genie: Generative interactive environments. In *International Conference on Machine Learning (ICML)*, 2024.

Chenjie Cao, Jingkai Zhou, Shikai Li, Jingyun Liang, Chaohui Yu, Fan Wang, Xiangyang Xue, and Yanwei Fu. Uni3C: Unifying Precisely 3D-Enhanced Camera and Human Motion Controls for Video Generation. In *ACM SIGGRAPH Asia Conference Papers*, 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 (NeurIPS)*, 2024.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley, 2 edition, 2006.

Justin Cui, Jie Wu, Ming Li, Tao Yang, Xiaojie Li, Rui Wang, Andrew Bai, Yuanhao Ban, and Cho-Jui Hsieh. Self-forcing++: Towards minute-scale high-quality video generation. In *International Conference on Learning Representations (ICLR)*, 2026.

James E. Cutting, Jordan E. DeLong, and Christine E. Nothelfer. Attention and the evolution of Hollywood film. *Psychological Science*, 2010.

DreamX Team, Yancheng Bai, Rui Chen, Xiangxiang Chu, Rujing Dang, Hao Dou, Bingjie Gao, Qiwen Gu, Siyu Hong, Jiachen Lei, Geng Li, Jifan Li, et al. Dreamx-world 1.0: A general-purpose interactive world model. *arXiv preprint arXiv:2606.16993*, 2026.

Haoyi Duan, Hong-Xing Yu, Sirui Chen, Fei-Fei Li, and Jiajun Wu. WorldScore: A unified evaluation benchmark for world generation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

Yasin Farmani and Robert J. Teather. Viewpoint snapping to reduce cybersickness in virtual reality. In *Graphics Interface*, 2018.

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

Samuel Garcin, Thomas Walker, Steven McDonagh, Tim Pearce, Hakan Bilen, Tianyu He, Kaixin Wang, and Jiang Bian. Beyond pixel histories: World models with persistent 3D state. In *International Conference on Machine Learning (ICML)*, 2026.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Yoav HaCohen, Benny Brazowski, Nisan Chiprut, Yaki Bitterman, Andrew Kvochko, Avishai Berkowitz, Daniel Shalem, Daphna Lifschitz, Dudu Moshe, Eitan Porat, Eitan Richardson, Guy Shiran, Itay Chachy, Jonathan Chetboun, Michael Finkelson, Michael Kupchick, Nir Zabari, Nitzan Guetta, Noa Kotler, Ofir Bibi, Ori Gordon, Poriya Panet, Roi Benita, Shahar Armon, Victor Kulikov, Yaron Inger, Yonatan Shiftan, Zeev Melumian, and Zeev Farbman. LTX-2: Efficient joint audio-visual foundation model. *arXiv preprint arXiv:2601.03233*, 2026.

Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2 edition, 2004.

Mary Hayhoe and Dana Ballard. Eye movements in natural behavior. *Trends in Cognitive Sciences*, 2005.

Hao He, Yinghao Xu, Yuwei Guo, Gordon Wetzstein, Bo Dai, Hongsheng Li, and Ceyuan Yang. CameraCtrl: Enabling camera control for video diffusion models. In *International Conference on Learning Representations (ICLR)*, 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 *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025b.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2015.

Xianglong He, Chunli Peng, Zexiang Liu, Boyang Wang, Yifan Zhang, Qi Cui, Fei Kang, Biao Jiang, Mengyin An, Yangyang Ren, et al. Matrix-Game 2.0: An Open-Source Real-Time and Streaming Interactive World Model. *arXiv preprint arXiv:2508.13009*, 2025c.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022.

Junchao Huang, Xinting Hu, Boyao Han, Shaoshuai Shi, Zhuotao Tian, Tianyu He, and Li Jiang. Memory forcing: Spatio-temporal memory for consistent scene generation on Minecraft. *arXiv preprint arXiv:2510.03198*, 2025a.

Junchao Huang, Guian Fang, Shengju Qian, Xianghao Kong, Zhuoran Zhao, Wei Huang, Yihua Du, Zixin Zhang, Justin Cui, Yuchao Gu, Yukang Chen, Xinting Hu, Tianyu He, Shaoshuai Shi, Zhuotao Tian, Xin Wang, Mike Zheng Shou, and Li Jiang. SolarWM: Open Data and Scalable Training for Long-Horizon Video World Models. *arXiv preprint arXiv:2609.02886*, 2026.

Tianyu Huang, Wangguandong Zheng, Tengfei Wang, Yuhao Liu, Zhenwei Wang, Junta Wu, Jie Jiang, Hui Li, Rynson Lau, Wangmeng Zuo, et al. Voyager: Long-range and world-consistent video diffusion for explorable 3D scene generation. *ACM Transactions on Graphics*, 2025b.

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 (NeurIPS)*, 2025c.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are RNNs: Fast autoregressive transformers with linear attention. In *International Conference on Machine Learning (ICML)*, 2020.

Hanyang Kong, Xingyi Yang, Xiaoxu Zheng, and Xinchao Wang. WorldWarp: Propagating 3D geometry with asynchronous video diffusion. *arXiv preprint arXiv:2512.19678*, 2025.

Michael F. Land. The coordination of rotations of the eyes, head and trunk in saccadic turns produced in natural situations. *Experimental Brain Research*, 2004.

Jia Li, Han Yan, Yihang Chen, Siqi Li, Xibin Song, Yifu Wang, Jianfei Cai, Tien-Tsin Wong, and Pan Ji. I3DM: Implicit 3D-aware Memory Retrieval and Injection for Consistent Video Scene Generation. *arXiv preprint arXiv:2603.23413*, 2026.

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 (NeurIPS)*, 2025a.

Runjia Li, Philip Torr, Andrea Vedaldi, and Tomas Jakab. VMem: Consistent Interactive Video Scene Generation with Surfel-Indexed View Memory. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025b.

Zhen Li, Chuanhao Li, Xiaofeng Mao, Shaoheng Lin, Ming Li, Shitian Zhao, Zhaopan Xu, Xinyue Li, Yukang Feng, Jianwen Sun, et al. Sekai: A Video Dataset towards World Exploration. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2025c.

Zhiqiu Lin, Siyuan Cen, Daniel Jiang, Jay Karhade, Hewei Wang, Chancharik Mitra, Yu Tong Tiffany Ling, Yuhan Huang, Rushikesh Zawar, Xue Bai, Yilun Du, Chuang Gan, and Deva Ramanan. Towards understanding camera motions in any video. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2025.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations (ICLR)*, 2023.

Kunhao Liu, Wenbo Hu, Jiale Xu, Ying Shan, and Shijian Lu. Rolling forcing: Autoregressive long video diffusion in real time. In *International Conference on Learning Representations (ICLR)*, 2026.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

Wenchao Ma, Changran Liu, Sharon X. Huang, and Haomiao Jiang. Closing the Loop: Training-Free Revisit Consistency for Autoregressive Generative Rendering. *arXiv preprint arXiv:2607.21848*, 2026.

Poly Haven. Poly Haven: The public 3d asset library. `https://polyhaven.com/`, n.d. CC0 assets.

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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra. Habitat: A platform for embodied AI research. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

Vincent Sitzmann, Ana Serrano, Amy Pavel, Maneesh Agrawala, Diego Gutierrez, Belen Masia, and Gordon Wetzstein. Saliency in VR: How do people explore virtual environments? *IEEE Transactions on Visualization and Computer Graphics*, 2018.

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.

Team Wan, Ang Wang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1991.

Dani Valevski, Yaniv Leviathan, Moab Arar, and Shlomi Fruchter. Diffusion models are real-time game engines. In *International Conference on Learning Representations (ICLR)*, 2025.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. VGGT: Visual geometry grounded transformer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025a.

Ruicheng Wang, Sicheng Xu, Cassie Dai, Jianfeng Xiang, Yu Deng, Xin Tong, and Jiaolong Yang. MoGe: Unlocking Accurate Monocular Geometry Estimation for Open-Domain Images with Optimal Training Supervision. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025b.

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. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 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 Conference Papers*, 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, et al. Matrix-game 3.0: Real-time and streaming interactive world model with long-horizon memory. *arXiv preprint arXiv:2604.08995*, 2026a.

Zun Wang, Han Lin, Jaehong Yoon, Jaemin Cho, Yue Zhang, and Mohit Bansal. AnchorWeave: World-consistent video generation with retrieved local spatial memories. In *European Conference on Computer Vision (ECCV)*, 2026b.

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 (NeurIPS)*, 2025.

Chendong Xiang, Jiajun Liu, Jintao Zhang, Xiao Yang, Zhengwei Fang, Shizun Wang, Zijun Wang, Yingtian Zou, Hang Su, and Jun Zhu. Geometry-Aware Rotary Position Embedding for Consistent Video World Model. *arXiv preprint arXiv:2602.07854*, 2026.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *International Conference on Learning Representations (ICLR)*, 2024.

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

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*, 2026a.

Tian-Xing Xu, Zi-Xuan Wang, Guangyuan Wang, Li Hu, Zhongyi Zhang, Peng Zhang, Bang Zhang, and Song-Hai Zhang. UCM: Unified Modeling of Camera Control and Memory with Time-aware Positional Encoding Warping for World Models. In *ACM SIGGRAPH Conference Papers*, 2026b.

Ting-Bing Xu, Jiacheng Sui, Zhe Gao, Kewei Shi, Wenjin Yang, Zhicheng Liu, Zhaoxu Sun, Mingchao Sun, Hongyu Pan, Fan Jiang, Mu Xu, Qi Fan, Yang Gao, Yong Li, and Baoquan Chen. WorldRoamBench: An Open-World Benchmark for Long-Horizon Stability of Interactive World Models. *arXiv preprint arXiv:2606.31672*, 2026c.

Shuai Yang, Wei Huang, Ruihang Chu, Yicheng Xiao, Yuyang Zhao, Xianbang Wang, Muyang Li, Enze Xie, Yingcong Chen, Yao Lu, Song Han, and Yukang Chen. Longlive: Real-time interactive long video generation. In *International Conference on Learning Representations (ICLR)*, 2026.

Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Frédo Durand, and William T. Freeman. Improved distribution matching distillation for fast image synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Tianwei Yin, Qiang Zhang, Richard Zhang, William T Freeman, Fredo Durand, Eli Shechtman, and Xun Huang. From Slow Bidirectional to Fast Autoregressive Video Diffusion Models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Kaining Ying, Hengrui Hu, Siyu Ren, Jiamu Li, Fengjiao Chen, Ziwen Wang, Xuezhi Cao, Xunliang Cai, and Henghui Ding. WBench: A comprehensive multi-turn benchmark for interactive video world model evaluation. *arXiv preprint arXiv:2605.25874*, 2026.

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 *ACM SIGGRAPH Asia Conference Papers*, 2025a.

Mark Yu, Wenbo Hu, Jinbo Xing, and Ying Shan. TrajectoryCrafter: Redirecting camera trajectory for monocular videos via diffusion models. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025b.

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. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025c.

Wei Yu, Runjia Qian, Yumeng Li, Liquan Wang, Songheng Yin, Sri Siddarth Chakaravarthy P, Dennis Anthony, Yang Ye, Yidi Li, Weiwei Wan, and Animesh Garg. MosaicMem: Hybrid Spatial Memory for Controllable Video World Models. *arXiv preprint arXiv:2603.17117*, 2026.

Jiahan Zhang, Muqing Jiang, Nanru Dai, Taiming Lu, Arda Uzunoglu, Shunchi Zhang, Yana Wei, Jiahao Wang, Vishal M. Patel, Paul Pu Liang, Daniel Khashabi, Cheng Peng, et al. World-in-world: World models in a closed-loop world. In *International Conference on Learning Representations (ICLR)*, 2026.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Yizhou Zhao, Yifan Wang, Xiaoyuan Wang, Yushu Wu, Hao Zhang, Moayed Haji-Ali, Rameen Abdal, Ashkan Mirzaei, Yanyu Li, Willi Menapace, Laszlo Jeni, Sergey Tulyakov, Peter Wonka, and Chaoyang Wang. GeoStream: Toward precise camera controlled streaming video generation. *arXiv preprint arXiv:2606.15162*, 2026.

Guangcong Zheng, Teng Li, Rui Jiang, Yehao Lu, Tao Wu, and Xi Li. CamI2V: Camera-Controlled Image-to-Video Diffusion Model. *arXiv preprint arXiv:2410.15957*, 2024.

Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *ACM Transactions on Graphics*, 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 Theory and Proofs

A.1 Assumptions and Scope

The results below hold under the following modeling assumptions. Fix a static pinhole scene, intrinsics K, and a command sequence $W_k \in \mathrm{SE}(3)$. For a returning chunk c, y_c^* denotes the ideal view at the commanded pose of the content that the model's own first visit represents (A2). The generator has conditional law $p_\theta(z_c \mid z_{\mathcal{W}_c}, u_c)$ of equation 2, with fresh sampling randomness independent of earlier frames given this conditioning set. Write f for the backbone generator of decoded chunks, p_f for the induced conditional law of decoded chunk x_c, and y_c for its return frame. Commands and other fixed inputs are suppressed in information-theoretic expressions. If commands depend on generated content, they are included in the conditioning set. A return is defined by Section 3.2. Theorem 1 additionally requires $j \notin \mathcal{W}_c$.

A0 (estimator). Let d be the geodesic metric on $\mathrm{SO}(3)$, so $\angle R = d(\mathrm{Id}, R)$. On ideal image pairs, the relative-rotation estimator E has error at most ϵ_E. On the image domain under consideration, it satisfies $d(E(y_1, y), E(y_1, y')) \le L_E \|y - y'\|$, where the norm is mean absolute RGB difference with intensities in $[0, 1]$. Relative rotations used together in the rotation completion ratio are obtained from a common set of estimated orientations. The estimator floor is measured with ground-truth renders with the same estimator.

A1 (small camera motion in training). Large relative camera changes exceed the small motions represented by the training trajectories and can be executed only partly by the learned mapping from relative motion to the view. In our training trajectories, the rotation between adjacent latent frames has a median of $1.89°$ and a 90th percentile of $4.73°$. A1 supports the reading of Theorem 1 and is not used in the proof of Theorem 2.

A2 (rendering). On static, jointly visible content with fixed geometry and visibility, the rendered condition $r_c = \Pi(\mathcal{S}_c, W_{p_c}^{\mathrm{pix}})$ satisfies $\|r_c - y_c^*\|_{\Omega_c} \le \delta$, where p_c is the pixel frame of y_c, Ω_c is the covered target region and its area fraction is $\kappa \in [0, 1]$. The restricted norm averages over covered pixels. Rotation induces $H = KRK^{-1}$, with error from writing and resampling. Translation additionally includes depth error (Hartley & Zisserman, 2004). For a return scored against the model's own first visit, the ideal target is the view represented by the first observation, so δ bounds reconstruction relative to that observation.

A3 (reproduction and completion). On covered pixels, $\|y_c - r_c\|_{\Omega_c} \le \epsilon_I$. On uncovered pixels, completion differs from the reference view by mean absolute error at most Δ. ϵ_I is measured on the trained adapter (Appendix A.7). A0 must apply to the ideal view given by the model's own first visit in A2. The zero-target-angle bound below concerns same-pose returns. The tolerance within which a command counts as a return under Definition 2 does not make nearby poses identical. For a nonzero target separation, its angle must be added to a bound on raw return error.

A.2 Return Blindness

Theorem 1 (return blindness). *If f is conditioned only on $(z_{\mathcal{W}_c}, u_c)$ and $j \notin \mathcal{W}_c$, then*

$$x_c \perp y_j \mid (z_{\mathcal{W}_c}, u_c), \qquad I(x_c; y_j) \le I(z_{\mathcal{W}_c}; y_j).$$

The information inequality is for fixed commands. Return content can reproduce only information about the first visit retained in the window. The camera input specifies relative transforms to available frames, so returning to the first-visit view still requires the generator to execute those transforms.

Proof. The conditional law defining f gives $p_f(x_c \mid z_{<c}, u_c) = p_f(x_c \mid z_{\mathcal{W}_c}, u_c)$, where $z_{<c}$ denotes the full prior latent history. Including y_j cannot change this law once the window and commands are fixed, proving conditional independence. The resulting Markov chain $y_j \to z_{\mathcal{W}_c} \to x_c$ yields the information inequality by data processing (Cover & Thomas, 2006). Equivalently, draw $\widetilde{y}_j$ independently from $p(y_j \mid z_{\mathcal{W}_c})$ conditional on the window. Replacing y_j by $\widetilde{y}_j$ leaves the joint distribution with x_c unchanged, and therefore leaves every measurable return-consistency statistic unchanged. Finally, each camera factor in equation 1 relates the target to an available frame. No such factor supplies the missing first-visit appearance as a target view, so the camera input alone cannot supply the first-visit view.

Scope. Retained information can still support a correct return: the theorem rules out access to first-visit information beyond that retained information. Attention sinks (Xiao et al., 2024) or global memory tokens that preserve the first frame violate $j \notin \mathcal{W}_c$ for $j = 0$, where the conditioning set must include all retained memory. Retrieval-based memory restores a first-visit frame to that set, but still requiring the model to learn to map relative motion to the view (Xiao et al., 2025; Yu et al., 2025a; Sun et al., 2025). Retrieval supplies what the return should show and assumes the camera is at the commanded pose. Placement requires the commanded view itself as the alignment target. Rendered conditions, including geometric rendering in GEN3C (Ren et al., 2025), instead supply the view at the commanded pose.

A.3 Return-Error Bound

Theorem 2 (return-error bound). *Under A0, A2, and A3, with the first-visit frame as the reference (A2),*

$$e \le \epsilon_E + L_E(\delta + \epsilon_I) + L_E(1-\kappa)\Delta, \qquad |\rho - 1| \le \frac{e}{\theta_{\text{need}}}.$$

Here $\theta_{\text{need}} > 0$, and all angular quantities use the same units. The bound has no explicit dependence on jump angle, the number of chunks since the first visit, or window length.

Proof. Let y_c^* be the ideal same-pose reconstruction of the first-visit reference. A2 and A3 give $\|y_c - y_c^*\|_{\Omega_c} \le \delta + \epsilon_I$ by the triangle inequality. Splitting the full-image mean over covered and uncovered pixels gives

$$\|y_c - y_c^*\| \le \kappa(\delta + \epsilon_I) + (1-\kappa)\Delta \le \delta + \epsilon_I + (1-\kappa)\Delta.$$

By A0, $\angle E(y_j, y_c^*) \le \epsilon_E$, and perturbing the second image changes the estimated rotation by at most $L_E\|y_c - y_c^*\|$. A second triangle inequality therefore gives equation 4. The reference is the first visit itself: no discrepancy between that generated scene and a different external scene enters this comparison.

The ratio follows from the reverse triangle inequality for rotations. Write $\widehat{R}_{\text{pre}}$, $\widehat{R}_j$, and $\widehat{R}_c$ for the common estimated orientations. Then

$$\left| d(\widehat{R}_{\text{pre}}, \widehat{R}_c) - d(\widehat{R}_{\text{pre}}, \widehat{R}_j) \right| \le d(\widehat{R}_j, \widehat{R}_c) = e.$$

Division by θ_{need} yields the stated bound. Same-pose condition tokens make their PRoPE transform the identity, so this argument introduces no term that depends on the jump angle.

A.4 Writing and Identity Before the First Return

Lemma 1 (written content is fixed). *Equation 3 preserves every written pixel through all subsequent writes. On already covered regions with fixed geometry, visibility, and sampling, repeated reads have a single write/read error δ. Overwriting on every visit admits an additive error recurrence and a linearly accumulating bound.*

Proof. If canvas location s is first written at chunk c_0, then $M_{c_0+1}(s) = 1$. Every subsequent update satisfies $U_c(s) = B_c(s)(1 - M_c(s)) = 0$, so $C_{c+1}(s) = C_c(s)$. Induction preserves the first written value. A fixed read of those locations therefore returns the same samples regardless of intervening chunks. Locations not yet observed may still be written later. With a last-write update, $U_c = B_c$ and the stored value depends on newly generated content. Under a nonexpansive read/write map with additional error at most δ' per overwrite, the accumulated error satisfies $\delta_c \le \delta_{c-1} + \delta'$, hence $\delta_n \le \delta_0 + n\delta'$ after n overwrites.

Lemma 2 (identity before the first return). *Let $\mathcal{R}$ be the returning chunks selected by Definition 2. For equal initial inputs and random draws, the CamCanvas generator f_G equals the backbone f exactly for every $c < \min \mathcal{R}$. Differences can occur only on returns or their subsequent chunks. A trajectory with no return remains identical throughout.*

Proof. Before the first return, both executions start with the same input latents and cache. On a non-returning chunk, the adapter scale is zero, no canvas tokens enter attention, and the original window and backbone operations are used. Equal inputs and random draws therefore produce equal

outputs and the same next cache. Induction proves identity up to the first return. Canvas writes occur outside the network and cannot alter this induction while rendered conditions are absent. A return can change output and thus later cached history, giving support contained in $\mathcal{R} \cup \{c : c > \min \mathcal{R}\}$. The rule depends only on the command prefix. When no prefix satisfies Definition 2, no return is marked. The same induction applies to the CamCanvas w/o canvas variant with its adapter active only at returns, so all differences among the three variants originate at returns.

A.5 Design of the Condition

Proposition 1 (noise-free condition). *The timestep-zero condition minimizes the attainable reproduction error.*

Proof. On the covered region, the Gaussian render-noise model states that the target latent equals the render up to noise: $z = r + \tau\xi$, $\xi \sim \mathcal{N}(0, I)$, with $\tau \geq 0$ the noise scale. The forward process is $z_t = \alpha_t z + \sigma_t \varepsilon$.

Clean injection. Conditioning on (z_t, r),

$$\mathbb{E}[z \mid z_t, r] = r + \frac{\alpha_t \tau^2}{\alpha_t^2 \tau^2 + \sigma_t^2}\,(z_t - \alpha_t r), \qquad \text{mmse} = \frac{\tau^2 \sigma_t^2}{\alpha_t^2 \tau^2 + \sigma_t^2} \leq \tau^2.$$

As $\tau \to 0$ the Bayes-optimal denoiser output is the render r itself.

Condition supplied at the target's noise level. If the condition enters as $r'_t = \alpha_t r + \sigma_t \varepsilon'$ with prior $r \sim \mathcal{N}(\mu, s^2 I)$, then $\hat{r} = \mathbb{E}[r \mid r'_t]$ has residual variance $\nu_t^2 = s^2\sigma_t^2/(\alpha_t^2 s^2 + \sigma_t^2) > 0$, and the attainable error becomes $\tau^2 + \nu_t^2$.

Conclusion. Noise-free injection keeps the error at most τ^2; injecting the condition at the target's noise level adds the residual variance of the forward process, with equality only at $\sigma_t = 0$. This error corresponds to ϵ_I in A3, so the noise-free condition minimizes the reproduction term of Theorem 2.

Scope. This concerns the Bayes-optimal denoiser under a Gaussian render-noise model, not the trained network.

Proposition 2 (pose alignment). *When condition and target share a pose, the target tokens can be formed from the condition tokens by attention weights alone; otherwise, forming them requires a scene-dependent reprojection.*

Proof. With PRoPE, $o_i = \sum_j a_{ij}(P_i P_j^{-1})v_j$.

Aligned case. A condition token rendered at the target's commanded pose carries that pose, so $P_i P_j^{-1} = \mathrm{Id}$ and the condition grid is the target grid. Choosing a_{ij} concentrated on the aligned token gives $o_i = v_{j(i)}$: the target tokens are formed from the condition tokens by attention weights alone, with no weight encoding the camera motion.

Misaligned case. A stored view at pose $W' \neq W$ enters with the fixed transform $P_i P_j^{-1}$, which acts on the pose-carrying channels. Placing its content into the target view requires the reprojection $\Pi_W \circ \Pi_{W'}^{-1}$, a resampling of token positions determined by per-pixel depth and the camera motion, so no assignment of attention weights that is independent of the scene realizes it; so the mapping must be learned in the weights, as for relative commands (Theorem 1).

Proposition 3 (adapter rank). *One rank-one update per head suffices for attention to reproduce the aligned condition; the backbone has 24 heads and the adapter has rank 32.*

Proof. Let each head h receive $\Delta W_q^h = u_h a_h^\top$ and $\Delta W_k^h = w_h b_h^\top$, where a_h, b_h read a feature that distinguishes condition tokens. Their contribution to the logit of an aligned condition token is $\gamma_h = (a_h^\top x)(b_h^\top x')\, u_h^\top w_h$, which can be scaled to dominate the backbone logits and the cross terms. As γ_h grows, a_{ij} concentrates on the aligned token and $o_i \to v_{j(i)}$, the target token constructed in Proposition 2. Stacking one rank-one block per head gives an update of the full projection of rank at most n_h. For this backbone $n_h = 24$ heads at width 3072, and the adapter has rank 32, so the capacity needed for the aligned-attention path in every head is available.

Scope. This is sufficiency of capacity, not a claim that training finds this solution, and completion of uncovered pixels may use the value and output updates as well.

Proposition 4 (no accumulation). *The bound does not grow with the number of returns.*

Proof. Keeping the first observation of each location (Lemma 1) makes every read of a written location carry the same single write/read error δ, and chunks that are not returns run the backbone unchanged (Lemma 2). Theorem 2 therefore applies at each return with the same constants, and neither the number of returns nor the spacing between them enters the bound.

A.6 Camera and Metric Invariances

Lemma 3 (camera gauge). *With all other inputs fixed, the common right action $W_k \mapsto W_k g$, $g \in \mathrm{SE}(3)$, leaves the camera conditioning unchanged. Jointly scaling scene depth and camera translation by a positive constant leaves image formation unchanged.*

Proof. The relative product becomes $P_i g (P_j g)^{-1} = P_i P_j^{-1}$, and for a scene point X and translation $\mathbf{b}$, homogeneous division cancels $\alpha > 0$ in $K(R\alpha X + \alpha\mathbf{b}) = \alpha K(RX + \mathbf{b})$.

Lemma 4 (pose-score invariance). *First-frame-relative scores remove a common rigid transform, translation-normalized scores remove positive translation scale, and Sim(3)-aligned scores remove both. Each is insensitive to errors consisting only of its removed transformations (Zheng et al., 2024; He et al., 2025a; Zhu et al., 2026; Umeyama, 1991).*

Proof. For camera-to-world poses $T_k = W_k^{-1}$, $(gT_0)^{-1}(gT_k) = T_0^{-1}T_k$. For nonzero translations $N(\mathbf{b})_k = \mathbf{b}_k / \max_j \|\mathbf{b}_j\|$, $N(\alpha\mathbf{b}) = N(\mathbf{b})$, and a common similarity is absorbed by the inverse change of fitted alignment variable.

Lemma 5 (rotation completion ratio invariance). *A common rigid motion leaves e and ρ unchanged, so both compare returns without alignment to ground truth.*

Proof. Every relative rotation used by E is unchanged under the common action, hence so are its angle, the numerator, and the nonzero denominator. The reference remains the model's own first visit, independent of the chosen world coordinate system.

A.7 Measured Constants

Measured return error tracks reproduction error across Table 1. Coverage is one for every return. Error correlates with reproduction error at 0.59, with render error at 0.02, and with jump angle at 0.21 (Figure 6). All returns above 40° have reproduction error at least 0.10, whereas successful returns in the same difficulty levels have median reproduction errors of 0.06–0.09. Appendix A.7 reports the constants and measurement procedure.

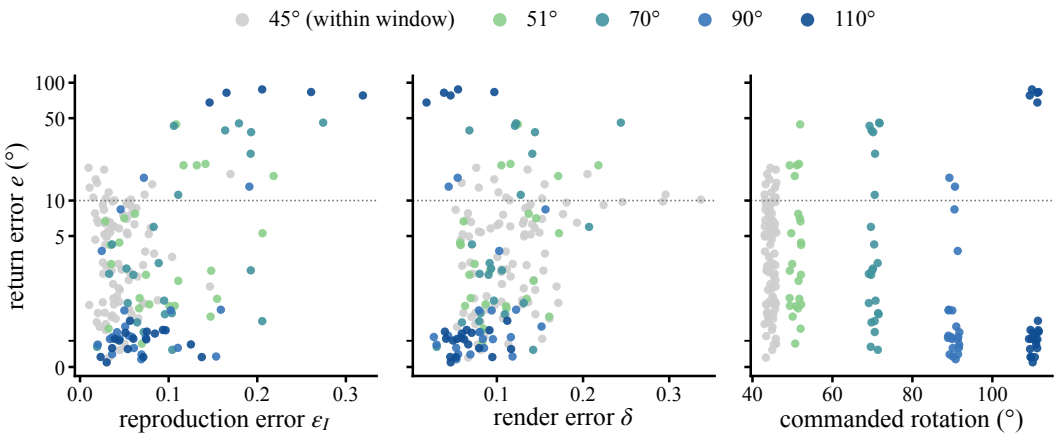

Figure 6: **Return error versus the terms of Theorem 2.** Returns are grouped by jump level, and target coverage is one throughout. Return error correlates most strongly with the reproduction error ϵ_I and only weakly with the render error δ or the commanded rotation.

The constants are measured on 182 returns across 91 panoramas. The canvas is reconstructed from each model's decoded frames preceding the return by keeping the first observation of each location, then rendered at the commanded return poses. Coverage is the rendered validity-mask mean. Reproduction error is covered-region mean absolute difference between output and render. The render error δ compares the render with the model's own first-visit frame at the target pose, which is the constant of A2. δ_{gt} compares the same render with the ground-truth panorama render and also contains the first visit's drift from the commanded sweep. VGGT supplies return error and the ground-truth estimator floor through the same pose pipeline (Wang et al., 2025a).

Table 4: **Measured terms of Theorem 2 by jump level.** n counts returns. Image errors are mean absolute differences of $[0, 1]$ intensities; δ is the render error against the model's own frames (A2), and δ_{gt} the error against ground-truth renders.

Level	n	κ	δ	δ_{gt}	ϵ_I (median)	ϵ_E ($^\circ$)
Within-window						
return 45°	91	1.000	0.120	0.177	0.038	0.30
51°	24	1.000	0.110	0.147	0.089	0.24
70°	23	1.000	0.109	0.105	0.096	0.18
90°	22	1.000	0.081	0.060	0.057	0.15
110°	22	1.000	0.064	0.030	0.075	0.10

Full coverage removes the completion term for every return in Table 4. Correlations of return error with reproduction error, render error against the model's own frames, ground-truth render discrepancy, and jump angle are 0.59, 0.02, -0.10, and 0.21, respectively. The measured relationship with reproduction error is threshold-like.

Large-error returns coincide with poor reproduction: all ten returns above 40° have reproduction error at least 0.10, mostly 0.15–0.32, with rotation completion ratios 0.23–0.49. Successful returns in the same difficulty levels have median reproduction errors 0.06–0.09, attributing the large-error group to failure to reproduce the supplied condition despite full coverage.

B Evaluation Protocols and Scoring

A command trajectory is $\mathcal{T} = (W_0, \ldots, W_{L-1})$, where W_k is the commanded world-to-camera pose at latent frame k and L is the number of latent frames. The attention window holds $N_w = 18$ latent frames. A return at latent m to target latent $j < m$ sets $W_m = \cdots = W_{m+h-1} = W_j$, a camera jump kept for h latent frames. For each return, the out-of-window fraction ϕ is the fraction of pixels in the target view whose scene direction was last in view at latent $\leq m - N_w$. It is computed from the commands alone before generation. The target age is $m - j$ latent frames.

Return rule. The return rule (Definition 2) selects chunks from camera commands alone. We use the poses W_k and chunk indices of Section 3.1, with indices starting at zero. Let $\mathbf{c}_k$ be the camera center obtained from W_k^{-1}. For chunk c, let $\bar{D} = \max_{k \leq 3c+2} \|\mathbf{c}_k - \mathbf{c}_0\|_2$ over the commands issued up to that chunk.

A chunk c is a return when a commanded pose in that chunk is near an earlier pose k, and an intervening command departed from k. For three-latent-frame chunks, we test some $q \in \{3c, 3c + 1, 3c + 2\}$ and $k \leq 3c - 6$. Near means rotational distance at most 10° and camera-center distance at most $0.15\bar{D}$. Departure means that some j with $k < j < 3c$ has rotational distance at least 25° or camera-center distance at least $0.4\bar{D}$ from k. When $\bar{D} \leq 10^{-3}$, as on rotation-only trajectories, the translation terms are dropped. The departure condition distinguishes a return from continued observation during a slow sweep.

Table 5 gives the resulting schedules. The rule is causal with respect to generated observations and identical for all variants because it reads only the supplied commands. Wan2.2-5B, CamCanvas w/o canvas, and CamCanvas share the same first visit before the first selected chunk. CamCanvas w/o canvas activates its adapter on the same schedule as CamCanvas. A trajectory that never satisfies the return condition of Definition 2 has no selected chunks.

Panorama jump protocol. The outbound sweep changes yaw by 2.2° per latent frame with a 4° sinusoidal pitch oscillation. Each trajectory has $L = 63$ latent frames, corresponding to 249 pixel frames, with one out-of-window return at $m = 54$ kept for $h = 3$ latents. Targets $j \in \{30, 21, 12, 3\}$ are assigned cyclically across clips with random turn sign. These targets give the four out-of-window jump levels in Table 6. Each clip also has a within-window return at $m = 60$ to $j = 36$, kept for $h = 3$ latents. The protocol uses 91 CC0 Poly Haven panoramas not used in training (Poly Haven, n.d.).

Each return is scored against that method's own first-visit frame at the target latent. Table 1 reports return error e and success S ($e < 10^\circ$) as defined in Section 3.2, with y_c the frame of the last latent in the returning chunk.

RealEstate10K jump protocol. Each trajectory has $L = 24$ latents, corresponding to 93 pixel frames. Latents 0–20 follow the real trajectory. A camera jump at $m = 21$, kept for $h = 3$ latents, returns to the pose of real frame 0 in half of the clips and of real frame 3 in the other half, visited 21 and 17 latents earlier, respectively. The protocol uses 300 clips of the RealEstate10K test set (Zhou et al., 2018). We take CamI2V's 1,000 RealEstate10K test clips (Zheng et al., 2024) in order and retain clips whose 16-frame camera path ends with displacement between 1.0 and 4.7 in RealEstate10K units and rotation between 5° and 40°. We keep the first 300 qualifying clips, whose identifiers appear in Appendix F. Table 2 reports the 150 returns to real frame 0. Return error e and success S ($e < 10^\circ$) follow Section 3.2, with the first-visit frame at latent 0 and y_c at latent 23; PSNR and LPIPS compare the 12 pixel frames of the returning chunk with real frame 0.

RealEstate10K out-and-back protocol. Each trajectory has $L = 42$ latents. Latents 0–20 follow the real trajectory, and latents 21–41 revisit the same poses in reverse, with $W_k = W_{41-k}$ for $21 \le k \le 41$. This protocol uses the same 300 clips. Each return frame is paired with the outbound frame at the same commanded pose; $\bar{e}$ is the mean orientation error over these pairs, and MEt3R measures their content consistency (Asim et al., 2025). Six return frames coincide with the poses of real frames 0, 3, 6, 9, 12 and 15, and PSNR and LPIPS compare them with those real frames. Translation uses a depth canvas with MoGe depth (Wang et al., 2025b).

Panorama cycles. Each trajectory has $L = 63$ latent frames, corresponding to 249 pixel frames. Outbound latents 0–31 are followed by return latents 32–62 along the outbound path in reverse. The same 91 panoramas each supply one rotation cycle and one translation cycle. Each return frame is paired with the outbound frame at the same commanded pose, following the pairing used by HY-WorldPlay and VMem (Sun et al., 2025; Li et al., 2025b). Rotation uses the panoramic canvas, and translation uses the depth canvas. On the return leg, Table 10 reports PSNR, LPIPS and MEt3R

Table 5: **Return schedules.** All schedules follow from Definition 2 and are shared by all variants.

Protocol	Returning chunks (zero-based)
Panorama jump return, including the within-window return	18, 19, 20
RealEstate10K jump return	7
Panorama rotation cycle	15–20
Panorama translation cycle	14–20

Table 6: **Panorama jump levels.** Target and age are given in latent frames. Each clip has one out-of-window return at latent 54 and one within-window return at latent 60; the within-window row covers the same 91 clips.

Level	Target latent	Age	Jump (°)	ϕ	Clips
51°	30	24	50.8	0.23	24
70°	21	33	70.4	0.45	23
90°	12	42	90.4	0.60	22
110°	3	51	110.1	0.77	22
Within-window return 45°	36	24	44.5	0.02	91

(Asim et al., 2025) between paired frames and the mean orientation error $\bar{e}$ between their estimated poses.

Pose estimation and estimator floor. VGGT-1B estimates the camera at each latent frame from the generated video, with frame zero as the reference (Wang et al., 2025a). The evaluation uses the first pixel frame associated with each latent frame. We apply the same estimator pipeline to ground-truth panorama renders. These renders achieve 100% return success, with median return error of 0.1–0.2° across the difficulty levels. This is the estimator floor.

Rotation completion ratio and return success. The rotation completion ratio uses each model's own first visit as its reference. Using the estimator E and angle $\angle$ from Section 3.2, define

$$\theta_{\text{need}} = \angle E(y_{\text{pre}}, y_j), \qquad \rho = \frac{\angle E(y_{\text{pre}}, y_c)}{\theta_{\text{need}}}, \qquad e = \angle E(y_j, y_c).$$

Here y_j is the first-visit frame, y_{pre} the pre-jump frame, and y_c the frame of the last latent in the returning chunk. The denominator is nonzero and measures the rotation needed to return to the model's own first-visit orientation, rather than the commanded jump angle. No alignment to ground truth enters these quantities. The main tables in Section 4 report success S, the fraction with $e < 10^\circ$, rather than ρ, because S also detects overshoot, undershoot, and rotation about the wrong axis.

C Training and Implementation

Backbone and adapter. The backbone is a block-causal Wan2.2-5B distilled with Self Forcing from a bidirectional camera-conditioned model (Yin et al., 2025; Huang et al., 2025c; Yin et al., 2024; Huang et al., 2026). CamCanvas adapts the distilled generator based on Wan2.2-5B (Huang et al., 2026). It generates three latent frames per chunk with four denoising steps. Output images are 864×480 pixels, with horizontal and vertical fields of view of 89° and 58°. Rank-32 LoRA layers (Hu et al., 2022) are attached to the query, key, value, and output projections in all 30 self-attention blocks, totaling 23.6M trainable parameters. The backbone, cross-attention, and feed-forward layers are frozen. For a projection W, the adapter adds $(\alpha/r)BA$ with $r = \alpha = 32$. Matrix A uses Kaiming initialization (He et al., 2015), and B starts at zero.

Conditioning and objective. Training uses teacher-forced flow matching on 21-latent-frame clips. A diffusion timestep is sampled for each three-frame chunk. Clean rendered conditions are interleaved as [3 noisy | 3 condition], producing 42 latent frames. Condition timesteps are zero, and each condition has the same pose as its corresponding target. The video VAE encodes conditions in both training and inference. The scheduler-weighted masked mean squared error supervises flow velocity on noisy latent frames only. Condition frames and the supplied clean first frame are excluded from the loss. First-frame conditioning and the complete rendered condition each have dropout probability 0.1.

Table 7: **Adapter optimization settings.**

Training setting	Value
Optimizer	AdamW (Loshchilov & Hutter, 2019)
Learning rate	10^{-4}
Betas	$(0.9, 0.95)$
Weight decay	0.01
Warm-up	100 steps
Learning-rate schedule	Cosine with warm-up
Gradient norm clipping	1
Batch size	1 clip
Precision	bfloat16
Training duration	3000 steps

Training data. Rotation training uses approximately 900 CC0 Poly Haven panoramas at 2048×4096 pixels (height by width) (Poly Haven, n.d.). Trajectory families are smooth sweeps, random

walks, jumps, and smooth sweeps with jumps. Training jump magnitudes are sampled uniformly from 10° to 45°. For each target, the condition retains only content observed in the first frame or in earlier chunks, rendered at that target's command pose. Translation samples come from TartanAir and Sekai (Wang et al., 2020; Li et al., 2025c), with depth supplied by MoGe (Wang et al., 2025b). The mixed rotation and translation training set contains 3275 clips. Table 8 gives the sampled family proportions.

Table 8: **Training mix.** Trajectory families in the mixed training set of 3275 clips; shares are rounded and therefore sum to 101%.

Trajectory family	Sampled share
Translation	33%
Random walk	20%
Jump	17%
Smooth sweep	16%
Smooth sweep with jumps	15%

Canvas and inference. The inference canvas is 8192×16384 pixels, in height-by-width order. It is initialized from the given first frame and updated from decoded generated frames. We keep the first observation of each location, as in equation 3. A read renders the canvas at the commanded pose. Uncovered pixels are zero with an associated coverage mask. On chunks the command sequence does not mark as returns, adapter scale is 0, condition tokens are absent, and the attention window is 18 latent frames. On returning chunks marked by the command sequence, CamCanvas uses adapter scale 1, interleaved conditions, and an attention window of 42 latent frames, which covers the same span of generated frames as the window of 18 together with their interleaved condition frames. CamCanvas w/o canvas uses its adapter at the same returning chunks without condition tokens and retains window 18.

Adaptation to SANA-WM. As for Wan2.2-5B, the condition is the canvas rendered at the commanded pose and encoded by the backbone's VAE (Zhu et al., 2026). SANA-WM uses linear attention and an LTX-2 VAE (HaCohen et al., 2026) with eightfold temporal compression. The adaptation uses 128 latent content channels and an additional coverage channel, injected through a zero-initialized projection added to image tokens, together with rank-32 adapters. The backbone is the chunk-causal SANA-WM checkpoint at 720p with 1.6B parameters. The adapters are trained for 1500 steps at learning rate 10^{-4} on 16-frame clips with the backbone frozen. At inference, the rendered condition is supplied and the adapters are active only on returning chunks marked by the command sequence, as in Section 3.4; every other chunk runs the backbone unchanged.

D Additional Results

This appendix collects results summarized in Section 4.

Table 9: **Return test on four backbones from three families** (Huang et al., 2026; Gao et al., 2026; Zhu et al., 2026). S: success (%, $e < 10^\circ$) against the model's own first visit; e: median return error. Ground-truth renders all succeed (median 0.1–0.2°, same estimator). The last row uses a rotation cycle with a smooth return instead of a jump.

Backbone	Return	S (%)	e (°)
Wan2.2-5B (block-causal)	panorama, within-window return 45°	1	54
	panorama, level 51°	4	60
	panorama, level 70°	0	70
	panorama, level 90°	0	70
	panorama, level 110°	5	54
	real video, return to the first frame	17	21.5
MiniMax-H3-33B (block-causal)	panorama, within-window return 45°	32	19.0
	panorama, level 51°	8	24.1
	panorama, level 70°	35	18.9
	panorama, level 90°	82	3.5
	panorama, level 110°	91	2.0
LingBot-World (Wan2.1-1.3B)	panorama, within-window return 45°	2	65
	panorama, level 51°	0	44
	panorama, level 70°	0	63
	panorama, level 90°	0	74
	panorama, level 110°	9	58
SANA-WM	rotation cycle with smooth return, 91 trajectories	75	4.6

Table 10: **Rotation and translation cycles on 91 panoramas.** $\bar{e}$: mean return error (°) against paired first-visit orientations. PSNR (dB), LPIPS, and MEt3R compare paired return and first-visit frames. Best results are in bold.

	Rotation cycle				Translation cycle			
Method	PSNR↑	LPIPS↓	MEt3R↓	$\bar{e}$ ↓	PSNR↑	LPIPS↓	MEt3R↓	$\bar{e}$ ↓
Wan2.2-5B	15.1	0.408	0.225	6.6	15.8	0.463	0.320	7.9
HY-WorldPlay	11.2	0.568	0.265	27.5	14.4	0.467	0.296	4.2
LingBot-World	**23.4**	**0.164**	**0.070**	22.1	17.2	0.394	0.312	13.9
CamCanvas w/o canvas	15.4	0.416	0.224	6.6	16.1	0.477	0.335	8.9
CamCanvas	18.9	0.257	0.130	**3.2**	**19.0**	**0.361**	**0.245**	**3.0**

Table 11: **SANA-WM transfer on the rotation cycle.** Metrics as in Table 10. Best results are in bold.

	Rotation cycle, 91 panoramas			
Method	PSNR↑	LPIPS↓	MEt3R↓	$\bar{e}$ ↓
SANA-WM	15.2	0.434	0.269	7.5
SANA-WM + CamCanvas	**19.4**	**0.306**	**0.181**	**3.3**

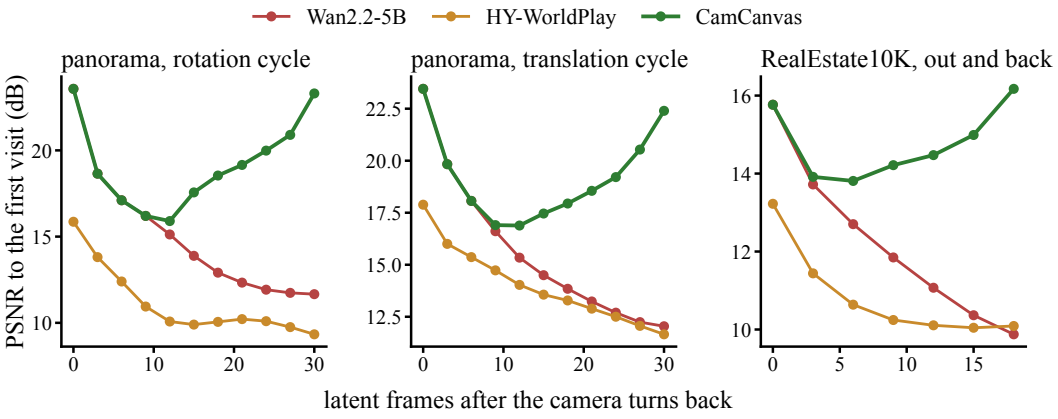

Figure 7: **Content consistency along returns.** PSNR against the paired first-visit frames for Wan2.2-5B, HY-WorldPlay, and CamCanvas on the panorama rotation cycle, the panorama translation cycle, and RealEstate10K out-and-back trajectories. CamCanvas recovers the first-visit content as the camera comes back, while the baselines keep drifting.

E SOURCES FOR CAMERA JUMPS

Camera jumps test whether a requested view can be reached directly. Table 12 collects primary sources. They document view retargeting in natural behavior, interactive interfaces, visual media, and embodied tasks. Section 4.1 uses this evidence to motivate the jump command in the return test.

Table 12: **Sources for camera jumps.** Settings in which the view is redirected to a chosen target.

Setting	Reference	Relevance
Gaze follows task-relevant targets	Hayhoe & Ballard (2005)	Fixations are timed to task demands and move between relevant locations.
Natural retargeting can be large and rapid	Land (2004)	Coordinated eye, head, and trunk rotations rapidly redirect gaze in everyday tasks.
VR exploration involves view retargeting	Sitzmann et al. (2018)	Head orientation, gaze, and kinematic constraints jointly determine viewing behavior.
Discrete viewpoint changes are an interaction design	Farmani & Teather (2018)	Viewpoint snapping reduces cybersickness while preserving presence and task performance.
Cuts are a standard visual transition	Cutting et al. (2010)	Abrupt transitions dominate contemporary film editing.
Navigation can specify a destination	Savva et al. (2019)	PointGoal navigation provides goal coordinates and evaluates whether the agent reaches the goal.
Control matters for embodied utility	Zhang et al. (2026)	Closed-loop world-model evaluation separates visual quality from task success and controllability.
World models can be interactive environments	Bruce et al. (2024)	Genie supports actions at each frame in a learned generative environment.
Generation can accept a requested camera trajectory	GEN3C (Ren et al., 2025); ReCamMaster (Bai et al., 2025)	GEN3C renders a geometric cache along a user-specified trajectory; ReCamMaster renders video from new viewpoints.

F EVALUATION DATA IN REALESTATE10K

The IDs of the 300 RealEstate10K test clips used in Table 2 are listed below in the order they were selected (Appendix B); each identifier is the name of the clip's camera file in the RealEstate10K test split (Zhou et al., 2018).

‘dcdc007caf2285af’, ‘60b21485e6a87336’, ‘bb0d9ff59505b17e’, ‘b8a77ae2419b398f’,
‘802982739759b171’, ‘0f12b97e0e4c7e21’, ‘d70fc3bef87bffc1’, ‘39c662fa32a3b5c6’,
‘99556a626666c2e5’, ‘8127997a38df9d32’, ‘f9e4896193305279’, ‘7a2ad72c32c415bd’,
‘735cfe5b4fcadb57’, ‘5019e16d5e41ff8d’, ‘22666111b2180af9’, ‘867e133ab7e43326’,
‘63601178de25dcf3’, ‘4e6334d5ae0c157a’, ‘646a8d75b0253ec0’, ‘1402fce28722610b’,
‘34b35070a9e24f32’, ‘ec3b36ce3794c18a’, ‘565ff739ca86758e’, ‘3b9e04113b202116’,
‘68a17cc5d6e41d9c’, ‘6ecda7688473aa25’, ‘77917a56e2650f83’, ‘0542630de1d734de’,
‘e67c43bee6f33ba0’, ‘403951b5d632b5ab’, ‘fa53b8c13f6caabb’, ‘1fcfd8a36e171639’,
‘9dfc6db5375952c3’, ‘633766029a6f910a’, ‘60579e8876b01b2f’, ‘25f590b3de5b28e8’,
‘1d4d81f8629b119e’, ‘a25c24839b6d427c’, ‘af4b71cad42eae90’, ‘da7649f08f0ecb90’,
‘4ce58504b055463e’, ‘8f06df2fca9350ba’, ‘b0882e9423cd8a12’, ‘9bd7c28ddb5bcc58’,
‘f28ef96bc6478898’, ‘da27974057f07122’, ‘0b530eea368f626e’, ‘15a138312ad94718’,
‘0b55abc1ca2fe909’, ‘2d60837ef2e52abd’, ‘c4a95e304041e85a’, ‘d942e48c948b3546’,
‘377769942e6a748d’, ‘d76607d37608e7ce’, ‘a79b3300f99683bc’, ‘395caf9235fde098’,
‘96d5582decc8c3cf’, ‘fb08b3c2668c99c3’, ‘f8f9ae6839b997d4’, ‘49235d402cbb8895’,
‘bf335dfd4f3e4fc8’, ‘d72cdf31bb064b1c’, ‘49d5b942442449b4’, ‘a03594dbdf6e6992’,
‘ad543396707ee644’, ‘fe6690ee4f6c7244’, ‘f4769a32a225c172’, ‘221205ddf59c5156’,
‘b156889fde2dbb85’, ‘739b73b8848f72cb’, ‘7a4d944a9f4cfed8’, ‘61192009ce2e8d25’,
‘6e2921154309ebf3’, ‘dc7c7a8a4d5ac4e0’, ‘e885583f5cc4bce7’, ‘27f3ccfb3199499e’,
‘632a121cf77513a2’, ‘9cf5050816fcefaa’, ‘997cf2dce8841462’, ‘173a82eeea56aed3’,

‘47a76ca10546fe8f’, ‘d91c20bd6f99aca4’, ‘4047ea5cbe353c08’, ‘63840de6e5746741’,
‘75e56e8c9600ae26’, ‘c161bcf6082b40af’, ‘e29b483e3b210c75’, ‘ce69e6f9387f37b3’,
‘df242c69a2941d5e’, ‘2ff40df261e17697’, ‘d63597aeceeebf11’, ‘c040e7598e43eef1’,
‘74598c6dc1d84d8e’, ‘cf845955d579f84c’, ‘8d360bd4931a8f83’, ‘9e013fd3bd563b6b’,
‘6e77ac6af5163f5b’, ‘929ed71c97317c55’, ‘a38c00f474b0df82’, ‘385f9444d20eb160’,
‘f3c66c5552e28f20’, ‘f91749843a42fa3e’, ‘1515d37824dd6b22’, ‘195ca8350ff27f6f’,
‘01fa6190cd47d125’, ‘ead2bb83c1955f14’, ‘e9f5da5f5d7c0de4’, ‘f1226d5483f1c1cd’,
‘a18f5d8a3c80ddc3’, ‘d29aaf20f3457667’, ‘5fada958027ac4a6’, ‘50543c3f94b1ea8d’,
‘28f5ebf3c3e2fe54’, ‘d5c7972d2fc031d3’, ‘fc508faa91c9807d’, ‘f82a67e46c5e1434’,
‘e071ef79c9a7d612’, ‘e54631e5313e78d7’, ‘5a832370d2d05215’, ‘c4688d5ce226a8a3’,
‘128bf83073de3ba1’, ‘4a7e531e1a35d424’, ‘b6d4a0650cc6452e’, ‘7a18ec7dc0bdf907’,
‘59350c4847ade4c7’, ‘f591e1921f2c850c’, ‘c192371973923223’, ‘84ab392d682f296b’,
‘10ad4fc499c48b38’, ‘d438687082422f1e’, ‘8e143c65e5b4d26a’, ‘c623c6a79373330b’,
‘8408cca2093ea869’, ‘ce6c6d81af6b62cd’, ‘2e64a2d17f9a76f7’, ‘316035dd285c5e27’,
‘07ba5489b56b7d62’, ‘c8c2d887b38f6dc5’, ‘07225d96742d2a6c’, ‘2b1da1fbe7f18f7e’,
‘20f8c6738e22e764’, ‘0b1e61c69c98026b’, ‘17d841670d2da942’, ‘1de1b73fe4d6aa77’,
‘ecc06f0818f48464’, ‘5c549af41b41e826’, ‘a1a21aee1df9fa71’, ‘7d9958c362abfcb2’,
‘f2d0e69416df7741’, ‘d2f2da7454cc7d9f’, ‘9f39cb8fb654b30e’, ‘375f9c448cf31ccb’,
‘34766105ea9d7a2b’, ‘1840ae9e2494443e’, ‘7869d609f86fe0da’, ‘f01e8b6f8e10fdd9’,
‘c0f6817b3e55c661’, ‘81b0fd734d794027’, ‘8c8ed2adba3dad7a’, ‘5e275e7af840e344’,
‘6c42b1d8b83548c1’, ‘a0948de3902ae476’, ‘cebfe607ccf4b943’, ‘b6b1244092471403’,
‘02c0c9192fb9a6a1’, ‘a6ca747ffac74a34’, ‘a8cc9d1e56928758’, ‘6f61f1648c6f6c58’,
‘0fbe6d76015f75d4’, ‘be4ecfc1baabdf63’, ‘f478d5f55e41f4f7’, ‘c34b59ec59eb886c’,
‘c43892f312b3c85b’, ‘a705032c068c7caf’, ‘a815c2f38988bd21’, ‘dd9ab2885314a176’,
‘e9b506f37c2389cb’, ‘be5146d052ba89e9’, ‘2d6f9fa00dcee664’, ‘dbdeb6adf84634d4’,
‘dd5142700ac59a94’, ‘5dbf866479511338’, ‘c2adadff1502b19d’, ‘6b658b1d8562dc85’,
‘d9379ef7eb90e7e7’, ‘41f3291d82fc4d93’, ‘e1242403844f68dd’, ‘d5395d343ad4592b’,
‘129b5e9b80cc4b4b’, ‘3fc266558ec5c07b’, ‘4e2bb6d3056df6d8’, ‘763a4369298f49e2’,
‘b70c3d4dcc51f866’, ‘4054d32655ba5eda’, ‘a6588e3953a00b5d’, ‘b3fce22970354e15’,
‘51f96756456b33fe’, ‘6d764e8551754631’, ‘5be4f1f46b408d68’, ‘5590c5929892c362’,
‘e73ee867c3834f3d’, ‘5cf16d250189b7fe’, ‘b00a2bc609d0ac6e’, ‘83a593a5f24ab22a’,
‘04ec725465dc5329’, ‘a02db7e8cf77fea7’, ‘5f4ad1c4a3979f1a’, ‘ad69d907064db1f7’,
‘f7d372dde37d82a8’, ‘a75565ce57aae546’, ‘aa604fe5b2e59274’, ‘d4f002923ad8463f’,
‘464d97e527dd5f8a’, ‘eac13f3b1145593c’, ‘689bfbf638f18a83’, ‘4bc47dc7f8781812’,
‘f815dfc42d162fea’, ‘a4cc81ce7d4168bb’, ‘db33f36374e40bfd’, ‘5581754c07e626bf’,
‘6d6be1376c59a3db’, ‘0aa8646901d156e4’, ‘2d524e9324228d6e’, ‘f5b151019f9010a5’,
‘ade543e5f3a3d851’, ‘134d7e5a74497a82’, ‘4b7e7de9132f4149’, ‘95d7f4146108b286’,
‘a0bc9b509471e1dc’, ‘f289b5f7136bba85’, ‘89371ab4cdf5e682’, ‘db2a88aeac0528ef’,
‘069597e1fe899530’, ‘004e9db3337e8206’, ‘9e8dd4972a31efa3’, ‘c67003dad6fde3b3’,
‘e423e3493da1928e’, ‘2236eab1c5c86fc4’, ‘7b6a13ed373bafb1’, ‘31ee8cabc96b9a62’,
‘e8530fdec17b42eb’, ‘e4ce9925774dac97’, ‘ab55215118ee9fe9’, ‘617580a94b6fd65a’,
‘1eca36ec55b88fe4’, ‘c9753d2d1f66ab43’, ‘fa95ce310f223e51’, ‘f1c659ac9ea21f4f’,
‘868a0505d840d013’, ‘b33cff827e646ae0’, ‘64aab07da1c81c11’, ‘65a9c0856de40a68’,
‘0555b07fe6239b4a’, ‘bb6d106fb22b1f54’, ‘1a0f8c49b336ec38’, ‘e050b07b90580dba’,
‘d87b40f255fe1b0f’, ‘b4099665590548fc’, ‘3a6dc09185951ae3’, ‘01e18dbbf22ff263’,
‘4e96d43605f27475’, ‘c6e9e5366bec252a’, ‘fd020d75ab5e2fa3’, ‘688ea661582358cf’,
‘7567387078585ec3’, ‘946babe2aed33b04’, ‘c441a34ec98346e9’, ‘ecc2b1b9cb1fa9ad’,
‘36fb4d41b00581c4’, ‘abd304cbb9bd30ba’, ‘bc9cf914440f8fd9’, ‘eb5b73ca4a18bbf9’,
‘d2df35be33a3887f’, ‘acdcd593a8f50eb1’, ‘e9b3ce3fb857d86f’, ‘43759ced44693671’,
‘a0d174b988c5ead7’, ‘7a2291bc1b9a6e7d’, ‘36977643258aa392’, ‘ea665164c6291b91’,
‘0757b4bc82bf26b1’, ‘a011148adfba8ab2’, ‘a0dda4b2ec17cde1’, ‘d57f52c12e2d2189’,
‘c5f39167be4d1ec2’, ‘11bf4b38f88bfe9b’, ‘df78ec87fe403e80’, ‘650503eb1bc58d0d’,
‘5805ee451b7917cd’, ‘8765f603d39777c2’, ‘5a55518f5e8c384d’,
‘16cd4f1cb2a467c1’, ‘6a4a52e3c16e0f4a’, ‘ae4090e8832b90f9’,
‘ecf8f945dc42ec62’, ‘4a7f0556fb58a5cc’, ‘4ba7caa04cea37b7’,
‘1af55d4494e499b1’, ‘203b7543bb3387ce’, ‘a6c6cbe8f6e0787f’,