WORLD-T^3: TEST-TIME TRAINING AS PERSISTENT MEMORY FOR REAL-TIME VIDEO WORLD MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent advances in video world models have enabled plausible interactive environments, yet long-horizon generation places a fundamental demand on visual memory: the previously observed world must remain consistent as the model continues to generate. Existing autoregressive approaches often implicitly rely on the generative model itself to retain such information, making persistent scene memory difficult to maintain beyond the active context. We introduce World-T^3, a camera-controlled video world model with a test-time training memory that consolidates past observations into dedicated model parameters. To robustly update this memory, we propose a recursive least-squares rule that incrementally absorbs novel visual features via online pose-keyed memory updates. Therefore, our TTT memory enables consistent generation by reading out previously observed scene features via camera pose from the fast-weight state, while maintaining constant storage and per-step computation regardless of rollout length. To facilitate real-time interactive generation with this robust memory, we further distill the autoregressive model into a few-step generator via dual-teacher distribution matching. Experiments show that World-T^3 achieves precise camera control and faithful scene revisits while sustaining 20.6 FPS at 1280×704 on a single NVIDIA B200 GPU.

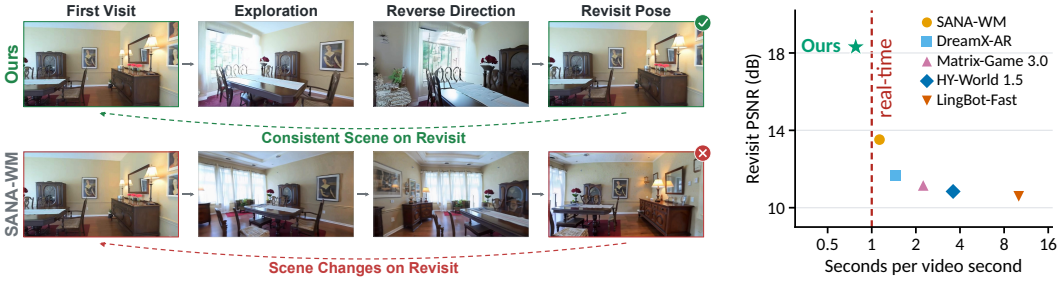

Figure 1: **Left:** Revisit consistency. When returning to a previously visited viewpoint, our World-T^3 enables faithful scene recall, whereas existing autoregressive baselines forget earlier observations beyond the context window and resynthesize new content. **Right:** Speed vs. quality on one NVIDIA B200 GPU. x-axis: wall-clock seconds required per second of generated 16 FPS 1280×704 video (points left of the dashed line are real-time). y-axis: PSNR on the long-horizon revisit benchmark.

1 INTRODUCTION

Video world models aim to simulate how a world evolves in response to user control, emerging as a powerful interface for embodied simulation, playable environment generation, and virtual exploration (Yang et al., 2023; Wang et al., 2026c; Bruce et al., 2024; Che et al., 2025; Zhang et al., 2025; Feng et al., 2026a; Shen et al., 2026). Conditioned on a reference image and a scene description, recent work utilizes autoregressive video diffusion models (VDMs) to allow users to explore and interact with generated environments through sequential control inputs, such as 6-DoF camera poses and natural language instructions (Team et al., 2026c;b; Zhu et al., 2026a; Team et al., 2026a; Xu et al.,

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.

2026a). However, as these models move from short-horizon generation toward sustained interaction, generating a plausible next observation is no longer sufficient. The autoregressive VDMs tend to remain locally coherent while gradually losing earlier scene information once it falls outside the active history context, causing previously established content to be resynthesized from the generative prior rather than preserved as persistent world state.

Existing approaches attempt to preserve scene information beyond the local generation window by maintaining explicit 3D scene representations (Huang et al., 2025; Zhao et al., 2026a; Wang et al., 2026b), retrieving historical frames via camera and geometric cues (Li et al., 2025; Yu et al., 2025; Xiao et al., 2026; Xu et al., 2026b; Wang et al., 2026d), or compressing past features into compact memory tokens (Kim et al., 2026; Bian et al., 2026; Wei et al., 2026). While these methods improve long-horizon consistency, maintaining an expanding collection of reconstructed geometry, historical frames, or tokens often introduces unbounded storage overhead and escalating attention complexity as rollouts progress. This motivates a complementary question: can a video world model consolidate successive observations into an online-updated, parametric memory that enables recovery of previous scene content, while keeping memory storage and per-step access costs independent of rollout length?

To this end, we propose World-T^3, a camera-controlled video world model that consolidates observation history into a fixed-size, pose-indexed parametric memory via test-time training (TTT). As generated video blocks leave the local context, their pose-content associations are written into dedicated fast-weight memory, which is subsequently queried by incoming camera poses to recover features of previously observed regions. A critical challenge in online parametric updating is handling smooth camera trajectories, where highly correlated pose features naturally introduce spatial redundancy across consecutive frames. To ensure that each update selectively incorporates novel visual content without degrading previously stored information, we propose Test-Time Memory Consolidation (§3.2.2), an online recursive least-squares update rule that incorporates the accumulated covariance of pose keys into each write. By solving a regularized regression objective over past pose-content associations, our method preconditions incoming key features to isolate non-redundant spatial information. Our TTT memory storage and per-step update complexity remain independent of rollout length, allowing history to be sequentially consolidated and accessed within a fixed budget.

Specifically, starting from a pretrained video diffusion model (Wan et al., 2025), we develop World-T^3 through a progressive three-stage training pipeline that establishes camera control, learns persistent scene memory, and enables few-step autoregressive generation (§3). Training draws on a curated corpus of 908K camera-annotated clips from real-world footage and synthetic environments (§B). In Stage 1, we adapt the pretrained backbone into a bidirectional teacher conditioned on 6-DoF camera trajectories. In Stage 2, we then convert it into a block-causal autoregressive model with bounded visual context and jointly train the backbone and memory module to consolidate past observations and retrieve them by camera pose. In Stage 3, to achieve real-time interactive generation, we design a dual-teacher distribution matching distillation framework (Song et al., 2023; Yin et al., 2024; Zhao et al., 2026b; Yin et al., 2025) that compresses the multi-step autoregressive model into a 4-step generator, leveraging both the bidirectional and memory-augmented teachers to balance local visual quality and long-horizon memory consistency. At deployment, our World-T^3 sustains 20.6 FPS generation at 1280×704 on a single NVIDIA B200 GPU with live online memory updates.

Our main contributions are: (i) We introduce World-T^3, a camera-controlled video world model that incorporates a pose-indexed test-time training memory. Through an online recursive least-squares update rule, our model robustly consolidates historical observations into a parametric memory without escalating storage or per-step computation complexity. (ii) We design a progressive three-stage training framework that converts a pretrained video diffusion backbone into an autoregressive world model with TTT memory, and further compresses it via dual-teacher distribution matching distillation into a few-step generator. (iii) Experiments demonstrate that World-T^3 achieves superior long-horizon revisit consistency and real-time inference speed compared to mainstream baselines.

2 Related Work

Video World Models. Recent advances in video generation have given rise to interactive world models that condition future observations on camera trajectories or action sequences (Team et al., 2026c;b; Zhu et al., 2026a; Team et al., 2026a; Xu et al., 2026a; Bruce et al., 2024; Che et al., 2025; Zhang et al., 2025; Yang et al., 2023). To incorporate camera control, various representations have

been explored, including pose vectors, dense Plücker ray maps, and relative positional encodings that integrate the geometric signal directly into the attention mechanism (He et al., 2024; Wang et al., 2024; Yin et al., 2026; Zhang et al., 2026a). While these approaches produce visually compelling results over short horizons, they suffer from progressive camera drift and temporal incoherence as the generation window extends (Xu et al., 2026a; Chen et al., 2026a; Ye et al., 2026). The fundamental issue is persistence: a revisited region should faithfully preserve the state it exhibited on first encounter, rather than being hallucinated anew from the model's learned prior (Lian et al., 2025; Lu et al., 2026). A parallel line of studies tackles this desideratum, whether by reconstructing and reprojecting previously observed geometry (Huang et al., 2025; Zhao et al., 2026a; Wang et al., 2026b), retrieving earlier frames according to spatial coverage (Li et al., 2025; Yu et al., 2025; Xiao et al., 2026; Xu et al., 2026b; Wang et al., 2026d), or distilling the ever-growing observation history into a fixed-size token buffer (Kim et al., 2026; Bian et al., 2026; Wei et al., 2026). These strategies, however, face an inherent dilemma: structured memory grows in cost with rollout length, while fixed-capacity alternatives inevitably dilute earlier observations as new content is compressed. In contrast, World-T^3 absorbs the growing history into the network weights via test-time training, incurring constant memory cost regardless of sequence length while preserving previously encountered states within the expressive capacity of the parameter space.

Test-Time Training as Memory. Test-time training casts persistent memory as an online associative-memory problem that continuously absorbs new observations into a fixed-size parametric state (Behrouz et al., 2026). Successive works have progressively coarsened the write granularity from a single gradient step per token (Sun et al., 2024), to objectives amortised over windows (Behrouz et al., 2025), to chunk-level updates sized to saturate modern accelerators (Zhang et al., 2026b). The same principle has since migrated to visual and embodied domains: video diffusion models adopt it for minute-long coherent generation (Dalal et al., 2025); 3D reconstruction systems recast recurrent updating as online learning over image streams (Chen et al., 2026b); and spatial understanding and robot control leverage it to maintain persistent state over long streams or to distill reusable memory from unlabelled demonstrations (Liu et al., 2026; Jiang et al., 2026; Feng et al., 2026b). Throughout this progression, a shared assumption remains: the gradient-based write presumes successive keys are diverse enough for a local step to meaningfully separate them. When keys are nearly collinear along camera poses in a smooth trajectory, each update largely reiterates what the store already contains. World-T^3 addresses this redundancy by solving the minimiser of the accumulated objective, providing video world models with a memory faithfully preserving scene state across long camera trajectories.

3 WORLD-T^3

World-T^3 is built on three successive stages from a pretrained text-image-conditioned video diffusion model (VDM). (i) *Bidirectional SFT* endows VDM with camera control and adapts it to the domain of freely navigated scenes using bidirectional attention (§3.1). (ii) *Autoregressive Training* introduces block-causal attention with bounded temporal context and persistent test-time-trained (TTT) world memory, updated online as blocks leave the context and dynamically queried by camera pose (§3.2). (iii) *Few-Step Distillation* reduces denoising sampling to four steps through consistency and self forcing distillation, guided jointly by both the bidirectional and autoregressive teacher models (§3.3).

3.1 STAGE 1: BIDIRECTIONAL CAMERA-CONDITIONED TEACHER

To establish a strong foundation capable of producing globally coherent geometry and motion, Stage 1 fine-tunes all parameters of the pretrained VDM backbone alongside a newly introduced camera conditioning module. By attending over the entire temporal horizon simultaneously, the teacher constructs each latent frame with global awareness of all others, thereby realizing a given trajectory as a single coherent motion. This bidirectional teacher model serves two pivotal roles in subsequent stages: it provides weight initialization for the autoregressive student in Stage 2

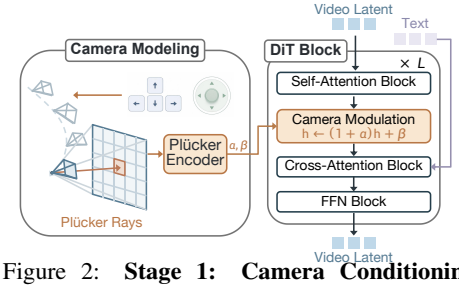

Figure 2: **Stage 1: Camera Conditioning.** Plücker raymaps modulate each bidirectional DiT block through camera-dependent scale and shift.

(§3.2), and supplies the reference score function for the distribution matching objective in Stage 3 (§3.3).

We encode camera geometry using Plücker raymaps (He et al., 2024; Yin et al., 2026), with all poses expressed relative to the first frame. For a latent frame f with rotation $\mathbf{R}_f$, camera origin $\mathbf{o}_f$, and intrinsics $\mathbf{K}$, the ray at spatial position (u, v) has direction $\mathbf{d}_{f,uv} \propto \mathbf{R}_f \mathbf{K}^{-1}[u, v, 1]^\top$ and is represented by its direction and moment: $\mathbf{r}_{f,uv} = \big(\mathbf{d}_{f,uv},\ \mathbf{o}_f \times \mathbf{d}_{f,uv}\big) \in \mathbb{R}^6$.

These rays form a six-channel field $\mathbf{\Pi}$ aligned with the latent grid, providing spatially resolved camera conditioning. Identical viewpoints yield identical raymaps, which also serve as the basis for pose-keyed retrieval in our TTT memory module (§3.2.2). An embedding layer projects $\mathbf{\Pi}$ into the VDM backbone's feature space and injects the features into every transformer block through adaptive layer-norm modulation, with its output projection initialized to zero for training stability.

3.2 Stage 2: Autoregressive Teacher with TTT Memory

The first stage endows the bidirectional backbone with camera control over fixed-length sequences. In Stage 2, we convert this model into a block-causal VDM capable of retaining long-horizon scene consistency via persistent test-time training (TTT) memory.

Generation produces a sequence of N latent blocks $\mathbf{x}_{1:N} = (\mathbf{x}_1, \ldots, \mathbf{x}_N)$, each containing B latent frames. The rollout is conditioned on a text prompt c, a reference image c_I, and a 6-DoF camera trajectory $\mathcal{T}_{1:N}$. To keep per-block computation $O(1)$ *w.r.t.* rollout length, we construct a bounded visual context C_b and an evolving TTT memory state $\mathcal{M}_b$. The joint distribution factorizes block-causally as:

$$p_\theta(\mathbf{x}_{1:N} \mid \mathcal{T}_{1:N}, c_I, c) = \prod_{b=1}^{N} p_\theta(\mathbf{x}_b \mid C_b, \mathcal{M}_b, \mathcal{T}_{1:b}, c_I, c)\,. \tag{1}$$

3.2.1 Block-Causal Adaptation

To maintain global identity while ensuring local continuity, as shown in Figure 3 (a), we construct the bounded visual context C_b for each generation step b as $C_b = \big[\ \mathbf{s}\ ;\ \mathbf{x}_{b-K:b-1}\ \big]$, where the first generated block $\mathbf{s} = \mathbf{x}_1$ acts as a persistent sink to anchor long-term visual appearance, and the sliding window of K recent clean blocks provides local temporal continuity across block boundaries.

We fine-tune the Stage 1 model using masked teacher forcing (Chen et al., 2024; Zhu et al., 2026b; Zhao et al., 2026b). A block-causal attention mask ensures that the target block attends only to C_b and its own noisy latents, excluding future blocks to allow parallel teacher-forced training.

3.2.2 Test-Time Memory Consolidation

Long-horizon scene generation with viewpoint revisits demands that the model consistently recovers previously observed visual content. To achieve this, we propose Test-Time Memory Consolidation, a test-time training module that represents each generated block as a key-value association between its camera pose and visual content. As generation progresses, blocks evicted from the active context window are consolidated into a fast-weight memory state via closed-form online updates (Sun et al., 2024; Behrouz et al., 2026; 2025; Zhang et al., 2026b; Dalal et al., 2025; Chen et al., 2026b). The updated memory reconciles new observations with scene history, enabling subsequent generation steps to query and retrieve long-term visual information using camera pose as the key.

Pose-Keyed Fast Weights. Each transformer layer in the backbone is augmented with a lightweight TTT module as parametric scene memory. The memory architecture consists of a two-layer MLP: a fixed projection layer $\mathbf{W}_1$ that maps pose keys into a non-linear feature space via $\boldsymbol{\phi}(\mathbf{k}) = \mathrm{gelu}(\mathbf{W}_1\mathbf{k})$, and a dynamically updated readout matrix $\mathbf{W}_2$ that stores accumulated scene representations across time. Crucially, memory operations are decoupled from diffusion noise, as keys depend solely on camera raymaps, while values are derived from clean latent representations.

Consolidating Writes via Recursive Least Squares. When block e_j is evicted from the context window at the j-th write step, its camera raymap $\mathbf{\Pi}_{e_j}$ and hidden state $\mathbf{h}_{e_j}$ are projected into a key-value association:

$$\mathbf{k}_j = \varphi_K(\mathbf{\Pi}_{e_j}), \qquad \mathbf{v}_j = \varphi_V(\mathbf{h}_{e_j}), \tag{2}$$

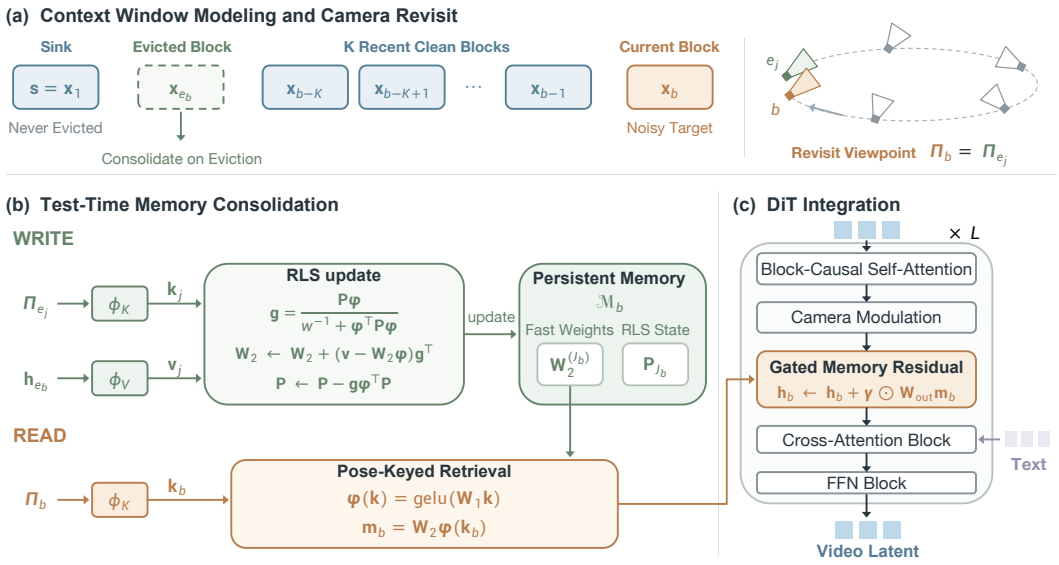

Figure 3: **Stage 2: Autoregressive Teacher with TTT Memory. (a)** Overview of the persistent sink and K recent clean blocks form the bounded context. Revisited views may lie beyond this window. **(b)** WRITE encodes an evicted block's raymap $\mathbf{\Pi}_{e_j}$ and clean hidden state $\mathbf{h}_{e_j}$ into a key-value pair and updates TTT memory state $\mathcal{M}_b$ by recursive least squares (Eq. 4). READ uses the target pose key $\mathbf{k}_b$ to retrieve memory feature $\mathbf{m}_b$ from the fast weights (Eq. 5), without distorting the memory state with J_b writes before block b. **(c)** The retrieved memory is adaptively injected into each DiT block through a gated residual (Eq. 6) for long-horizon consistency during viewpoint revisits.

where φ_K and φ_V denote spatial pose and content encoders, respectively. As illustrated in Figure 3 (b), upon context eviction, the pair $(\mathbf{k}_j, \mathbf{v}_j)$ is consolidated into the TTT memory state $\mathcal{M}_b$.

Memory updates are conducted online, with key-value blocks presented sequentially upon context eviction. When a camera traverses a scene smoothly, consecutive evictions often originate from closely aligned viewpoints, resulting in correlated and near-collinear pose keys. A naive write strategy would repeatedly reinforce these dominant directions, allowing redundant observations to bias the stored memory. To ensure each write contributes novel information, we formulate memory consolidation as a weighted ridge regression problem over the key-value pairs observed up to write j:

$$\min_{\mathbf{W}_2} \sum_{i \le j} w_i \left\| \mathbf{v}_i - \mathbf{W}_2 \boldsymbol{\phi}_i \right\|^2 + \lambda \left\| \mathbf{W}_2 - \mathbf{W}_2^{(0)} \right\|_F^2, \tag{3}$$

where $\boldsymbol{\phi}_j = \mathrm{gelu}(\mathbf{W}_1 \mathbf{k}_j)$ is the pose key feature, w_j is a learned scalar weight, and $\mathbf{W}_2^{(0)}$ is the learned base readout matrix. To solve Eq. 3 online per eviction, we recursively track the inverse regularized feature correlation matrix $\mathbf{P}_j = \left(\sum_{i \le j} w_i \boldsymbol{\phi}_i \boldsymbol{\phi}_i^\top + \lambda \mathbf{I} \right)^{-1}$. Starting from pre-write states $\mathbf{W}_2 = \mathbf{W}_2^{(j-1)}$ and $\mathbf{P} = \mathbf{P}_{j-1}$, the recursive least squares update rule for the j-th write is:

$$\mathbf{g} = \frac{\mathbf{P}\boldsymbol{\phi}}{w^{-1} + \boldsymbol{\phi}^\top \mathbf{P} \boldsymbol{\phi}}, \qquad \mathbf{W}_2 \leftarrow \mathbf{W}_2 + \left(\mathbf{v} - \mathbf{W}_2 \boldsymbol{\phi} \right) \mathbf{g}^\top, \qquad \mathbf{P} \leftarrow \mathbf{P} - \mathbf{g}\, \boldsymbol{\phi}^\top \mathbf{P}, \tag{4}$$

where index j is omitted for brevity. The update direction $\mathbf{P}\boldsymbol{\phi}$ acts as a whitening preconditioner that attenuates feature components along directions already spanned by prior memory, ensuring that only the novel residual updates $\mathbf{W}_2$. Since $\mathbf{W}_1$ remains fixed, Eq. 4 yields the exact minimizer of Eq. 3 at every step, and therefore the consolidated memory state is invariant to the observation order.

We apply Eq. 4 once per eviction step per layer and yield the memory state $\mathcal{M}_b = \left(\mathbf{W}_2^{(J_b)}, \mathbf{P}_{J_b} \right)$ for generating target block b, where J_b counts the total writes completed prior to block b. We set $\lambda = 10^{-3}$ and initialize $\mathbf{P}_0 = \mathbf{I}/\lambda$. This incorporates the base prior $\mathbf{W}_2^{(0)}$ while stabilizing the whitening process to prevent unbounded amplification of subtle key residuals.

Memory Retrieval and Injection. During generation of a target block b, memory reading operates using the target camera raymap $\mathbf{\Pi}_b$ via its pose key $\mathbf{k}_b = \varphi_K(\mathbf{\Pi}_b)$. As shown in Figure 3 (b), memory retrieval is performed via a forward pass:

$$\mathbf{m}_b = \text{read}(\mathbf{k}_b) = \mathbf{W}_2\, \boldsymbol{\phi}(\mathbf{k}_b) = \mathbf{W}_2\, \text{gelu}(\mathbf{W}_1 \mathbf{k}_b). \tag{5}$$

The retrieved memory feature $\mathbf{m}_b$ is then projected back to the backbone feature dimension and injected into the layer's hidden state $\mathbf{h}_b$ via a learned per-channel gate $\boldsymbol{\gamma}$ as shown in Figure 3 (c):

$$\mathbf{h}_b \leftarrow \mathbf{h}_b + \boldsymbol{\gamma} \odot \mathbf{W}_{\text{out}}\, \mathbf{m}_b. \tag{6}$$

Outer-Loop Learning. The TTT memory module is trained jointly with the DiT backbone. Gradients from the standard VDM objective propagate through the recursive update in Eq. 4 to optimize the key projector $\mathbf{W}_1$, the base readout matrix $\mathbf{W}_2^{(0)}$, the encoders φ_K, φ_V, the adaptive sample weights w, and the gating parameters $\boldsymbol{\gamma}$. By jointly optimizing the update mechanism, this outer-loop training aligns the TTT memory with the DiT backbone, enabling the model to dynamically adapt its internal state while preserving a shared representation space. The distinction between learned parameters and online state is central to our formulation. During deployment, the backbone, pose and content encoders, and key projector remain fixed; only the readout matrix and inverse feature-correlation state evolve through Eq 3. Each write therefore incorporates a clean observation without backpropagating through the diffusion transformer, while subsequent reads condition generation on the accumulated scene history.

3.3 Stage 3: Dual-Teacher Few-Step Distillation

The autoregressive model in Stage 2 requires tens of sampling steps per block, which is computationally incompatible with real-time interactive generation. Stage 3 addresses this by distilling the model into a few-step generator.

Following Zhao et al. (2026b), we initialize the few-step student via Consistency Distillation, which discretizes the continuous flow-matching into a 4-step schedule.

Dual-Teacher Distribution Matching Distillation. While consistency distillation enables fast sampling, direct multi-step jumps tend to degrade frame-level visual fidelity and spatial consistency across frames. To recover both, we design a dual-teacher framework with Distribution Matching Distillation (DMD) (Yin et al., 2024; Huang et al., 2026), which factorizes the real-data score into complementary guidance:

- *Bidirectional Score* ($\mathbf{s}_{\text{bidir}}$), provided by the Stage 1 teacher operating over its fixed temporal window, enforcing short-range visual realism and motion coherence.
- *Memory-Augmented Autoregressive Score* ($\mathbf{s}_{\text{mem}}$), provided by the Stage 2 teacher with TTT memory, evaluating latents along extended rollouts to preserve long-horizon spatial consistency.

To compute distribution matching gradients, each teacher is paired with an architecturally isomorphic critic network trained on student-generated samples. Specifically, $\text{critic}_{\text{bidir}}$ mirrors the bidirectional teacher for local quality, while $\text{critic}_{\text{mem}}$ mirrors the autoregressive teacher with an identical TTT memory module. The generator gradient is then formulated by distribution matching updates:

$$\nabla_\theta \mathcal{L}_{\text{DMD}} = \nabla_\theta \mathcal{L}_{\text{DMD}}\big(\text{teacher}_{\text{bidir}}, \text{critic}_{\text{bidir}}\big) + \nabla_\theta \mathcal{L}_{\text{DMD}}\big(\text{teacher}_{\text{mem}}, \text{critic}_{\text{mem}}\big), \tag{7}$$

where each term computes the score-difference gradient on student rollout latents by:

$$\nabla_\theta \mathcal{L}_{\text{DMD}}(\text{teacher}, \text{critic}) = -\,\mathbb{E}_{t,\mathbf{z}_t}\left[\big(\hat{\mathbf{x}}_{0,\text{teacher}}(\mathbf{z}_t) - \hat{\mathbf{x}}_{0,\text{critic}}(\mathbf{z}_t)\big)\frac{\partial \mathbf{x}_\theta}{\partial \theta}\right]. \tag{8}$$

Here, $\mathbf{x}_\theta$ is the clean block generated by the few-step student under parameters θ, and $\mathbf{z}_t$ is its noised counterpart at a randomly sampled timestep t. Prior to computing $\mathbf{s}_{\text{mem}}$, the memory states of both the autoregressive teacher and its critic are initialized by running their memory update schedule over the student's rollout. , Thus, the memory-augmented guidance penalizes cumulative trajectory drift and inconsistent generation of the student under its own rollout conditions.

Table 1: **Comparison on the Revisit Set.** We average across seven VBench visual quality dimensions (0-100 scale). Revisit fidelity evaluates scene consistency via DINOv2 similarity ("Semantic Sim.") between paired views and via PSNR against ground truth (GT). Camera following metrics use scale-only translation alignment and dataset-macro averaging. "Mem" is peak memory during generation of a 30s video on a single NVIDIA B200 GPU in GB. Bold marks the best generated result, and underline indicates the second best. VAE-decoded GT provides a reference for the evaluation.

		Visual Quality ↑	Revisit Fidelity ↑		Camera Following ↓				Efficiency	
Method	Steps	VBench	Semantic Sim.	PSNR	RotErr°	CamMC	RPE Rot.°	RPE Trans.	FPS ↑	Mem ↓
GT (VAE-decoded)	–	75.69	0.9982	–	0.194	0.0055	0.073	0.00102	–	–
Matrix-Game 3.0	3	74.72	0.8225	11.15	11.179	0.2958	0.357	0.00285	7.1	76
HY-World 1.5	4	**75.32**	0.8563	10.84	12.864	0.3484	0.315	0.00395	4.5	110
DreamX-AR	4	74.83	0.8768	11.66	8.011	0.2276	0.448	0.00508	11.0	76
LingBot-Fast	4	74.98	0.8580	10.59	9.817	0.2639	0.317	0.00290	1.6	144
SANA-WM	4	72.87	0.9200	13.52	3.344	0.1200	0.253	0.00439	<u>14.2</u>	**48**
World-T^3 Stage 2	50	<u>75.15</u>	<u>0.9524</u>	<u>16.69</u>	<u>1.308</u>	<u>0.0381</u>	**0.157**	<u>0.00199</u>	4.4	<u>54</u>
World-T^3 Stage 3	4	74.52	**0.9610**	**18.31**	**1.075**	**0.0318**	<u>0.159</u>	**0.00196**	**20.6**	<u>54</u>

4 Experiments

We evaluate whether World-T^3 recovers previously observed scenes beyond its recent context window while following the requested camera trajectory. We measure revisit fidelity, camera control, and visual quality, then examine larger camera sweeps and 4-step distillation. Together, these experiments assess the two requirements motivating our design: retaining scene history within a bounded memory budget and supporting real-time generation.

4.1 Evaluation Protocol

Revisit Benchmark. We use out-and-back camera trajectories to compare generated views at the same requested pose before and after a camera excursion. The revisit set contains 120 clips sampled from the test set of our curated data, each with 160 frames (10 s at 16 FPS) at 1280×704 resolution. The trajectory reverses at frame 80. All methods receive the same initial frame, text prompt, and prescribed camera trajectory. Appendix D.1 details the curation and preprocessing.

Baselines. We compare against Matrix-Game 3.0 (Wang et al., 2026d), HY-World 1.5 (Sun et al., 2025), DreamX-AR (Team et al., 2026b), LingBot-Fast (Team et al., 2026c), and SANA-WM (Zhu et al., 2026a), using their official checkpoints and code. We evaluate both the Stage 2 autoregressive teacher and the Stage 3 few-step student of World-T^3.

Revisit Fidelity. Adapting the revisit protocol of WorldMark (Xu et al., 2026c), we measure memory retention by aggregating revisit consistency across four temporal gaps. The trajectory reverses at frame 80, so we pair outbound frames t with return frames $160 - t$ at the same requested camera pose. We use the zero-indexed pairs $(8, 152)$, $(16, 144)$, $(24, 136)$, and $(32, 128)$, with gaps of 36, 32, 28, and 24 latent frames, respectively. All four gaps exceed our 21-latent-frame recent context window. We average the DINOv2 cosine distances over the four pairs and report one minus this mean as semantic similarity, jointly summarizing scene retention across these revisit intervals. For reference fidelity, we compute PSNR between each generated return view and its corresponding VAE-decoded ground-truth (GT) view, then average the four scores. We compare views at their prescribed times without post-hoc alignment, so both metrics remain sensitive to camera-following errors. Appendix D.2 provides further details.

Camera Control and Visual Quality. We recover camera trajectories with VGGT-Ω (Wang et al., 2026a) and report absolute rotation error (RotErr), camera-matrix error (CamMC), and consecutive-frame rotational and translational relative pose errors (RPE). Translation alignment uses a single scale, preserving the shared first-frame coordinate system. Camera scores are macro-averaged across source datasets, and VAE-decoded GT provides a reference for estimator error. For visual quality, we average seven VBench (Huang et al., 2024) dimensions: imaging quality, aesthetic quality, subject consistency, background consistency, motion smoothness, temporal flickering, and overall consistency. Metric definitions and aggregation are given in Appendix D.2; paired comparisons use exact two-sided sign tests (Appendix D.5).



Figure 4: **Scene Consistency along Out-and-Back Camera Trajectories.** Qualitative comparisons on an indoor scene (top) and an outdoor scene (bottom). Each 10-second camera input follows the forward trajectory for 5 seconds and then retraces it in reverse for 5 seconds. Columns show selected times shared across methods; the first and last columns compare the initial and return views.

4.2 Revisit Fidelity and Camera Control

Revisits beyond the Recent Context. World-T^3 improves both semantic return consistency and pixel-level reference fidelity (Table 1). Compared with SANA-WM, the strongest external baseline on revisit fidelity, our Stage 3 model roughly halves DINOv2 distance, from 0.0800 to 0.0390, and raises PSNR by 4.79 dB. Both gains are supported by paired tests (Appendix D.5). These metrics provide complementary evidence: outbound–return similarity measures retention of generated content, whereas GT-referenced PSNR measures agreement with the reference scene. Their joint improvement supports more faithful scene recovery beyond the recent context. Additional comparisons across DINOv2, PSNR, SSIM, and LPIPS show that both stages outperform all external baselines on the evaluated revisit measures (Appendix D.4).

Trajectory Following. Faithful revisits also require reaching the requested viewpoint. Relative to SANA-WM, World-T^3 Stage 3 reduces absolute RotErr from 3.344° to 1.075°, alongside lower rotational and translational RPE (Table 1). Several baselines combine small consecutive-frame errors with much larger absolute errors, a pattern consistent with accumulated trajectory drift. The lower errors of World-T^3 on both scales indicate more accurate local motion and overall camera following.

Together with the revisit results, these findings show improved scene recovery while executing the prescribed camera excursion.

Qualitative Revisits. Figure 4 compares World-T^3 with Matrix-Game 3.0, HY-World 1.5, SANA-WM, and LingBot-Fast on two out-and-back camera trajectories. The indoor example contains a staircase, windows, mirrors, and sinks; the outdoor example contains a signboard beside a curved path. Comparing the initial and return views reveals changes in scene layout, object appearance, and viewpoint across methods, complementing the quantitative revisit evaluation. Additional qualitative results along a 30-second real camera motion sequence are provided in Appendix A.

Larger Camera Sweeps. We group the clips post hoc into three equal-sized tiers by total yaw sweep normalized by horizontal field of view, a proxy for how much content leaves the view. This grouping depends only on the prescribed trajectories. The student maintains high revisit similarity across all tiers (Table 2). Matrix-Game and DreamX-AR degrade under larger sweeps, whereas several other baselines vary less but remain lower in absolute similarity. Our student model's advantage over SANA-WM is significant in every tier, with the largest mean difference in the highest-sweep tier. Its lower camera error in that tier further supports accurate revisit under larger viewpoint changes (Appendix D.6).

Table 2: **Revisit Consistency under Increasing Camera Sweep.** T1–T3 are equal-sized terciles (40 clips each) of yaw sweep normalized by horizontal field of view. Higher tiers indicate larger sweeps. Bold and underline mark the best and second-best generated similarities in each tier, respectively.

Method	T1	T2	T3	T3−T1
GT (VAE-decoded)	0.9981	0.9983	0.9981	0.000
Matrix-Game 3.0	0.8743	0.8196	0.7737	−0.101
HY-World 1.5	0.8612	0.8581	0.8497	−0.012
DreamX-AR	0.9074	0.8941	0.8289	−0.079
LingBot-Fast	0.8532	0.8701	0.8506	−0.003
SANA-WM	0.9201	0.9311	0.9090	−0.011
World-T^3 Stage 2	<u>0.9434</u>	<u>0.9501</u>	<u>0.9494</u>	+0.006
World-T^3 Stage 3	**0.9582**	**0.9644**	**0.9603**	+0.002

4.3 Step Distillation and Efficiency

Stage 3 distillation (§3.3) reduces sampling from 50 to 4 steps while retaining strong revisit performance. The student increases revisit PSNR from 16.69 to 18.31 dB and improves revisit LPIPS, with both gains supported by paired tests. Its mean DINOv2 revisit distance is also lower, with $72/120$ paired wins ($p = 0.0353$). Paired tests detect no significant difference between the teacher and student in the four camera metrics (Appendix D.7). The trade-off is in visual quality and local consistency: VBench decreases from 75.15 to 74.52, and Local Memory from 0.9442 to 0.9258. The four-step student therefore combines improved pixel-level revisit fidelity with modest reductions in these quality scores.

To support real-time generation, we combine memory-read caching, compilation, efficient attention kernels, and optimized VAE decoding(Appendix E). With online memory updates enabled, the four-step student achieves an end-to-end throughput of 20.6 FPS at 1280×704 on a single NVIDIA B200 GPU (Table 1; Fig. 1b), exceeding the 16 FPS playback rate. The fixed-size memory and bounded recent context keep memory storage and per-step access costs independent of rollout length (§3.2).

5 Conclusion

We presented World-T^3, a camera-controlled world model that consolidates historical observations into a parametric memory by test-time training (TTT). To address spatial redundancy along smooth camera trajectories, we introduced an online recursive least-squares update rule that selectively writes novel visual contexts while preventing stored scene memory from degradation. We instantiated World-T^3 through progressive three-stage training that transitions a pretrained video diffusion into an autoregressive world model with TTT memory, followed by dual-teacher distillation for few-step generation. Evaluations show that World-T^3 reliably preserves persistent world state, achieving superior revisit consistency and sustaining 20.6 FPS at 1280×704 on one NVIDIA B200 GPU.

AI USE STATEMENT

In this work, we used generative AI tools for editing a research paper to improve readability, editing code for method implementation, and searching for information. We have not used generative AI tools for generating synthetic data sets, helping develop theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proving mathematical claims, assisting in the writing of proofs, proposing or refining hypotheses, designing or providing feedback on research methodology or experiments, assisting with translation, cleaning and reformating datasets, supporting qualitative and thematic data analysis, or interpreting results.

We have reviewed all AI-assisted work. Specifically, LLM-assisted paper writing edits and code implementations were thoroughly verified and tested for correctness, rigor, and factual precision by the authors. We take full responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provided comprehensive details regarding our model architecture, mathematical formulations, training pipeline, and evaluation protocols throughout the paper and appendix. Specifically, the pose-keyed fast-weight architecture and the closed-form online update rule based on recursive least squares are detailed in Section 3. The progressive three-stage training pipeline, including Bidirectional SFT, Block-Causal Autoregressive Training, and Dual-Teacher Few-Step Distillation, is further elaborated with optimization settings, learning rate schedules, and loss weights in Appendix C. For dataset construction, Appendix B provides a thorough description of our curated training corpus, covering camera pose annotation, filtering, and the clip replay procedure for viewpoint revisit construction. Finally, evaluation protocols, benchmark, metrics, and baseline methods are specified in Section 4 and Appendix D.

REFERENCES

Jason Ansel, Edward Yang, Horace He, Natalia Gimelshein, Animesh Jain, Michael Voznesensky, Bin Bao, Peter Bell, David Berard, Evgeni Burovski, et al. Pytorch 2: Faster machine learning through dynamic python bytecode transformation and graph compilation. In *Proceedings of the 29th ACM international conference on architectural support for programming languages and operating systems, volume 2*, pp. 929–947, 2024.

Ali Behrouz, Zeman Li, Praneeth Kacham, Majid Daliri, Yuan Deng, Peilin Zhong, Meisam Razaviyayn, and Vahab Mirrokni. Atlas: Learning to optimally memorize the context at test time. *arXiv preprint arXiv:2505.23735*, 2025.

Ali Behrouz, Peilin Zhong, and Vahab Mirrokni. Titans: Learning to memorize at test time. *Advances in Neural Information Processing Systems*, 38:113506–113543, 2026.

Yuxuan Bian, Zeyue Xue, Songchun Zhang, Shiyi Zhang, Weiyang Jin, Yaowei Li, Junhao Zhuang, Haoran Li, Jie Huang, Haoyang Huang, et al. Echo-infinity: Learning evolving memory for real-time infinite video generation. *arXiv preprint arXiv:2606.04527*, 2026.

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

Haoxuan Che, Xuanhua He, Quande Liu, Cheng Jin, and Hao Chen. Gamegen-x: Interactive open-world game video generation. In *International Conference on Learning Representations*, volume 2025, pp. 37546–37593, 2025.

Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. *Advances in Neural Information Processing Systems*, 37:24081–24125, 2024.

Shuo Chen, Cong Wei, Sun Sun, Ping Nie, Kai Zhou, Ge Zhang, Ming-Hsuan Yang, and Wenhu Chen. Context forcing: Consistent autoregressive video generation with long context. *arXiv preprint arXiv:2602.06028*, 2026a.

Xingyu Chen, Yue Chen, Yuliang Xiu, Andreas Geiger, and Anpei Chen. Ttt3r: 3d reconstruction as test-time training. In *International Conference on Learning Representations*, volume 2026, pp. 50694–50718, 2026b.

Sharan Chetlur, Cliff Woolley, Philippe Vandermersch, Jonathan Cohen, John Tran, Bryan Catanzaro, and Evan Shelhamer. cudnn: Efficient primitives for deep learning. *arXiv preprint arXiv:1410.0759*, 2014.

Karan Dalal, Daniel Koceja, Jiarui Xu, Yue Zhao, Shihao Han, Ka Chun Cheung, Jan Kautz, Yejin Choi, Yu Sun, and Xiaolong Wang. One-minute video generation with test-time training. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17702–17711. IEEE, 2025.

Juechu Dong, Boyuan Feng, Driss Guessous, Yanbo Liang, and Horace He. Flex attention: A programming model for generating optimized attention kernels. In *Proceedings of Machine Learning and Systems*, 2025.

Ruili Feng, Han Zhang, Zhilei Shu, Zhantao Yang, Longxiang Tang, Zhicai Wang, Andy Zheng, Jie Xiao, Zhiheng Liu, Ruihang Chu, et al. The matrix: Infinite-horizon world generation with real-time moving control. *Advances in Neural Information Processing Systems*, 38:87318–87344, 2026a.

Yusen Feng, Bingchen Han, Jiangran Lyu, Kai Liu, Yixin Zheng, Yuxuan Wan, Weiheng Liu, Sun Han, Ruiqin Li, Yulong Zhang, et al. Wam-ttt: Steering world-action models by watching human play at test time. *arXiv preprint arXiv:2607.06988*, 2026b.

Hao He, Yinghao Xu, Yuwei Guo, Gordon Wetzstein, Bo Dai, Hongsheng Li, and Ceyuan Yang. Cameractrl: Enabling camera control for text-to-video generation. *arXiv preprint arXiv:2404.02101*, 2024.

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 (TOG)*, 44(6): 1–15, 2025.

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. *Advances in Neural Information Processing Systems*, 38:167283–167308, 2026.

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

Sam Ade Jacobs, Masahiro Tanaka, Chengming Zhang, Minjia Zhang, Shuaiwen Leon Song, Samyam Rajbhandari, and Yuxiong He. Deepspeed ulysses: System optimizations for enabling training of extreme long sequence transformer models. *arXiv preprint arXiv:2309.14509*, 2023.

Yunfan Jiang, Yevgen Chebotar, Ruijie Zheng, Fengyuan Hu, Yunhao Ge, Jimmy Wu, Tianyuan Dai, Scott Reed, Li Fei-Fei, Yuke Zhu, et al. Robottt: Context scaling for robot policies. *arXiv preprint arXiv:2607.15275*, 2026.

Youngrae Kim, Qixin Hu, C-C Jay Kuo, and Peter A Beerel. Memrope: Training-free infinite video generation via evolving memory tokens. *arXiv preprint arXiv:2603.12513*, 2026.

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

Kewei Lian, Shaofei Cai, Yitao Liang, and Anji Liu. Loopnav: Benchmarking spatial consistency in world models. *arXiv preprint arXiv:2505.22976*, 2025.

Fangfu Liu, Diankun Wu, Jiawei Chi, Yimo Cai, Yi-Hsin Hung, Xumin Yu, Hao Li, Han Hu, Yongming Rao, and Yueqi Duan. Spatial-ttt: Streaming visual-based spatial intelligence with test-time training. *arXiv preprint arXiv:2603.12255*, 2026.

Jinpeng Lu, Dexu Zhu, Haoyuan Shi, Linghan Cai, Guo Tang, Yinda Chen, Jie Cao, Duyu Tang, Yi Zhang, Yong Dai, et al. Current world models lack a persistent state core. *arXiv preprint arXiv:2606.20545*, 2026.

NVIDIA. CUDA C++ programming guide: CUDA graphs. https://docs.nvidia.com/cuda/cuda-c-programming-guide/, 2025.

Tianchang Shen, Sherwin Bahmani, Kai He, Sangeetha Grama Srinivasan, Tianshi Cao, Jiawei Ren, Ruilong Li, Zian Wang, Nicholas Sharp, Zan Gojcic, et al. Lyra 2.0: Explorable generative 3d worlds. *arXiv preprint arXiv:2604.13036*, 2026.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. *arXiv preprint arXiv:2303.01469*, 2023.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

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.

Yu Sun, Xinhao Li, Karan Dalal, Jiarui Xu, Arjun Vikram, Genghan Zhang, Yann Dubois, Xinlei Chen, Xiaolong Wang, Sanmi Koyejo, et al. Learning to (learn at test time): Rnns with expressive hidden states. *arXiv preprint arXiv:2407.04620*, 2024.

AlayaWorld Team, Kaipeng Zhang, Chuanhao Li, Yifan Zhan, Yongtao Ge, Yuanyang Yin, Jiaming Tan, Kang He, Liaoyuan Fan, Mingliang Zhai, et al. Alayaworld: Interactive long-horizon world modeling–full technical report. *arXiv preprint arXiv:2607.18367*, 2026a.

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

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

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

Jianyuan Wang, Minghao Chen, Shangzhan Zhang, Nikita Karaev, Johannes Schönberger, Patrick Labatut, Piotr Bojanowski, David Novotny, Andrea Vedaldi, and Christian Rupprecht. Vggt-$omega$. *arXiv preprint arXiv:2605.15195*, 2026a.

Weijie Wang, Haoyu Zhao, Yifan Yang, Feng Chen, Zeyu Zhang, Yefei He, Zicheng Duan, Donny Y Chen, Yuqing Yang, and Bohan Zhuang. Latent spatial memory for video world models. *arXiv preprint arXiv:2606.09828*, 2026b.

Yixuan Wang, Rhythm Syed, Fangyu Wu, Mengchao Zhang, Aykut Onol, Jose Barreiros, Hooshang Nayyeri, Tony Dear, Huan Zhang, and Yunzhu Li. Interactive world simulator for robot policy training and evaluation. *arXiv preprint arXiv:2603.08546*, 2026c.

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

Zile Wang, Zexiang Liu, Jiaxing Li, Kaichen Huang, Baixin Xu, Fei Kang, Mengyin An, Peiyu Wang, Biao Jiang, Yichen Wei, et al. Matrix-game 3.0: Real-time and streaming interactive world model with long-horizon memory. *arXiv preprint arXiv:2604.08995*, 2026d.

Zhengxuan Wei, Xu Guo, Xinghui Li, Xunzhi Xiang, Min Wei, Yiran Zhu, Qiulin Wang, Xintao Wang, Pengfei Wan, Xiangwang Hou, et al. Geometry-aware implicit memory for video world models. *arXiv preprint arXiv:2606.02436*, 2026.

Haoning Wu, Erli Zhang, Liang Liao, Chaofeng Chen, Jingwen Hou, Annan Wang, Wenxiu Sun, Qiong Yan, and Weisi Lin. Exploring video quality assessment on user generated contents from aesthetic and technical perspectives. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 20087–20097. IEEE, 2023.

Zeqi Xiao, Yushi Lan, Yifan Zhou, Wenqi Ouyang, Shuai Yang, Yanhong Zeng, and Xingang Pan. Worldmem: Long-term consistent world simulation with memory. *Advances in Neural Information Processing Systems*, 38:49632–49652, 2026.

Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, Fisher Yu, Dacheng Tao, and Andreas Geiger. Unifying flow, stereo and depth estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(11):13941–13958, 2023.

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 *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pp. 1–12, 2026b.

Xiaojie Xu, Zhengyuan Lin, Kang He, Yukang Feng, Xiaofeng Mao, Yuanyang Yin, Kaipeng Zhang, and Yongtao Ge. Worldmark: A unified benchmark suite for interactive video world models. *arXiv preprint arXiv:2604.21686*, 2026c.

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

Yixuan Ye, Xuanyu Lu, Yuxin Jiang, Yuchao Gu, Rui Zhao, Qiwei Liang, Jiachun Pan, Fengda Zhang, Weijia Wu, and Alex Jinpeng Wang. Mind: Benchmarking memory consistency and action control in world models. *arXiv preprint arXiv:2602.08025*, 2026.

Minghao Yin, Jiahao Lu, Wenbo Hu, Wang Zhao, Shan Ying, and Kai Han. Scope: Sightline-coordinate positional encoding for video diffusion transformers, 2026. URL https://arxiv.org/abs/2606.27345.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation, 2024. *URL https://arxiv.org/abs/2311.18828*, 3, 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 *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22963–22974. IEEE, 2025.

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 *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*, pp. 1–11, 2025.

Ted Zadouri, Markus Hoehnerbach, Jay Shah, Timmy Liu, Vijay Thakkar, and Tri Dao. Flashattention-4: Algorithm and kernel pipelining co-design for asymmetric hardware scaling. *arXiv preprint arXiv:2603.05451*, 2026.

Cheng Zhang, Boying Li, Meng Wei, Yan-Pei Cao, Camilo Gambardella, Dinh Phung, and Jianfei Cai. Unified camera positional encoding for controlled video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 38027–38037, 2026a.

Tianyuan Zhang, Sai Bi, Yicong Hong, Kai Zhang, Fujun Luan, Songlin Yang, Kalyan Sunkavalli, William Freeman, and Hao Tan. Test-time training done right. In *International Conference on Learning Representations*, volume 2026, pp. 157604–157638, 2026b.

Yifan Zhang, Chunli Peng, Boyang Wang, Puyi Wang, Qingcheng Zhu, Fei Kang, Biao Jiang, Zedong Gao, Eric Li, Yang Liu, et al. Matrix-game: Interactive world foundation model. *arXiv preprint arXiv:2506.18701*, 2025.

Jinjing Zhao, Fangyun Wei, Zhening Liu, Hongyang Zhang, Chang Xu, and Yan Lu. Spatia: Video generation with updatable spatial memory. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4245–4257, 2026a.

Min Zhao, Hongzhou Zhu, Kaiwen Zheng, Zihan Zhou, Bokai Yan, Xinyuan Li, Xiao Yang, Chongxuan Li, and Jun Zhu. Causal forcing++: Scalable few-step autoregressive diffusion distillation for real-time interactive video generation. *arXiv preprint arXiv:2605.15141*, 2026b.

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*, 2026a.

Hongzhou Zhu, Min Zhao, Guande He, Hang Su, Chongxuan Li, and Jun Zhu. Causal forcing: Autoregressive diffusion distillation done right for high-quality real-time interactive video generation. *arXiv preprint arXiv:2602.02214*, 2026b.

APPENDIX

A ADDITIONAL QUALITATIVE REVISIT RESULTS

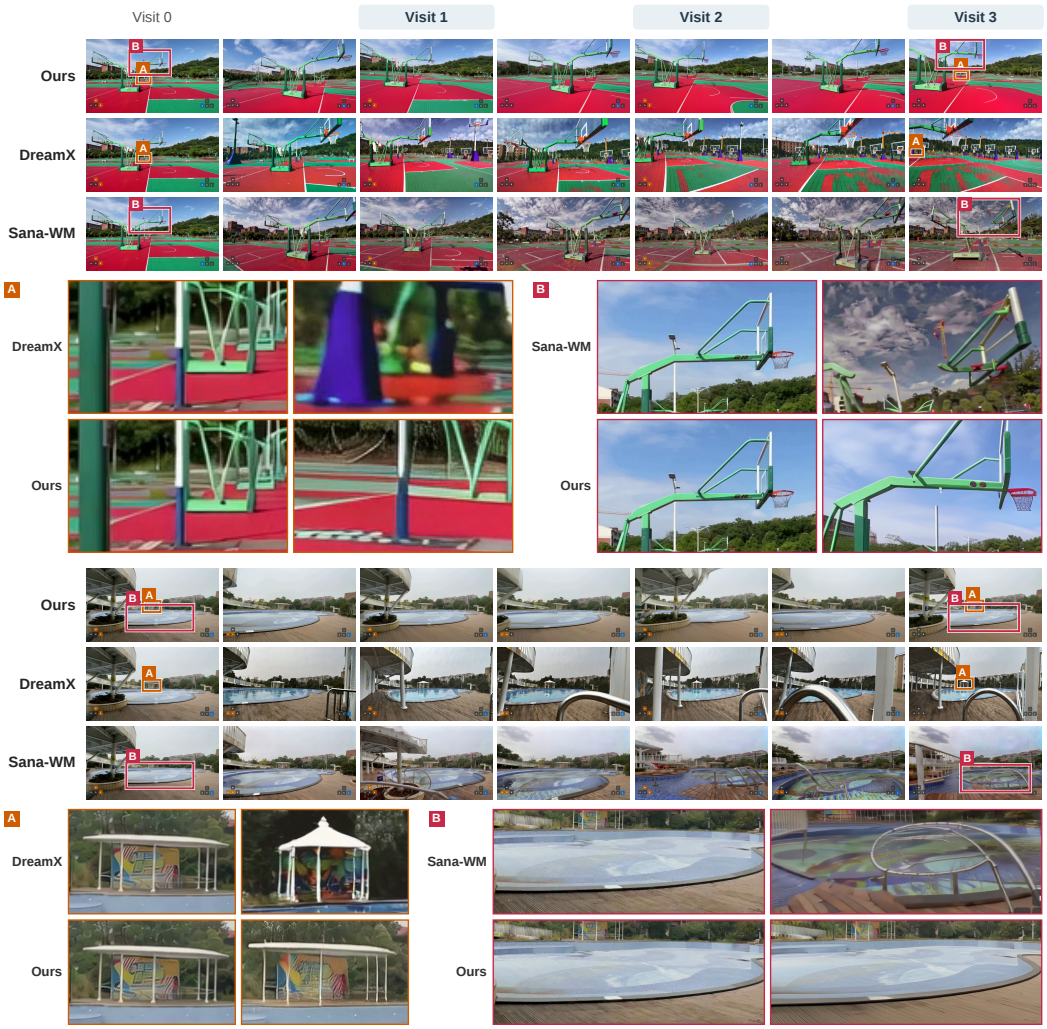

Figure 5: **Scene Consistency across Approximate Revisits.** We identify three approximate revisits within a 30-second real camera motion sequence. Unlike the prescribed out-and-back trajectory used for quantitative evaluation, these revisits observe the same scene regions from similar, rather than identical, camera poses. Visit 0 denotes the initial observation; Visits 1, 2, and 3 mark the selected revisits in temporal order. Colored boxes link full views to enlarged crops comparing local details in the baselines and World-T^3. All enlarged crops compare Visits 0 and 3.

Qualitative Revisits. Figure 5 provides a qualitative comparison along a 30-second real camera motion sequence, during which World-T^3 continually updates its scene memory through TTT. We select three approximate revisits and enlarge local details to inspect scene retention. The 10-second quantitative benchmark uses a prescribed out-and-back trajectory to pair views at the same requested pose; here, the selected views revisit the same scene regions without exact pose matching. At the basketball court, World-T^3 retains the slim blue lower section of the lamp post, whereas DreamX-AR produces a broad purple pedestal (crop A). SANA-WM disconnects a backboard from its support (crop B). At the pool, DreamX-AR changes the pavilion's flat roof to a pointed roof, and SANA-WM alters the pool-surface pattern. World-T^3 more consistently retains these structures and appearance cues across visits, despite some residual changes.

B TRAINING DATA

Learning a camera-controllable video world model that maintains persistent scene memory imposes four complementary requirements on the training corpus: visual richness and diversity, precise camera pose annotations to ground the control signal, sequence lengths commensurate with the target generation horizon, and viewpoint revisits, whereby a previously observed region must be faithfully recalled even after substantial intervening content has elapsed beyond the model's context. To satisfy these desiderata, we curate a base collection of 601K clips that provides broad coverage of appearance and camera motion and an augmented set of 307K clips designed for viewpoint recurrence.

B.1 CORPUS COMPOSITION

Following recently released work in video world models (Zhu et al., 2026a; Team et al., 2026b), our training corpus draws on both real-world footage and synthetic renderings to jointly cover natural appearance, diverse camera dynamics, and precise geometric annotation. Every clip is preprocessed into a unified data structure comprising video latents, per-frame camera intrinsics and extrinsics, and a free-form language scene caption, then resampled to 1280×704 resolution at 16 FPS.

Real-World Sources. Captured video provides the naturalistic appearance and complex illumination. Our real-world subset spans indoor and outdoor environments, encompassing casual orbiting and sweeping motions around objects or scenes, egocentric urban traversal through city streets, and smooth architectural walkthroughs of interior spaces. Together, these sources furnish a broad distribution over both scene category and camera trajectory, with an emphasis on forward-moving, first-person viewpoints in open environments.

Synthetic Sources. Game-engine renderings complement real capture along two axes. First, they supply ground-truth camera pose free of estimation error, serving as a noise-free supervision signal. Second, they introduce trajectory patterns such as sustained aerial flight and systematic ground-level exploration that are difficult to acquire in the physical world, thereby extending the motion distribution covered by the corpus.

B.2 CAMERA ANNOTATION

To obtain coherent camera parameterization, we apply VGGT-Ω (Wang et al., 2026a) to all training clips, recovering per-frame intrinsics and extrinsics within a unified scale and anchoring each trajectory to the initial frame as the world origin. The resulting camera trajectories serve as the conditioning signal for controllable video generation and further induce a per-clip co-visibility matrix that encodes pairwise surface overlap between frames, quantifying the degree of viewpoint revisit throughout each sequence.

B.3 DATA FILTERING

We filter each clip through a two-stage pipeline: an objective criteria drawn from the imagery and the recovered geometry, and a semantic assessment produced by vision-language models.

Objective Scoring. The objective criteria span both visual content and camera geometry, following the conventions established for camera-controllable corpora (Zhu et al., 2026a; Team et al., 2026c; Xu et al., 2026a; Team et al., 2026a). Perceptual quality is assessed with DOVER (Wu et al., 2023), which produces complementary technical and aesthetic scores: the technical score captures low-level distortions such as blur and artefacts, while the aesthetic score reflects content and composition. To characterize scene dynamics, we compute optical flow magnitude via UniMatch (Xu et al., 2023) between frames and impose bilateral bounds. Clips whose flow falls below a lower threshold convey negligible information per unit duration, whereas those exceeding an upper threshold suffer from motion blur and inter-frame disocclusion. Mean luminance is similarly constrained to a permissible band, eliminating under-exposed footage in which scene structure is irrecoverable and over-exposed footage in which highlight detail is clipped. On the geometric side, we retain only clips whose estimated field of view and focal variation remain within the regime where the Plücker ray representation of §3.1 provides a faithful camera embedding.

Semantic Profiling. Beyond photometric and geometric criteria, we employ a vision-language model to perform semantic assessment. Video frames are sampled uniformly across each clip and presented to the model, which annotates the scene along several aspects: the count of dynamic entities (pedestrians, vehicles, animals), the presence of synthetic overlays (UI chrome, loading screens, watermarks), perceptual blur, and viewpoint characterization. Responses conform to a structured schema in which factual observations and a concise rationale precede the binary usability verdict, enforcing a chain-of-thought discipline that grounds the decision in explicit evidence. Final admission into the corpus is gated jointly on the entity count and the usability label.

B.4 Text Captioning

Captions serve as the textual conditioning signal and are generated by a vision-language model prompted to produce scene-static descriptions (Team et al., 2026c; Zhu et al., 2026a) in a single paragraph. A scene-static caption characterizes the environment in terms of its constituent objects, spatial arrangement, materials, colour palette, and illumination, together with dynamics intrinsic to the scene itself, such as flowing water or drifting fog. Crucially, the caption abstracts away all camera-related language: viewpoint references, cinematographic terminology, and motion verbs such as *pan*, *dolly*, or *fly over* are systematically excluded. The guiding principle is invariance: a given scene should elicit semantically equivalent captions regardless of the trajectory along which it is observed, ensuring that camera motion is communicated exclusively through the pose conditioning branch and that the two modalities of control remain disentangled.

B.5 Revisit Construction

Revisits, denoting returns to previously observed regions, arise sparingly in natural video, since a camera traversing a scene seldom retraces its path. As a result, the co-visibility statistic remains low across most of the base set. This scarcity hinders learning an effective memory mechanism, which relies on repeated observations of the same content. To expose the model to such recurrence in real-world data, we construct revisits through a clip replay procedure. Specifically, we loop and re-index the camera trajectory of a video sequence so that the temporal path periodically returns to previously observed viewpoints at distinct temporal intervals within the same training sequence. We confine the replay to static scenes and exclude dynamic entities that change between visits to avoid abnormal movements. For synthetic data, we further design camera trajectories as closed loops through the rendered environments, so that the viewpoint deliberately returns to previously visited locations. Taken together, these procedures yield 307K clips with recurring viewpoints.

C Training Details

The three stages of our method are implemented as five consecutive training runs: bidirectional camera-conditioned supervised fine-tuning (SFT), block-causal adaptation, memory-augmented training, consistency distillation (CD), and distribution matching distillation (DMD). The first run constitutes Stage 1, the next two constitute Stage 2, and the last two constitute Stage 3. Table 3 summarizes their optimization settings.

C.1 Shared Configuration

We initialize from Wan2.2-TI2V-5B, whose diffusion transformer (DiT) has 30 layers, hidden width 3072, and 24 attention heads of dimension 128. All runs use 1280×704 videos at 16 FPS and BF16 mixed precision. The frozen WanVideoVAE38 maps videos to 48-channel latents with $16\times$ spatial compression and $4\times$ temporal compression. A video of T frames produces $F = 1 + (T-1)/4$ latent frames on a 44×80 spatial grid; the DiT uses patches of size $(1, 2, 2)$. Thus, 21 and 63 latent frames correspond to 81 and 249 video frames, respectively. All context-window, sink, block-size, and rollout-length settings below count latent frames.

The first frame conditions generation through its clean latent at timestep zero, inserted directly into the temporal sequence. The denoising loss excludes this first latent frame. Text conditioning uses cached T5 embeddings through cross-attention; the text encoder and VAE remain frozen. Camera conditioning uses Plücker raymaps and camera-dependent adaptive layer normalization (CamAdaLN).

Table 3: **Training configuration.** AR denotes block-causal adaptation, and Memory denotes subsequent memory-augmented full fine-tuning. Learning rates are peak values; clipping thresholds are gradient norms. DMD steps count critic iterations; the generator is updated every five iterations.

	Stage 1	Stage 2		Stage 3	
	SFT	AR	Memory	CD	DMD
Training iterations	6450	10,000	15,000	2500	800
Latent frames	21	21	63	63	63
GPUs	512	256	128	128	128
Batch per GPU	2	2	1	1	1
Global batch	1024	512	128	128	128
Backbone/generator LR	5×10^{-5}	5×10^{-5}	5×10^{-5}	2×10^{-6}	2×10^{-6}
Memory LR multiplier	–	–	15	1	–
Critic LR	–	–	–	–	4×10^{-7}
Warm-up iterations	500	500	500	200	50
LR floor / peak	0.01	0.1	0.1	0.1	0.1
AdamW β_1	0.9	0.9	0.9	0	0
AdamW β_2	0.999	0.999	0.999	0.999	0.999
Weight decay	0.01	0.01	0.01	0	0
Gradient clipping	0.5	0.5	0.5	0.5	1.0

We use AdamW throughout, with stage-specific hyperparameters in Table 3. Gradient accumulation is one, so the effective global batch is the per-GPU batch multiplied by the number of data-parallel workers.

C.2 Stage 1: Bidirectional Camera-Conditioned SFT

We train rank-stabilized LoRA adapters of rank 256 and scaling parameter $\alpha = 16$ on the attention query, key, value, and output projections and both feed-forward projections, together with CamAdaLN modules of rank 512. The remaining pretrained backbone weights are frozen during this run. The effective adapter scaling is $\alpha/\sqrt{r} = 1$. Training uses bidirectional attention over 21 latent frames and a camera-conditioned denoising objective. Camera conditioning is dropped with probability 0.05, with one dropout decision shared across the batch.

We use distributed data parallelism and AdamW with $\epsilon = 10^{-8}$. The learning rate warms up for 500 iterations to 5×10^{-5}, then follows cosine decay toward 5×10^{-7} with a configured horizon of 25,000 iterations.

C.3 Stage 2: Autoregressive Adaptation and Memory Training

Block-Causal Adaptation. Starting from the Stage 1 checkpoint, we continue training the same LoRA adapters and camera modules for 10,000 iterations using masked teacher forcing. Attention is block-causal, with three latent frames per block, and each block attends to itself and preceding blocks within a 21-latent sequence. We retain distributed data parallelism, AdamW with $\epsilon = 10^{-8}$, and the peak learning rate of 5×10^{-5}. After 500 warm-up iterations, cosine decay reaches 5×10^{-6} at iteration 10,000.

Memory-Augmented Full Fine-Tuning. We merge the iteration-10,000 LoRA adapters into the backbone with effective scaling 1, initialize the TTT memory modules, and jointly train the entire DiT, camera modules, and memory modules for 15,000 iterations. The memory parameter group uses a learning-rate multiplier of 15, giving a peak learning rate of 7.5×10^{-4}, while the backbone peak is 5×10^{-5}. AdamW uses $\epsilon = 10^{-8}$. The backbone schedule uses 500 warm-up iterations followed by global-step cosine decay with an absolute floor of 5×10^{-6}. Its cosine horizon is 8000 through iteration 8000 and 16,000 thereafter; the continued run evaluates the schedule at the continuing global iteration rather than restarting the step count. We use FSDP2 full sharding for the DiT and replicate the memory modules across workers.

Each training episode contains 63 latent frames, with three latent frames per autoregressive block, a 21-latent attention window, and a persistent three-latent sink. Memory writes use the recursive

least-squares readout update (`rls_w2`) with regularization $\lambda = 10^{-3}$, yielding 14 writes per episode. Clean-context keys and values are encoded once and detached; the episode uses one backward pass without truncated backpropagation through time (`tbptt_k=0`). The training manifest contains 627,053 clips, of which 48.6% use ping-pong replay. Replay clips are constructed by decoding an 81-frame source segment, alternating forward and reverse traversal to form 249 frames, and re-encoding to 63 latents, with camera poses following the same temporal indexing.

C.4 Stage 3: Few-Step Distillation

Consistency Distillation. We initialize the student from the Stage 2 memory teacher and train all DiT, camera, and memory parameters for 2500 iterations on the same dataset. CD retains 63-latent episodes, three-latent blocks, the 21-latent window, the three-latent sink, and the RLS update with $\lambda = 10^{-3}$. The consistency discretization has $N = 48$; the four-step sampling schedule uses indices $[0, 12, 24, 36]$ on this grid. The teacher guidance scale is 1.0. All trainable parameter groups, including memory, use a peak learning rate of 2×10^{-6}, with 200 warm-up iterations and cosine decay to 10% of the peak. The gradient clipping threshold is 0.5, and the trainer's gradient-spike threshold is 3.0. We use FSDP2, seed 0, and an exponential moving average (EMA) with decay 0.99, starting at iteration 200 and updated every iteration. The iteration-2500 EMA weights initialize the DMD generator.

Dual-Teacher Distribution Matching. DMD uses a new optimizer initialized from the CD weights. The frozen bidirectional teacher from the Stage 1, and the frozen memory teacher from the Stage 2. Each teacher has a trainable critic with the same architecture, initialized by copying that teacher's weights; the memory critic is created when its loss term is activated. Both memory-augmented scoring networks construct their memory states from the student's own rollout.

The retained DMD run spans iterations 800. Iterations 1–399 use only the bidirectional score term, with weight $w_A = 1$. Iterations 400–800 use both terms, with $w_A = 1$ and memory-term weight $w_B = 0.4044$. At iteration 400, the generator learning rate undergoes an additional 50-iteration warm-up; this boundary warm-up applies only to the generator. Critics are updated every iteration and the generator every fifth iteration, giving 160 generator updates over the retained run. The peak learning rates are 2×10^{-6} for the generator and 4×10^{-7} for the critics, with 50 initial warm-up iterations and cosine decay to 10% of each peak. AdamW uses $(\beta_1, \beta_2) = (0, 0.999)$, zero weight decay, and gradient clipping at 1.0. We use FSDP2, seed 0, and generator EMA with decay 0.99 starting at iteration 200.

For each iteration, the rollout length is sampled uniformly from $\{21, 24, \ldots, 63\}$ latent frames and synchronized across workers. The DMD loss is applied to the final contiguous 21 latent frames, using the same four-step denoising schedule as CD. The final model is taken from the iteration-800 checkpoint after the dual-teacher phase.

D Long-Horizon Revisit: Protocol and Additional Results

D.1 Evaluation Set and Shared Preprocessing

The revisit test set contains 120 clips. Selection uses a total angular sweep divided by horizontal field of view of at least 0.30, followed by the smallest camera-translation spans within each source domain under quotas approximating the pool's domain proportions. The median angular sweep is 38°, and the median sweep/FOV ratio is 0.46. The geometry-based rules aim to test return views after appreciable camera motion while limiting translation. The subset was finalized during exploratory evaluation, including examination of model quality on candidate subsets, rather than as a blind held-out benchmark. Its domain stratification addresses the earlier subset's overrepresentation of large camera turns; it does not establish representativeness for all possible interactive trajectories.

Every method uses the same per-case seed and prescribed trajectory. We evaluate exactly 160 frames at 16 FPS and 1280×704, using frame-count truncation rather than duration-based truncation to avoid occasional 161-frame outputs. Videos are encoded with H.264 (CRF 12, yuv420p). The GT reference is reconstructed from 41 cached VAE latents, which decode to 161 frames, then truncated to 160. GT passes through one video-encoding stage, whereas generated outputs are re-encoded from their native videos and thus pass through two; this asymmetry can slightly favor GT. The cached source

sequence contains only 249 decoded frames (about 15.6 s), so it cannot support a GT-referenced 30 s evaluation.

D.2 Metric Definitions

Revisit Similarity. For generated clip x^m, let $f(I)$ be the L2-normalized mean of the non-CLS patch tokens from DINOv2 ViT-B/14 (`facebook/dinov2-base`). Images are resized to short side 256, center-cropped to 224×224, and normalized using the ImageNet mean and standard deviation. With zero-indexed frame pairs $\mathcal{P} = \{(8, 152), (16, 144), (24, 136), (32, 128)\}$, define

$$d(I, I') = 1 - f(I)^\top f(I'), \qquad \bar{d}_m = \frac{1}{4} \sum_{(a,b)\in\mathcal{P}} d(x_a^m, x_b^m), \qquad S_m = 1 - \bar{d}_m. \tag{9}$$

We average these per-clip quantities over the 120 clips. The gaps correspond to 36, 32, 28, and 24 latent frames, respectively, all exceeding the 21-latent-frame recent KV window. This tests return views beyond the recent window; it does not by itself isolate fast-weight memory from other persistent conditioning. Our probe adapts WorldMark's revisit construction (Xu et al., 2026c), but uses four predetermined pairs instead of flow-matched pairs. Fixed-time pairing evaluates the commanded camera trajectory and intentionally retains camera-following errors. We report raw distance or similarity, not WorldMark's normalized $\mathrm{RM} = (1 + \bar{d}/\tau_c)^{-1}$. To complement self-consistency, we compare each generated outbound and return view with its corresponding VAE-decoded reference g and average over the same four pairs:

$$d_{\text{first}} = \frac{1}{4} \sum_{(a,b)\in\mathcal{P}} d(x_a^m, g_a), \qquad d_{\text{return}} = \frac{1}{4} \sum_{(a,b)\in\mathcal{P}} d(x_b^m, g_b).$$

We average these per-clip scores over the 120 clips and report $\Delta = d_{\text{return}} - d_{\text{first}}$, the mean increase in reference distance from the first visits to the return visits. These reference-based comparisons and camera errors help distinguish faithful replay from self-consistent but incorrect scene content. Center cropping limits sensitivity to content near the image boundary.

Pixel Fidelity. For each outbound–return pair $(a, b) \in \mathcal{P}$, we compute revisit PSNR and SSIM between the generated return view x_b^m and its VAE-decoded reference g_b at the full 1280×704 resolution. We average the scores over the four pairs within each clip and then across clips. All evaluated frames are extracted losslessly as PNGs. For RGB values in $[0, 1]$, $\mathrm{PSNR} = 10 \log_{10}(1/\mathrm{MSE})$; values with MSE below 10^{-12} are capped at 99 dB. SSIM uses `skimage.metrics.structural_similarity` with `channel_axis=2`, `data_range=1.0`, and its default 7×7 window. First-visit PSNR is computed analogously between x_a^m and g_a over the same four pairs. The student has first-visit PSNR 19.99 dB and return PSNR 18.31 dB, a decrease of 1.68 dB. We additionally report self-SSIM, averaged between the generated outbound and return views (x_a^m, x_b^m) over the four pairs. Pixel metrics penalize misalignment from camera errors as well as appearance changes; they should not be interpreted as pure memory-module scores.

Camera Following. The same VGGT-Ω estimator (Wang et al., 2026a) is applied to all generated videos and to GT, requiring exactly 160 recovered poses. Recovered and prescribed camera-to-world poses are expressed relative to frame 0. We align translations with a scalar $s^* = \sum_t \mathbf{t}_t^\top \widetilde{\mathbf{t}}_t / \sum_t \|\widetilde{\mathbf{t}}_t\|_2^2$; no additional global rotation or translation is fitted. This scale-only convention preserves the shared initial-frame coordinate system and avoids unstable rotational alignment for nearly planar trajectories. Let $P_t = [R_t, \mathbf{t}_t; \mathbf{0}^\top, 1]$ and $\widetilde{P}_t$ denote the reference and scale-aligned recovered poses. With $\theta(R) = \frac{180}{\pi} \arccos(\mathrm{clip}((\mathrm{tr}\, R - 1)/2, -1, 1))$, we compute

$$\mathrm{RotErr} = \frac{1}{T} \sum_t \theta(R_t^\top \widetilde{R}_t), \qquad \mathrm{CamMC} = \frac{1}{T} \sum_t \|P_t - \widetilde{P}_t\|_F, \quad T = 160. \tag{10}$$

For consecutive frames, set $A_t = P_t^{-1} P_{t+1}$, $\widetilde{A}_t = \widetilde{P}_t^{-1} \widetilde{P}_{t+1}$, and $E_t = A_t^{-1} \widetilde{A}_t$. RPE rotation is the mean of $\theta(E_t[1{:}3, 1{:}3])$ and RPE translation is the mean of $\|E_t[1{:}3, 4]\|_2$ over $T - 1$ pairs (one-based matrix indices). Rotation uses no alignment; translation uses the same scalar alignment as the absolute metrics. After temporal averaging within each clip, all camera columns use a dataset-macro mean: average clips within each source dataset, then average the four dataset scores equally. GT's nonzero

Table 4: **Visual Quality on the Revisit Set.** All columns use a 0–100 scale; higher is better. VB-4 is the unweighted mean of Imaging, Aesthetic, Motion, and Flicker. Bold denotes the best result.

Method	Imaging	Aesthetic	Motion	Flicker	VB-4
GT (VAE-decoded)	74.00	53.84	98.27	94.37	80.12
Matrix-Game 3.0	74.03	50.66	98.42	94.59	79.42
HY-World 1.5	**76.29**	52.44	97.79	93.82	**80.09**
DreamX-AR	73.59	52.49	97.67	94.48	79.56
LingBot-Fast	74.13	**52.84**	97.08	94.62	79.67
SANA-WM	65.21	47.04	**98.86**	**95.40**	76.63
World-T^3 Stage 2	73.17	51.96	98.35	94.70	79.55
World-T^3 Stage 3	71.46	50.14	98.47	95.28	78.83

Table 5: **Complementary Consistency Measures on the Revisit Set.** $\bar{d}$ is the four-pair DINOv2 self-distance. d_{first} and d_{return} compare frames 8 and 152, respectively, to GT; Δ is their difference. Self-SSIM compares generated frames 8 and 152, while GT-SSIM compares the generated return frame to its reference. Bold marks the best generated result.

Method	$\bar{d}\downarrow$	$d_{\text{first}}\downarrow$	$d_{\text{return}}\downarrow$	$\Delta\downarrow$	Self-SSIM↑	GT-SSIM↑	LM↑
GT (VAE-decoded)	0.0018	–	–	–	0.9470	–	0.9756
Matrix-Game 3.0	0.1775	0.0411	0.2185	0.1774	0.3759	0.3748	0.8875
HY-World 1.5	0.1437	0.0274	0.1867	0.1593	0.3576	0.3526	**0.9874**
DreamX-AR	0.1232	0.0327	0.1610	0.1282	0.3805	0.3781	0.9838
LingBot-Fast	0.1420	0.0279	0.1832	0.1553	0.3431	0.3409	0.9703
SANA-WM	0.0800	0.0587	0.1218	0.0631	0.4704	0.4360	0.9006
World-T^3 Stage 2	0.0476	**0.0158**	0.0581	0.0423	0.5968	0.5313	0.9442
World-T^3 Stage 3	**0.0390**	0.0281	**0.0499**	**0.0218**	**0.7201**	**0.5795**	0.9258

errors in Table 1 characterize estimator error under this pipeline. Absolute errors reveal accumulated pose drift, whereas RPE measures consecutive-frame motion; neither substitutes for the other.

Visual Quality and Aggregation. We use the unmodified VBench 0.1.5 evaluator (Huang et al., 2024) in `custom_input` mode with its default temporal sampling and image preprocessing. Table 4 reports imaging quality, aesthetic quality, motion smoothness, and temporal flickering on a 0–100 scale, and their unweighted four-dimension mean (VB-4). All visual-quality scores are reconstructed as clip means from per-video results, rather than unweighted means of unequal evaluation shards. The main table instead averages seven dimensions: imaging quality, aesthetic quality, subject consistency, background consistency, motion smoothness, temporal flickering, and overall consistency, each on a 0–100 scale. For overall consistency, the text condition is the source-video caption. VB-4 uses the same scale but only the four dimensions listed above. The evaluator's imaging-quality preprocessing reduces the longest side to 512, and its aesthetic score uses a 224×224 center crop, so these scores do not fully describe native-resolution or boundary detail.

Local Temporal Consistency. We additionally report the WorldMark-inspired Local Memory score $\text{LM} = (1 - r_{\text{cut}})(1 - r_{\text{mut}})$. Hard cuts are detected with TransNetV2 at threshold 0.5; mutations are adjacent sampled-frame pairs with DINOv2 distance above 0.10. Sampling uses a fixed 0.5 s interval, excluding pairs within 0.5 s of a detected cut, rather than WorldMark's flow-based matching. LM evaluates short-term changes and does not establish whether a return view matches the intended scene.

D.3 Additional Results and Trade-offs

The student's revisit gains coexist with lower imaging and aesthetic scores than the teacher (Table 4). HY-World 1.5 has the highest generated VB-4 score, 80.09, while the student scores 78.83. The decoded reference scores 74.00 on imaging quality, below several generated methods, so this metric is not a calibrated distance to the real reference and GT is not a universal upper bound.

Table 6: **Complementary Consistency Measures on the Revisit Set.** DINOv2 and SSIM scores are averaged over the four outbound–return pairs within each clip, then across the 120 clips. $\bar{d}$ and Self-SSIM compare paired generated views. d_{first} and d_{return} compare the outbound and return views, respectively, with their corresponding GT views; $\Delta = d_{\text{return}} - d_{\text{first}}$. GT-SSIM compares return views with GT. LM measures local temporal consistency over each full clip. Bold marks the best generated result.

	Outbound-Return Consistency			GT-Referenced Fidelity		
Method	DINO $\bar{d}\downarrow$	SSIM $\uparrow$	LPIPS $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$
GT (VAE-decoded)	0.0018	0.9470	0.0221	–	–	–
Matrix-Game 3.0	0.1775	0.3759	0.4478	11.15	0.3748	0.4629
HY-World 1.5	0.1437	0.3576	0.4687	10.84	0.3526	0.4784
DreamX-AR	0.1232	0.3805	0.4619	11.66	0.3781	0.4699
LingBot-Fast	0.1420	0.3431	0.4775	10.59	0.3409	0.4911
SANA-WM	0.0800	0.4704	0.3805	13.52	0.4360	0.4272
World-T^3 Stage 2	0.0476	0.5968	0.1932	16.69	0.5313	0.2261
World-T^3 Stage 3	**0.0390**	**0.7201**	**0.1278**	**18.31**	**0.5795**	**0.1822**

Table 5 shows that the student and teacher rank first and second among generated methods on both DINOv2 self-consistency and GT-SSIM, averaged over the four revisit pairs; rankings among the external baselines vary by metric. Across these pairs, the student has a larger mean GT-referenced semantic distance than the teacher on the first visits (0.0281 vs. 0.0158), but a smaller one on the return visits (0.0499 vs. 0.0581). Conversely, HY-World 1.5 and DreamX-AR have higher LM than either of our models, and the teacher has higher LM than the student. Smaller adjacent-frame changes can raise LM without ensuring accurate long-range replay; these scores alone do not establish an over-smoothing mechanism. Using whole-frame resizing to 224×224 instead of the DINOv2 center crop retains the same leading three methods for revisit distance (our Stage 3 student model, Stage 2 teacher model, and SANA-WM).

Memory Retention during Rollout. We assess scene retention by comparing mean GT-referenced DINOv2 distances at the first and return visits across the four pairs. HY-World and our student have similar first-visit distances (both approximately 0.028), but their return distances rise to 0.1867 and 0.0499, respectively (Table 5). The smaller increase for our student indicates better retention of scene content across the evaluated revisit intervals. HY-World nevertheless achieves the highest VBench average among generated methods, exceeding our student's score (75.32 versus 74.52). These comparisons assess scene retention of the complete models and distinguish it from aggregate visual quality.

D.4 Complementary Metrics for Revisit Fidelity

Table 6 extends the main comparison in Table 1 to feature, perceptual, and pixel-level measures. Outbound-return consistency evaluates retention of generated content, while GT-referenced fidelity evaluates agreement with the reference scene. Our World-T^3 Stage 3 and Stage 2 models rank first and second, respectively, among generated methods under all six measures. Their advantage therefore persists across the evaluated distance metrics, although all remain sensitive to camera alignment.

D.5 Paired Statistical Comparisons

All pairwise tests use per-clip paired outcomes and an exact two-sided sign test. Means describe effect magnitudes, while these tests assess how consistently one method outperforms another across clips. Against SANA-WM, the student has lower DINOv2 revisit distance on $111/120$ clips ($p = 1.7 \times 10^{-23}$) and higher revisit PSNR on $112/120$ ($p = 1.4 \times 10^{-24}$); the teacher has lower DINOv2 distance on $108/120$ ($p = 1.8 \times 10^{-20}$). The student exceeds every external baseline on both Mem sim. and revisit PSNR on the same clip in $96/120$ cases. Its DINOv2 distance reduces the gap from SANA-WM to the GT reference distance of 0.0018 by approximately 52%.

For the seven-dimension VBench average, paired comparisons favor Matrix-Game and HY-World over the student; differences from DreamX-AR and LingBot-Fast are not significant. These outcomes complement the revisit comparison by documenting the visual-quality trade-off.

D.6 Camera-Sweep Analysis

The tiers in Table 2 contain 40 clips each, ordered by total yaw sweep divided by horizontal field of view. The grouping is post hoc and uses no model outputs. It measures camera sweep rather than rollout duration or the exact time that each scene region is out of view. Against SANA-WM, the student has lower DINOv2 distance on $37/40$, $36/40$, and $38/40$ clips in T1–T3, respectively ($p = 1.95 \times 10^{-8}$, 1.86×10^{-7}, and 1.49×10^{-9}, respectively). The largest mean difference occurs in T3, where the student's distance is lower by 0.0513.

On T3, the student's RotErr is $1.288°$, compared with $2.241°$ for the teacher, $4.316°$ for SANA-WM, and $11.075°$–$15.477°$ for the other baselines. These tier results average clips directly, unlike the dataset-macro camera averages in Table 1. For context, the student's main-table translational RPE of 0.00196 is 49.7% of the prescribed mean per-frame displacement (0.00395), so its small absolute magnitude should not be read as negligible.

D.7 Teacher–Student Comparisons

The student improves over the teacher on revisit PSNR in $91/120$ clips ($p = 1.2 \times 10^{-8}$) and on revisit LPIPS in $86/120$ ($p = 2.3 \times 10^{-6}$). Its lower mean DINOv2 revisit distance is accompanied by $72/120$ wins ($p = 0.0353$), significant at the 0.05 level. First-visit PSNR and all four camera metrics have p-values between 0.14 and 0.36; the absence of a significant difference is not an equivalence test. Revisit SSIM also improves descriptively from 0.5313 to 0.5795. The Local Memory score falls from 0.9442 to 0.9258, alongside the VBench and image-quality changes reported above.

E Inference Efficiency

Our 4-step World-T^3 student achieves 20.6 FPS at 1280×704 on a single NVIDIA B200 GPU with online memory updates enabled, as shown in Fig. 1b. The teacher uses 50 denoising steps per block and the student uses 4, a $12.5\times$ reduction in step count; we report measured speedup separately from this ratio. This section describes how the 4-step student reaches real-time generation on a single GPU. All runtime efficiency optimizations do not change the model architecture and preserve the numerical fidelity: the student weights, its 4-step schedule, the latent-frame context window size, and the TTT memory with online updates are unchanged. Appendix E.1 defines the workload and settings, Appendix E.2 introduces the optimizations, Appendix E.3 demonstrates their measured effect, and Appendix E.4 tests that the optimization remains numerically equivalent to the baseline. Appendix E.5 compares inference cost across the baselines of Table 1 on the same hardware.

E.1 Workload and Measurement Settings

Per-block workload. The World-T^3 student inherits the Wan2.2-TI2V-5B backbone (Wan et al., 2025): 30 DiT blocks of width 3072 with 24 attention heads, and a causal 3D VAE with $16\times$ spatial and $4\times$ temporal compression. At 1280×704, a latent frame contains 880 tokens. A generation block holds $B = 3$ latent frames (2,640 tokens) and decodes to 12 video frames, i.e., 0.75 s at 16 FPS. For every block, the implementation performs four steps: (i) it encodes the clean context, i.e., the persistent sink and the most recent clean blocks, at most 18 latent frames or 15,840 tokens, under the block-causal mask and caches the resulting keys and values; (ii) it runs the four denoising forwards of the target block, each attending to the cached context and to the block itself; (iii) it writes the block evicted from the window into the TTT memory (Eq. 4) and reads the memory at the target pose (Eq. 5); (iv) it decodes the finished block with the causal VAE decoder, carrying the decoder's temporal convolution cache from block to block. The context is re-encoded at every block rather than extended incrementally: when the window slides, each context block loses its oldest predecessor, so its hidden states change. Once the window is full, the per-block cost is therefore constant and does not depend on the rollout length. The fixed-size fast-weight and RLS states likewise keep memory storage constant (Sec. 3.2); this does not imply unlimited scene capacity.

Metrics. *End-to-end latency* is the wall-clock time from the inputs on disk (camera trajectory, first frame, and text prompt) to the last generated frame as 8-bit RGB in host memory. It includes first-frame VAE encoding, text encoding, camera embedding, sampling, TTT memory updates, decoding, and the device-to-host copy, and it excludes one-time model loading and video-container writing. We report it directly, as frames divided by latency (FPS), and as wall-clock seconds per second of generated video. *Peak memory* is the device memory in use, polled through NVML every 50–100 ms during the timed rollouts.

Settings. Every configuration runs in its own warm process on NVIDIA B200 GPUs (180 GB HBM3e): one discarded warm-up rollout absorbs compilation, kernel autotuning, and allocator growth, and we report the median of three timed rollouts. Runs execute one at a time on an 8-GPU node. We time two lengths: L5 (81 frames, 5.06 s), the training horizon, and L30 (465 frames, 29.06 s), where the window has been full for most of the rollout. Our model runs in default bf16 with PyTorch 2.15 (development build), CUDA 12.8, and cuDNN 9.18.

E.2 Optimizations

We group the optimizations by the source of the saving. Table 7 applies them cumulatively, in the order in which we added them.

Reusing Features across Denoising Steps and Blocks. The keys and values of the clean context are computed once per block and reused by all four denoising steps; this KV cache is part of the starting point. We extend reuse in three places. First, the TTT memory state $\mathcal{M}_b$ and the target pose key $\mathbf{k}_b$ do not change during the four steps, so the memory feature $\mathbf{m}_b$ of Eq. 5 is computed once per block instead of once per step; the output is bit-identical. Second, context keys are cached before rotary position embedding (RoPE) (Su et al., 2024) because their positions shift as the window slides; we rotate them once per block rather than at every step. Third, under the block-causal mask the keys and values of a context block depend only on that block and the context blocks before it. Blocks that form an unchanged prefix of the previous block's context therefore have unchanged keys and values and are not recomputed: all earlier context blocks while the window is still filling, and the sink afterwards.

Compilation and Attention Kernels. We compile the per-layer bodies of the denoising forward and of the context encoder with `torch.compile` (Ansel et al., 2024), which fuses the elementwise chains around attention (adaptive layer-norm modulation, gating, normalization) and keeps their intermediates in FP32. Self-attention in both passes uses FlashAttention-4 (Zadouri et al., 2026), which targets the Blackwell architecture, in place of FlexAttention (Dong et al., 2025); the block-causal context attention is issued as one dense call per query block over its visible key prefix, and both calls are registered as custom operators so that they remain inside the compiled graphs. The reference RoPE casts queries and keys to FP64 and multiplies them by complex128 frequencies; inside a compiled graph this runs as separate unfused kernels with FP64 temporaries. We apply the same rotation in real FP32 arithmetic, whose relative error (about 6×10^{-8}) is far below the bf16 resolution of the rotated tensors, so the rotation fuses with the preceding normalization.

VAE Decoding. We enable cuDNN 9 (Chetlur et al., 2014) convolutions, store decoder weights and activations in channels-last (NDHWC) layout, and let cuDNN select its fastest engine for each convolution shape by benchmarking all candidates, restricting the compiler's autotuning to cuDNN convolutions. The causal convolutions of the reference decoder also materialize a zero-padded copy of every input; we pass the spatial padding to the convolution itself and keep only the temporal padding, which comes from the causal cache. Decoding block by block with the causal cache performs the same convolutions as decoding the full latent sequence at once.

Host-Side Overhead. Two host-side operations left the GPU idle between kernels. The RoPE frequency tables were rebuilt on the CPU and copied from pageable host memory at every denoising step, five 16 MB transfers per block; we build them once per window shape and keep them on the device. The per-block Gaussian noise is drawn from a CPU generator for reproducibility, and each draw with its synchronous copy stalled the stream; a background thread now draws the noise of upcoming blocks from the same generator in the same order into pinned memory, so the transfer is asynchronous. Both changes leave the output bit-identical.

Table 7: **Cumulative Effect of the Optimizations on one B200.** Each row is a separate warm process; values are medians of three generations of a 30 s video (465 frames at 16 FPS). E2E is end-to-end latency, FPS is frames per second of latency, and Mem. is peak GPU memory in GB.

#	Change (cumulative)	E2E (s)	FPS	Mem.
1	Baseline	70.48	6.6	72.1
2	+ `torch.compile`	63.55	7.3	71.6
3	+ cuDNN 9, channels-last VAE decoder	37.23	12.5	52.8
4	+ FlashAttention-4 for self-attention	30.00	15.5	52.8
5	+ memory read once per block	29.55	15.7	52.8
6	+ RoPE of cached keys once per block	29.05	16.0	52.8
7	+ context prefix reuse	27.56	16.9	54.9
8	+ shape-aware conv. engines	24.25	19.2	54.4
9	+ prefetch Gaussian noise	23.83	19.5	53.5
10	+ FP32 Rotation	22.59	20.6	54.3

Figure 6: **Time per Steady-State Block on one GPU**, start (#1) vs. final (#10): CUDA-event timers over the 32 full-window blocks of a 30 s rollout.

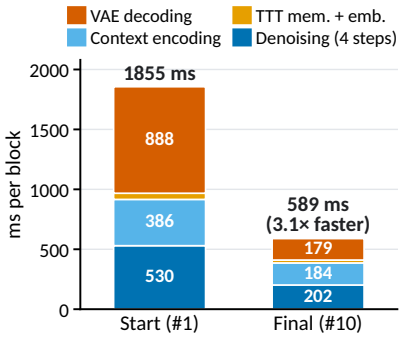

Table 8: **Numerical Fidelity of the Accelerated Implementation.** Operator-level errors use the captured inputs of block 5 of a held-out clip and an FP32 copy of the model with TF32 disabled; the decoder decodes the same 21 latent frames. Rollout rows compare full 81-frame bf16 rollouts with FP32 rollouts from the same inputs and seed.

Test	Measure	Reference bf16	Accelerated
Denoising forward	rel. RMS error to FP32 ↓	1.54×10^{-2}	8.46×10^{-3}
Context encoding	rel. RMS error to FP32 ↓	1.28×10^{-2}	8.25×10^{-3}
VAE decoder	rel. RMS error to FP32 ↓	7.17×10^{-3}	5.95×10^{-3}
	PSNR to FP32 (dB) ↑	53.82	55.44
Rollout	PSNR to FP32 rollout (dB) ↑	27.40	29.61
	SSIM to FP32 rollout ↑	0.9034	0.9257

Multiple GPUs. In the Pipelined Streaming Rollout, a second GPU decodes block b while the first samples block $b{+}1$; the decoder state stays on the second GPU, so the decoding is the same as in the serial schedule. For higher single-stream throughput, we additionally split the tokens of each forward across GPUs with Ulysses sequence parallelism (Jacobs et al., 2023), which exchanges them through all-to-all communication before and after attention, replay the per-layer loops as CUDA graphs (NVIDIA, 2025) to remove kernel-launch overhead, and split the decoder layer-wise into a two-stage pipeline over two further GPUs.

E.3 Results

Table 7 applies the optimizations cumulatively to a 30 s rollout on one GPU. End-to-end latency falls from 70.48 to 22.59 s (3.12×). Both stages contribute: sampling time per block falls from 943 to 391 ms and decoding time from 890 to 182 ms.

Figure 6 splits a steady-state block into its phases. In the starting point, decoding takes almost half of the block (888 of 1855 ms). After optimization, the four denoising forwards and the context encoding account for about two thirds of the block (386 of 589 ms), and decoding for 179 ms. One context encoding costs about as much as all four denoising forwards together (184 vs. 202 ms): with the sink reused, it still processes five times as many query tokens as a single forward (13,200 vs. 2,640). The remaining 24 ms include the TTT memory write, the timestep and camera embeddings, scheduling, and host-side work.

E.4 Numerical Fidelity

Most of the proposed optimizations in Table 7 produce bit-identical outputs. The remaining optimizations reorder floating-point operations (row 3, 8), select different kernels (row 4), change the precision of the RoPE rotation (row 10), so their outputs differ from the reference at the level of floating-point rounding. Comparing generated videos against the reference would not measure this

Table 9: **Inference Cost Comparison on NVIDIA B200 GPUs.** Metrics are median over three generations, run one at a time. L30 is the 30 s video settings. Mem. is peak memory in GB per GPU. For multi-GPU rows we use maximum peak memory over GPUs.

			E2E latency (s)		L30		
Method	Steps	GPUs	5s	30s	s / video s	FPS	Peak Mem (GB)
One GPU							
Matrix-Game 3.0	3	1	12.23	69.73	2.24	7.1	76
HY-World 1.5	4	1	7.96	107.04	3.59	4.5	110
DreamX-AR	4	1	7.23	42.22	1.45	11.0	76
LingBot-Fast	4	1	44.56	292.54	10.07	1.6	144
SANA-WM	4	1	6.69	32.75	1.13	14.2	48
World-T^3 Stage 3	4	1	3.63	22.59	0.78	20.6	54
Multiple GPUs							
Matrix-Game 3.0	3	8	6.50	41.95	1.35	11.8	61
HY-World 1.5	4	8	2.30	24.16	0.81	19.7	70
LingBot-Fast	4	8	16.72	100.32	3.45	4.6	107
World-T^3 Stage 3	4	8	1.79	8.28	0.28	53.2	47

difference: the autoregressive sampler amplifies rounding differences into a different but equally valid rollout. We therefore test at two levels (Table 8).

Operator Level. For each implementation, we capture the inputs of one denoising forward and one context encoding at a steady-state block (block 5) of a held-out clip, and compare its bf16 outputs with those of an FP32 copy of the model evaluated on FP32 copies of the same inputs, with TF32 disabled. The accelerated implementation passes if its relative RMS error to the FP32 output is at most 1.05 times that of the reference; this criterion was fixed before measurement. The decoder is tested analogously on the same latents: the eager bf16 decoder, the accelerated decoder, and an FP32 decoder decode the 21 latent frames of one clip. The accelerated implementation is closer to the FP32 result than the reference in all three cases, mainly because compiled kernels keep fused intermediates in FP32 where the reference rounds to bf16 after every operation.

Rollout Level. We generate complete 81-frame rollouts with an FP32 model and measure how closely each bf16 implementation reproduces them. The accelerated rollout is closer to the FP32 rollout than the reference bf16 rollout on 7 of 8 clips. For scale, the reference rollout with a different noise seed reaches only 17.90 dB against the FP32 rollout, so the differences between bf16 implementations are small compared with sampling variability. The sequence-parallel sampler with CUDA graphs, tested with three GPUs, and the layer-pipelined decoder pass the same operator-level criterion (error ratios 0.56 for the denoising forward, 0.62 for the context encoding, and a decoder error of 5.96×10^{-3} against 7.19×10^{-3} for the eager decoder). The replay-benchmark videos in Sec. 4 were generated with the evaluation implementation and were not regenerated with the accelerated one; the tests above address their equivalence directly.

E.5 Efficiency Comparison with Other Video World Models

Table 9 gives a more detailed measurement of all the baselines in Table 1 with both the single-GPU and multiple-GPU settings. Each baseline runs its official inference code with the configuration used for its quality results in Table 1. Because the models produce different numbers of frames per chunk, their clips differ slightly in length (between 464 and 497 frames for L30); seconds per second of video normalizes for this. On one GPU, our student is the only model that generates a 30 s rollout faster than real time. SANA-WM reports peak memory from the PyTorch allocator instead of the device, and runs its 4-step generator followed by a 3-step refiner. With multiple GPUs, HY-World 1.5 also becomes faster than real time, while our model remains the most efficient. Multi-GPU baselines use their official parallel modes, which is not supported by DreamX-AR and SANA-WM.

F LIMITATIONS

Although World-T^3 demonstrates robust persistent scene memory and interactive generation in real-time, several limitations remain to be addressed in future work.

Our framework relies on 3D foundation models, such as VGGT-Ω (Wang et al., 2026a), to extract 6-DoF camera trajectories from video corpora for conditioning. However, off-the-shelf pose estimators occasionally exhibit tracking noise or subtle drift, particularly in challenging visual environments with weak textures or rapid camera maneuvers. Since our test-time memory consolidation performs pose-keyed retrieval, such inaccuracies in the camera trajectory can degrade the alignment between queries and fast-weight states. Over extended generation horizons, such pose estimation errors may also accumulate and lead to suboptimal memory recall during viewpoint revisits. Addressing this limitation necessitates training on higher-quality camera trajectory datasets or incorporating more robust camera estimation pipelines.

Additionally, while our pose-indexed test-time training memory improves temporal stability and viewpoint revisit consistency over extended rollouts, maintaining pristine visual fidelity and generating well beyond the training context length remain open challenges in video world modeling. As autoregressive rollouts proceed significantly past the temporal window seen during training, video frames can occasionally experience gradual visual attenuation or fine-grained texture degradation due to compounding errors. Achieving photorealistic high-resolution generation and unbounded temporal extrapolation demands further advances in scalable generative architectures, higher-capacity latent spaces, and dynamic context management strategies.

Despite these limitations, World-T^3 provides a viable and efficient path toward persistent memory in interactive video world models, and we hope our TTT memory formulation inspires future explorations in real-time long-horizon scene generation.