# NARRA: SELF-ADAPTING VIDEO WORLD MODELS AT TEST TIME

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Long-horizon video world models often suffer from visual drift: as generation progresses, the appearance and geometry of a scene can gradually change, even when the camera returns to a previously observed viewpoint. We introduce NARRA, a self-adaptive framework that enables video world models to learn and remember at test time. NARRA uses an initial generation as self-supervised context to specialize the model to each generated world. Inspired by Nested Learning, adaptation occurs across interacting learning processes with different update frequencies, allowing new information to be incorporated without overwriting pretrained knowledge. The framework can further enhance video quality by leveraging channel-wise memory of earlier temporal segments and known camera trajectories that identify revisited viewpoints. We show that NARRA significantly improves video quality, with consistent gains across VBench and other evaluation metrics on the SANA-WM benchmark, and particularly strong improvements on challenging trajectories that require the model to return consistently to previously observed viewpoints.

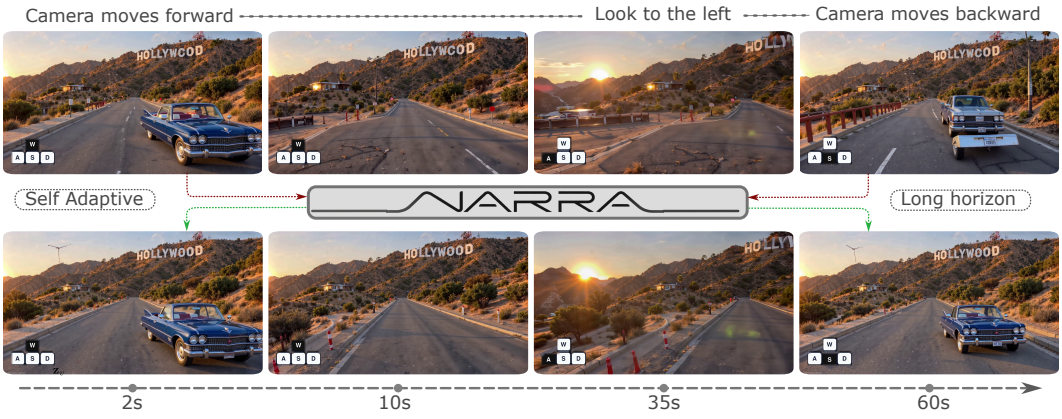



Figure 1: NARRA preserves scene identity across extended camera excursions. As the camera explores the highway, the context alters the car geometry and style (top), while self-adapting NARRA more closely retains the initial scene details (bottom).

## 1 INTRODUCTION

A useful world model should remember the world it creates. If a camera leaves a room, explores elsewhere, and returns tens of seconds later, the room should still contain the same objects, geometry, and visual details. Yet this seemingly basic requirement remains difficult for long-horizon video generation: when the camera returns to a previously observed location, the scene may no longer look the same (Xiao et al., 2025; Wu et al., 2025).

This challenge becomes increasingly important as video generation moves from short clips toward explorable visual worlds. Recent systems support minute-scale generation through block-wise diffusion, recurrent sequence modeling, causal generation, and efficient caching (Chen et al., 2026a;



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.

Dalal et al., 2025; Yang et al., 2026), while camera- and action-conditioned world models enable active exploration of generated environments (Zhu et al., 2026; Wang et al., 2026). These longer and more interactive trajectories increasingly require models to recover scene information that may have disappeared from view many seconds earlier.

Most existing approaches to long-horizon consistency focus on preserving access to the past: earlier frames, pose-indexed observations, landmarks, or historical features are stored and later retrieved to condition generation (Xiao et al., 2025; Team et al., 2026; Chen et al., 2026b; Xu et al., 2026). While effective, these approaches primarily use generated history as external context for future predictions. Yet as generation proceeds, the model accumulates increasingly rich evidence about the particular world it is simulating, its objects, appearance, layout, and geometry, while the generator itself remains largely unchanged. This suggests a complementary role for generated history: not only as memory to be retrieved, but as experience from which the model can learn.

*Can a video world model learn from what it creates?*

Test-time learning provides a natural mechanism for doing so. Rather than representing all past information exclusively through an activation state or external cache, part of the accumulated experience can be absorbed into model parameters. Test-time training has been used to represent sequence state through learnable parameter updates (Sun et al., 2025), while video-generation methods use test-time optimization for 3D-aware trajectory guidance (Zhang et al., 2026) and compositional text–layout alignment using a parameterized memory of prior optimization contexts (Qu et al., 2026). More broadly, neural-memory models study how observations can be written into parameterized memories at inference time (Behrouz et al., 2025b; 2026b). However, these approaches have not addressed using a world model's own generated experience to continuously specialize the generator for long-horizon scene persistence.

We introduce **NARRA**, a self-adaptive framework for long-horizon video world models built on SANA-WM (Zhu et al., 2026). NARRA turns an initially generated rollout into self-supervised experience and uses it to specialize the model to the particular generated world. Rather than conventional fine-tuning, NARRA uses a multi-timescale adaptation procedure inspired by Nested Learning (Behrouz et al., 2025a), allowing world-specific information to be incorporated through interacting fast and slow updates. We further meta-learn the starting point of this adaptation offline, explicitly preparing the model for rapid specialization at test time.

Parameter adaptation alone need not retain every detail encountered far in the past. NARRA therefore combines *parametric memory* with complementary mechanisms for accessing and preserving earlier visual evidence. A camera-trajectory return objective explicitly encourages agreement when the camera revisits previously observed locations, while channel-wise retrieval from aggregated recurrent segment states provides access to distant temporal context during final generation, building on ideas from Memory Caching (Behrouz et al., 2026a). Together, these mechanisms allow the model to adapt to the generated world while retaining information across camera excursions.

Our experiments show that this perspective is particularly useful when long-term memory matters most. On the evaluated SANA-WM benchmark, NARRA improves N-PSNR by $0.870$ dB and VBench by $1.015$ points over the frozen model, while also reducing MSE. The consistency gains become especially pronounced on long camera excursions: for trajectories spanning $8.1$–$9.6$ meters, NARRA improves PSNR by $1.08$ dB and N-PSNR by $1.46$ dB over frozen SANA-WM. We additionally introduce a revisit-focused benchmark and temporally stratified evaluation protocol designed to measure whether scene information survives prolonged absences from the active context.

Our main contributions are:

- **Self-adapting video world models.** We introduce NARRA, a framework in which a pretrained video world model uses its own generated rollout as self-supervision to specialize to the particular world being generated at test time.
- **Multi-timescale test-time adaptation.** We introduce a nested adaptation procedure that learns at multiple timescales, allowing the model to rapidly absorb information specific to the current video while more gradually adapting its underlying representation. We further meta-learn the starting point of this adaptation offline, so that the model is explicitly trained to adapt efficiently.

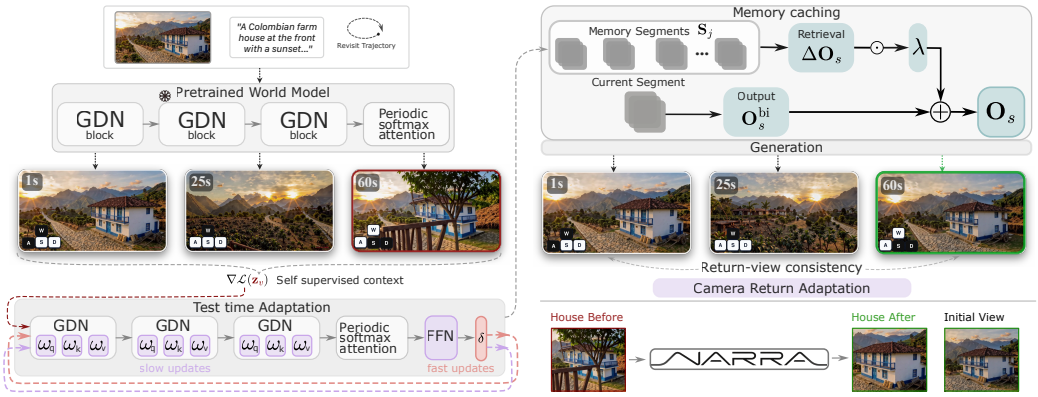



Figure 2: **Overview of NARRA**. Given an image, a text prompt, and a camera trajectory, a pretrained world model generates a latent video that serves as self-supervised context for test-time adaptation. Backbone: Gated DeltaNet (GDN) blocks with periodic softmax attention. NARRA couples slow updates of the GDN query, key, and value projections with fast updates of an additive parameter memory ($\delta$). The remaining pretrained parameters stay frozen. Camera-return adaptation: A return-view consistency encourages agreement between views captured when the camera revisits a location. Memory caching: Previous memory segments ($S_j$) provide a retrieved contribution ($\Delta O_s$), scaled by channel-wise ($\lambda$).

- **Long-horizon scene memory.** We combine parameter adaptation with two complementary mechanisms for preserving distant information: camera-return consistency explicitly constrains revisited viewpoints, while channel-wise segment-memory retrieval provides access to earlier temporal context during generation.
- **Evaluation of long-term world consistency.** We introduce a 45-video revisit-focused benchmark together with temporally stratified revisit metrics that explicitly evaluate consistency across extended camera excursions. On SANA-WM, NARRA improves both revisit consistency and overall video quality, with particularly strong consistency gains on long camera trajectories.

## 2 NARRA

NARRA uses an initially generated video as self-supervised context for adapting a pretrained world model. While NARRA can in principle be adapted to other video world models, we instantiate it on SANA-WM (Zhu et al., 2026) for its efficient minute-scale video generation and camera control. Narra combines three components for learning, preserving, and retrieving scene information: (i) *nested test-time adaptation*, which couples a fast parameter memory with slower updates to the recurrent representation and memory initialization; (ii) *camera-return consistency*, which anchors predictions at revisited viewpoints to earlier observations; and (iii) *channel-wise segment-memory retrieval*, which supplements final generation with independently computed summaries of earlier temporal segments. Figure 2 summarizes the complete pipeline.

### 2.1 Backbone: SANA-WM

Given an initial image $I$, a text condition $y$, and a camera trajectory $\mathcal{P}$, our goal is to generate a video that follows the prescribed motion while maintaining scene consistency. SANA-WM uses a two-stage generation pipeline. Its Stage 1 world model is a full-context bidirectional generator $G_\theta$ that produces long-horizon, camera-conditioned videos by combining frame-wise Gated DeltaNet (GDN) blocks with periodic softmax attention and a camera-conditioning mechanism. A Stage 2 refiner $R_\psi$ subsequently improves the generated video's visual quality and temporal consistency. NARRA adapts selected parameters of Stage 1 while keeping Stage 2 frozen.

Stage 1 learns to generate videos through flow matching in the latent space of a frozen encoder $E$. Given a video $\mathbf{X}$, SANA-WM first encodes it as $\mathbf{z}_0 = E(\mathbf{X})$. It then samples $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and a flow time $\tau \in [0, 1]$ to form the noisy latent $\mathbf{z}_\tau = (1 - \tau)\mathbf{z}_0 + \tau\boldsymbol{\epsilon}$. Conditioned on $\mathbf{z}_\tau$, the initial

image $I$, text $y$, and camera trajectory $\mathcal{P}$, the Stage 1 velocity field $v_\theta$ predicts the target direction $\boldsymbol{\epsilon} - \mathbf{z}_0$. SANA-WM trains this predictor using the flow-matching objective

$$\mathcal{L}_{\mathrm{FM}}(\theta; \mathbf{z}_0) = \|v_\theta(\mathbf{z}_\tau, \tau, I, y, \mathcal{P}) - (\boldsymbol{\epsilon} - \mathbf{z}_0)\|_2^2 . \tag{1}$$

### 2.2 Nested Test-Time Adaptation

We use the model's own generated rollout as supervision for test-time specialization. Starting from the pretrained generator $G_{\theta_0}$ and frozen refiner, we generate a video along the prescribed trajectory and combine it with the conditioning image at its initial pose to form the adaptation context $\mathcal{D}_v$. We encode this context once as $\mathbf{z}_v = E(\mathcal{D}_v)$.

A natural baseline directly fine-tunes the main-branch GDN query, key, value, and output projections, which we denote collectively by $\omega$. Using the same flow-matching objective as SANA-WM, this gives

$$\omega^{k+1} = \omega^k - \eta_\omega \nabla_{\omega^k} \mathcal{L}_{\mathrm{FM}}(\omega^k; \mathbf{z}_v). \tag{2}$$

NARRA instead separates adaptation across two timescales. We introduce a *fast parameter memory* $\delta$, implemented as an additive update to the final block's feed-forward projection weight, and let it rapidly absorb video-specific information. At a slower timescale, we adapt the recurrent projections $\omega$ and the initialization $\phi$ from which this fast memory starts. Inspired by the Continuum Memory System (CMS) in Nested Learning (Behrouz et al., 2025a), the slow update backpropagates through the fast updates. In this way, NARRA adapts not only the model representation, but also how the model acquires world-specific memory.

We initialize $\omega^0 = \omega_0$ and $\phi^0 = \Phi^\star$, where $\Phi^\star$ is learned offline. At slow iteration $k$, we initialize the fast memory as $\delta^{k,0} = \phi^k$ and, with $\omega^k$ fixed, perform $R$ fast updates:

$$\delta^{k,r+1} = \delta^{k,r} - \alpha \nabla_{\delta^{k,r}} \mathcal{L}_{\mathrm{FM}}(\omega^k, \delta^{k,r}; \mathbf{z}_v), \qquad r = 0, \ldots, R-1. \tag{3}$$

After these $R$ updates, we backpropagate through the resulting fast-adaptation trajectory to update both the recurrent representation and the memory initialization:

$$\begin{aligned} \omega^{k+1} &= \omega^k - \eta_\omega \nabla_{\omega^k} \mathcal{L}_{\mathrm{FM}}(\omega^k, \delta^{k,R}; \mathbf{z}_v), \\ \phi^{k+1} &= \phi^k - \eta_\phi \nabla_{\phi^k} \mathcal{L}_{\mathrm{FM}}(\omega^k, \delta^{k,R}; \mathbf{z}_v). \end{aligned} \tag{4}$$

The gradients differentiate through all $R$ fast updates within the current slow iteration, but not across slow iterations. Thus, $\omega$ learns a representation that supports rapid video-specific adaptation, while $\phi$ learns where that adaptation should begin. After $K$ slow iterations, we perform one final sequence of $R$ fast updates from $\phi^K$ with $\omega^K$ fixed, and use the resulting $\omega^K$ and $\delta^{K,R}$ for generation.

**Meta-learned initialization.** We learn the initial fast-memory state $\Phi^\star$ offline so that test-time adaptation begins from a parameter state already optimized for rapid specialization. For each training example, we generate a rollout, perform the fast updates on that rollout, and minimize the adapted model's flow-matching loss on the corresponding source video with conditions $(I_n, y_n, \mathcal{P}_n)$. We optimize only $\Phi^\star$, backpropagating through the fast updates while keeping the recurrent projections fixed at $\omega_0$.

### 2.3 Camera-Return Consistency

Flow matching adapts the model to the generated context, but does not explicitly enforce consistency when the camera revisits a previously observed viewpoint. We use the known camera trajectory to identify such returns and regularize the corresponding latent predictions, directly encouraging agreement between the returning prediction and the matched earlier observation.

Let $\mathcal{H}_v$ denote the set of return pairs $(a, b)$, with $a < b$, whose temporal separation exceeds a minimum gap and whose relative camera translation and rotation fall below prescribed thresholds. When several earlier frames qualify for the same returning frame $b$, we select the frame $a$ that minimizes the sum of the translation and rotation differences, with each difference divided by its corresponding matching threshold.

At iteration $k$, the predicted clean latent is $\widehat{\mathbf{z}}_0^k = \mathbf{z}_\tau - \tau v_{\omega^k,\delta^{k,R}}(\mathbf{z}_\tau, \tau, I, y, \mathcal{P})$, where $\mathbf{z}_\tau = (1-\tau)\mathbf{z}_v + \tau\boldsymbol{\epsilon}$. For each return pair $(a, b) \in \mathcal{H}_v$, we compare the predicted latent at the returning frame $b$ with the observed latent at its matched earlier frame $a$. This gives

$$\mathcal{L}_{\text{cam}}^{(v,k)} = \frac{1}{|\mathcal{H}_v|} \sum_{(a,b)\in\mathcal{H}_v} d_{\text{ret}}\left(\widehat{\mathbf{z}}_{0,b}^k, \mathbf{z}_{v,a}\right), \tag{5}$$

where $d_{\text{ret}}$ compares spatially pooled latent representations. In practice, we compute feature agreement over spatial groups and aggregate the resulting similarities according to their relative spatial support. We set $\mathcal{L}_{\text{cam}}^{(v,k)} = 0$ when no qualifying return pairs are found.

Throughout adaptation, we incorporate return consistency by replacing $\mathcal{L}_{\text{FM}}$ in both the fast and slow updates, Equations 3 and 4, with

$$\mathcal{L}_{\text{FM}} + \gamma_c \mathcal{L}_{\text{cam}}^{(v,k)}, \tag{6}$$

where $\gamma_c$ controls the strength of the return-consistency constraint.

### 2.4 Channel-Wise Segment Memory Retrieval

Camera-return consistency acts only when the trajectory identifies an approximate revisit, yet related scene content can recur across temporally distant segments without the entire viewpoint matching an earlier frame. We therefore complement parameter adaptation with channel-wise cross-segment memory retrieval, allowing later segments to selectively reuse information stored from earlier parts of the video. The channel-wise modulation lets the model emphasize different components of the retrieved scene representation, supporting consistency of localized details, objects, appearance, and other recurring content across disconnected frames. Following Memory Caching (Behrouz et al., 2026a), we summarize each temporal segment independently and retrieve from these summaries during later generation.

At each denoising step, we run an independent GDN forward recurrence over each temporal segment $j$, initializing the recurrence from zero. This produces a matrix state $\mathbf{S}_j \in \mathbb{R}^{d_v \times d_k}$ that summarizes key–value associations and a normalization state $\mathbf{d}_j \in \mathbb{R}^{d_k}$ that tracks key-side normalization, using the same gated delta updates as the backbone. Here, $d_k$ and $d_v$ denote the per-head key and value dimensions.

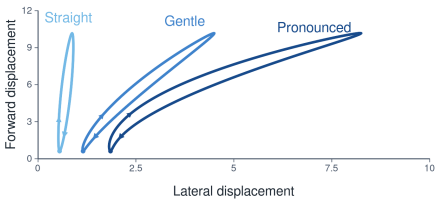
For a current segment $s > 1$, we aggregate the memory states from all preceding segments as $\overline{\mathbf{S}}_s = (s-1)^{-1}\sum_{j=1}^{s-1}\mathbf{S}_j$ and $\overline{\mathbf{d}}_s = (s-1)^{-1}\sum_{j=1}^{s-1}\mathbf{d}_j$. The queries of segment $s$ then retrieve relevant information from this cross-segment memory:

$$\Delta\mathbf{O}_s = \frac{\overline{\mathbf{S}}_s \widehat{\mathbf{Q}}_s^{\text{r}}}{\max\left(\overline{\mathbf{d}}_s^\top \widehat{\mathbf{Q}}_s, \varepsilon\right)}, \qquad \mathbf{O}_s = \mathbf{O}_s^{\text{bi}} + \boldsymbol{\lambda} \odot \Delta\mathbf{O}_s. \tag{7}$$

Here, $\widehat{\mathbf{Q}}_s \in \mathbb{R}^{d_k \times N_s}$ contains the backbone-normalized queries for the $N_s$ tokens in segment $s$, and $\widehat{\mathbf{Q}}_s^{\text{r}}$ denotes their rotary-position-transformed counterparts. The denominator normalizes the retrieved response token-wise, with $\varepsilon$ for numerical stability, while $\mathbf{O}_s^{\text{bi}}$ denotes the original bidirectional GDN output.

The vector $\boldsymbol{\lambda} \in \mathbb{R}^{d_v}$ controls how strongly each output channel incorporates the retrieved memory. Together, the token-dependent queries and channel-wise scaling make the memory response selective across both the current spatial tokens and feature channels, allowing relevant scene information to be reused even when only part of the current view corresponds to earlier content. We choose the entries of $\boldsymbol{\lambda}$ to increase linearly across channels and normalize them to have mean $\lambda$, which sets the overall retrieval strength. These weights remain fixed during generation, while the retrieved content varies with the current queries. We study alternative retrieval strengths and weighting schemes in our ablation studies.

During final generation, NARRA combines this cross-segment memory with the adapted parameters $\omega^K$ and $\delta^{K,R}$. We add the retrieved response before the GDN output gate and projection and




| Scene layout | Straight | Gentle | Pronounced | Total |
|---|---|---|---|---|
| Enclosed spaces | 7 | 5 | 3 | 15 |
| Open built environments | 5 | 5 | 5 | 15 |
| Natural landscapes | 6 | 4 | 5 | 15 |
| **Total** | **18** | **14** | **13** | **45** |

Figure 3: Composition of the custom test set. Left: representative camera paths for the three trajectory classes. Right: scenes grouped by spatial layout.

recompute the segment states at each denoising step. The first segment uses only the original bidirectional response because no earlier segment is available. The frozen Stage 2 refiner then processes the completed Stage 1 video.

## 3 EXPERIMENTS

We evaluate NARRA along three complementary dimensions. First, we introduce the NARRA Benchmark, a new evaluation 45-video set designed to measure long-horizon return consistency using departure-and-return camera trajectories. Second, we compare NARRA against other world-model pipelines in a controlled cross-pipeline evaluation. Third, we evaluate NARRA and its individual components on the 50-video SANA-WM benchmark, including analyses by trajectory difficulty and camera-travel distance. Across these evaluations, we report standard frame consistency, temporally stratified revisit metrics, perceptual video quality, and camera-trajectory accuracy. Additional results are available on the project website (nar, 2026a) and in the anonymous supplementary repository (nar, 2026b).

**Evaluation Data.** We use two complementary evaluation sets. Our primary evaluation uses the **NARRA Benchmark**, a new 45-scene test set designed to assess consistency after extended camera excursions, which are less represented in SANA-WM. Conditioning images are generated using *GPT* and *Nano Banana Pro*, following SANA-WM. As shown in Figure 3, the benchmark contains 15 scenes from each of three layouts with different depth and occlusion characteristics: *enclosed spaces*, including interiors, courtyards, walled gardens, and narrow streets, with limited depth and rapidly changing occlusions; *open built environments*, including roads, villages, waterfronts, and terraces inspired by real-world locations, combining nearby structures with mid-range depth; and *natural landscapes*, dominated by terrain and distant horizons. Each layout includes all three trajectory families.

Each scene follows a 60-second trajectory that leaves the conditioning viewpoint until the initial region is no longer visible, explores another region, and retraces its path to the initial pose. We use straight, gentle-curve, and pronounced-curve returns, distinguished by lateral curvature (Figure 3). Maximum displacement averages 9.60 m, ranging from 3.65 to 20.00 m. Revisit pairs span 26–60 s, with mean and median separations of 43.0 s; 88.9% span at least 30 s. All methods share the conditioning image, caption, camera trajectory, and sampling configuration.

For comparison with world models designed for shorter rollouts, we also evaluate a 16-second version using the same scenes, images, and captions. Each trajectory is uniformly resampled to preserve the departure-and-return path, then converted to each method's native action or pose interface where needed. We additionally use a 50-video subset of SANA-WM to evaluate broader trajectory coverage and individual NARRA components. It contains 25 simple and 25 hard trajectories, with 12 game-style, 12 indoor, 14 outdoor-city, and 12 outdoor-nature scenes, and both difficulty levels represented in every category.

**Evaluation Metrics.** The SANA-WM protocol averages PSNR, SSIM, and MSE over the five geometrically closest revisit pairs. However, in these pairs, 90.4% have both endpoints within 1 s of another selected pair, concentrating evaluation around similar return events. We introduce temporally stratified N-PSNR (NARRA-PSNR) and N-SSIM to cover different time horizons.

Candidates are sampled every two frames, requiring translation differences $\leq 1$ m, viewing-direction differences $\leq 10°$, and temporal gaps $\geq 3$ s. We select up to five geometrically closest pairs in each bin—$[3, 10)$, $[10, 20)$, $[20, 30)$, $[30, 40)$, $[40, 50)$, and $[50, \infty)$ s—with corresponding endpoints spaced at least 1 s apart. Scores are averaged within bins, across occupied bins, and across scenes, using identical pairs for all methods. We also report VBench Overall for general video quality.

### 3.1 Results on the NARRA Benchmark

Table 1: Long-horizon results on NARRA Benchmark.

| Method | PSNR↑ | SSIM↑ | MSE↓ | N-PSNR↑ | N-SSIM↑ | N-MSE↓ | VBench↑ |
|---|---|---|---|---|---|---|---|
| SANA-WM | 15.498 | 0.3617 | 2117.8 | 16.209 | 0.3990 | 1931.3 | 80.521 |
| **NARRA** | **16.331** | **0.3806** | **1694.0** | **17.157** | **0.4149** | **1521.6** | **81.920** |

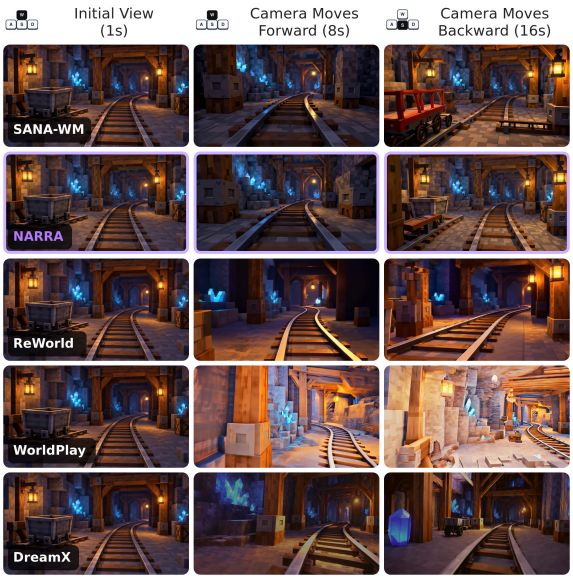



Figure 4: **Cross-pipeline return consistency on the 16-second NARRA Benchmark.** Initial, forward, and returning views. NARRA preserves the original minecart's identity and location and maintains a consistent rail layout. Competing pipelines remove, relocate, or alter the minecart and change the rail geometry.

Table 1 reports the paired comparison under identical evaluation conditions. NARRA improves over frozen SANA-WM across all reported metrics in the 60-second evaluation. It improves PSNR by 0.83 dB and N-PSNR by 0.95 dB, reduces MSE by 20.0% and N-MSE by 21.2% and VBench (+1.40). The gains are consistent at the scene level: NARRA achieves higher PSNR on 34 of the 45 scenes, higher N-PSNR on 39, and higher VBench on 34. The comparable gains in the standard and revisit-specific metrics, indicate that NARRA improves both overall frame consistency and the recovery of previously observed scene content after prolonged camera excursions.

**60-Second Long-Horizon Evaluation.** We first evaluate frozen SANA-WM Zhu et al. (2026) and NARRA on the complete 60-second NARRA Benchmark. Both generate minute-long videos from the same conditioning images, captions, sampling configuration, and trajectories. The benchmark contains 18 straight, 14 gentle-curve, and 13 pronounced-curve trajectories, distributed across the three scene layouts as illustrated in Figure 3. We measure overall frame consistency using PSNR, SSIM, and MSE; returning-view consistency with N-PSNR, N-SSIM, and N-MSE; and perceptual video quality using VBench.

Table 1 reports the paired comparison under identical evaluation conditions. NARRA improves over frozen SANA-WM across all reported metrics in the 60-second evaluation. It improves PSNR by 0.83 dB and N-PSNR by 0.95 dB, reduces MSE by 20.0% and N-MSE by 21.2%, and also improves SSIM (+0.019), N-SSIM (+0.016), and VBench **(+1.40)**. The gains are consistent at the scene level: NARRA achieves higher PSNR on 34 of the 45 scenes, higher N-PSNR on 39, and higher VBench on 34. The comparable gains in the standard and revisit-specific metrics, with a slightly larger improvement on the revisit-specific N-PSNR, indicate that NARRA improves both overall frame consistency quality and the recovery of previously observed scene content after prolonged camera excursions. Representative example videos are available on the project website sana-wm-60s

We also attempted to evaluate *ReWorld* Chen et al. (2026b), *WorldPlay* Sun et al. (2026), and *DreamX-World* Team et al. (2026) on the complete 60-second trajectories. At this horizon, their outputs frequently exhibited scene drift, scene replacement, or failure to follow the prescribed return path, preventing a meaningful comparison between the initial and returning views. We therefore omit their 60-second numerical results. Appendix A.4 describes these experiments.

**16-Second Comparison with Other World Models.** To enable a meaningful comparison with other world models, we additionally evaluate all five methods on shortened 16-second versions of the same departure-and-return trajectories. At this shorter horizon, the camera spends less time

Table 2: 16-second evaluation on the 45-video custom test set. All methods use the same conditioning images, captions, and camera trajectories.

| Method | PSNR↑ | SSIM↑ | MSE↓ | N-PSNR↑ | N-SSIM↑ | VBench↑ | RotErr$^{\circ}$ ↓ | TransErr↓ | CamMC↓ |
|---|---|---|---|---|---|---|---|---|---|
| SANA-WM Zhu et al. (2026) | 16.69 | **0.423** | 1543.5 | 17.03 | **0.428** | **85.80** | **0.84** | 0.539 | 0.540 |
| ReWorld Chen et al. (2026b) | 12.64 | 0.301 | 3814.9 | 12.79 | 0.308 | 84.32 | 2.33 | 2.816 | 2.817 |
| WorldPlay Sun et al. (2026) | 12.67 | 0.264 | 3924.8 | 13.03 | 0.271 | 84.40 | 3.65 | **0.441** | **0.455** |
| DreamX Team et al. (2026) | 12.98 | 0.341 | 3575.8 | 13.14 | 0.349 | 82.99 | 3.14 | 0.934 | 0.946 |
| **NARRA (ours)** | **16.72** | 0.421 | **1525.2** | **17.04** | 0.427 | 85.70 | 0.96 | 0.622 | 0.623 |

away from the initial region, making the return-consistency task substantially easier; in particular, the returning views generated by the competing methods remain sufficiently recognizable for quantitative comparison. As shown in Table 2, NARRA achieves the best performance across the majority of reported frame consistency and revisit metrics. Frozen SANA-WM obtains a slightly higher VBench score, and competitive SSIM, while NARRA better recovers the initial scene with comparable perceptual video quality. Comparing this with the 60-second results in Table 1 further show that NARRA's gains become more pronounced as the horizon increases and long-term scene memory becomes more challenging.

Figure 4 qualitatively compares the models. When the camera returns, NARRA preserves the original minecart and its location, whereas the other models remove, relocate, or substantially alter it. NARRA also maintains the curved rail layout across the initial, outbound, and returning views, while in the competing generations, the rails change direction, split, or are reconstructed with a different geometry. This shows that the metric gains correspond to improved preservation of both object identity and persistent scene structure. More visual examples are on the project website baselines-16s.

### 3.2 Results on the SANA-WM Benchmark

Table 3 compares frozen SANA-WM with the test-time adaptation variants across simple and hard trajectories. Both Narra basic TTTand Narra nested CMSimprove PSNR, N-PSNR, and VBench over the frozen baseline. Narra nested CMSfurther improves over Narra basic TTT, reducing the overall MSE from 2385.12 to 2266.41. These gains appear across both difficulty splits, supporting the benefit of coupling recurrent adaptation with a fast parameter memory.

Combining nested adaptation with camera-return consistency and segment-memory retrieval yields the strongest overall PSNR, MSE, N-PSNR, and VBench among the evaluated configurations. Relative to frozen SANA-WM, NARRA improves PSNR by 0.572 dB, N-PSNR by 0.870 dB, and VBench by 1.015 points, while reducing MSE by 12.33%. The camera-only variant achieves the highest PSNR and SSIM on hard trajectories.

**Trajectory difficulty.** NARRA improves PSNR, MSE, N-PSNR, N-SSIM, and VBench over frozen SANA-WM on both difficulty splits. On simple trajectories, it improves PSNR by 0.471 dB, N-PSNR by 0.919 dB, and VBench by 1.378 points. On hard trajectories, the corresponding gains are 0.674 dB, 0.821 dB, and 0.653 points, respectively. MSE decreases by 8.47% on simple trajectories and 15.02% on hard trajectories. Standard SSIM increases on hard trajectories but decreases slightly on simple trajectories, whereas N-SSIM improves on both.

The effects of the individual components of NARRA are not uniform across metrics. Adding segment-memory retrieval to Narra nested CMSimproves overall PSNR and MSE, but decreases VBench. Camera-return consistency achieves the highest N-SSIM on both difficulty splits and improves overall N-PSNR and VBench. On hard trajectories, the memory-only variant achieves the best PSNR, SSIM, and MSE, exceeding the full NARRA variant by only 0.007 dB in PSNR. NARRA nevertheless achieves the highest N-PSNR and VBench on both splits, together with the strongest overall PSNR and MSE.

**Qualitative consistency.** Figure 5 illustrates how scene appearance changes after the camera explores another region and returns. In the outdoor example, frozen SANA-WM alters the car's rear geometry and taillights, as well as roof details on the background house. Narra nested CMSand NARRA preserve these structures more consistently. In the cafe example, the frozen model changes the man's shirt, the menu board, and the appearance of the background couple. NARRA better preserves these details across the camera excursion, although some fine-scale variations remain.

Table 3: Results on the 50-video SANA-WM benchmark. N-PSNR, N-SSIM, and N-LPIPS use our temporally stratified revisit protocol. All metrics are evaluated over the complete 50-video benchmark.

| Method | PSNR ↑ | SSIM ↑ | MSE ↓ | N-PSNR ↑ | N-SSIM ↑ | VBench ↑ | N-LPIPS ↓ | Rot. (°) ↓ | Trans. ↓ | Cam-MC ↓ |
|---|---|---|---|---|---|---|---|---|---|---|
| **Simple trajectories** | | | | | | | | | | |
| SANA-WM Zhu et al. (2026) | 15.381 | **0.3593** | 2094.10 | 13.924 | 0.2970 | 79.700 | 0.5353 | 3.547 | 1.310 | 1.322 |
| Narra basic TTT | 15.514 | 0.3456 | 2071.58 | 14.545 | 0.3044 | 80.692 | 0.5182 | 3.616 | 1.438 | 1.446 |
| Narra nested CMS | 15.763 | 0.3489 | 1942.55 | 14.632 | 0.3041 | 80.728 | 0.5124 | 3.371 | 1.501 | 1.508 |
| Narra nested CMS+ Memory | 15.780 | 0.3496 | 1935.12 | 14.610 | 0.3034 | 80.467 | 0.5158 | 3.476 | 1.513 | 1.520 |
| Narra nested CMS+ Camera | 15.775 | 0.3502 | 1959.44 | 14.721 | **0.3081** | 81.025 | 0.5098 | **3.102** | 1.249 | 1.255 |
| **NARRA** | **15.852** | 0.3458 | **1916.78** | **14.843** | 0.3068 | **81.078** | **0.5094** | 3.293 | **1.246** | **1.253** |
| **Hard trajectories** | | | | | | | | | | |
| SANA-WM | 13.967 | 0.2667 | 3013.61 | 13.035 | 0.2415 | 82.535 | 0.5462 | **4.733** | **1.409** | **1.437** |
| Narra basic TTT | 14.419 | 0.2686 | 2698.67 | 13.621 | 0.2466 | 82.924 | 0.5312 | 7.577 | 1.458 | 1.523 |
| Narra nested CMS | 14.571 | 0.2704 | 2590.26 | 13.736 | 0.2471 | 83.001 | 0.5260 | 7.320 | 1.559 | 1.620 |
| Narra nested CMS+ Memory | **14.648** | **0.2740** | **2545.97** | 13.761 | 0.2466 | 82.686 | 0.5254 | 7.976 | 1.550 | 1.618 |
| Narra nested CMS+ Camera | 14.520 | 0.2735 | 2652.86 | 13.733 | **0.2511** | 82.821 | **0.5240** | 7.413 | 1.539 | 1.597 |
| **NARRA** | 14.641 | 0.2726 | 2561.00 | **13.856** | 0.2488 | **83.188** | 0.5249 | 8.284 | 1.618 | 1.688 |
| **Average** | | | | | | | | | | |
| SANA-WM | 14.674 | **0.3130** | 2553.86 | 13.480 | 0.2693 | 81.118 | 0.5408 | **4.140** | **1.359** | **1.380** |
| Narra basic TTT | 14.967 | 0.3071 | 2385.12 | 14.083 | 0.2755 | 81.808 | 0.5247 | 5.597 | 1.448 | 1.484 |
| Narra nested CMS | 15.167 | 0.3096 | 2266.41 | 14.184 | 0.2756 | 81.865 | 0.5192 | 5.346 | 1.530 | 1.564 |
| Narra nested CMS+ Memory | 15.214 | 0.3118 | 2240.54 | 14.186 | 0.2750 | 81.576 | 0.5206 | 5.726 | 1.531 | 1.569 |
| Narra nested CMS+ Camera | 15.148 | 0.3118 | 2306.15 | 14.227 | **0.2796** | 81.923 | **0.5169** | 5.258 | 1.394 | 1.426 |
| **NARRA** | **15.246** | 0.3092 | **2238.89** | **14.350** | 0.2778 | **82.133** | 0.5172 | 5.788 | 1.432 | 1.470 |




Figure 5: Qualitative revisit consistency. Initial, intermediate, and returning views are shown alongside enlarged details. SANA-WM alters the car's rear and background roof details in the outdoor scene, and the barista's shirt, menu board, and background people in the cafe. The adapted configurations preserve these features more consistently.

## 4 Conclusion

We introduced **NARRA**, which uses self-generated experience for test-time specialization through nested adaptation, camera-return consistency, and channel-wise memory retrieval. Experiments show improved long-horizon revisit consistency and video quality, supporting generated history as both context and a learning signal. Future directions include interleaving adaptation with generation, selecting reliable self-supervision to avoid reinforcing generation errors, and learning when to update parameters versus retrieve earlier context under limited compute and memory budgets.

## AI USE STATEMENT

In this work, we used generative AI tools to assist with coding and for generating synthetic data. In particular, generative AI was used to support implementation and debugging during development. We also used *GPT* and *Nano Banana Pro* to generate the conditioning images for the NARRA Benchmark, as described in Section 3. We did not use generative AI for other required-disclosure activities, and the remaining disclosure categories are not applicable to this work.

Additionally, we used generative AI tools to aid and polish the writing of the paper. All AI-assisted code and text were reviewed by the authors, and the technical claims, experimental procedures, results, and final presentation were verified and determined by the authors. We take responsibility for the final content of this work, including text, claims, code, and artifacts produced with the aid of generative AI.

## ETHICS STATEMENT

This work does not involve human-subject studies or the collection of sensitive personal data. The conditioning images used in the NARRA Benchmark are synthetically generated using *GPT* and *Nano Banana Pro*, as described in Section 3. Our work focuses on improving long-horizon consistency in generative video world models. As with generative video technologies more broadly, such models may potentially be used to produce misleading or synthetic visual content. We present NARRA for research purposes and report the use of generated data and AI-assisted tools transparently.

## REPRODUCIBILITY STATEMENT

We provide the information needed to reproduce our method and evaluation throughout the paper. Section 2 specifies the NARRA architecture, test-time adaptation procedure, camera-return objective, and segment-memory retrieval mechanism. Section 3 describes the evaluation datasets, construction of the NARRA Benchmark, evaluation metrics, and comparison protocol, while Section 4 reports controlled ablations of the main design choices. We have also made our implementation and evaluation code, together with representative generated videos, available to facilitate reproduction and qualitative inspection of the reported results.

## REFERENCES


NARRA project website. https://narrawm.netlify.app/, 2026a. Anonymous project page.

NARRA: Supplementary repository. https://anonymous.4open.science/r/NARRA-30CC/README.md, 2026b. Anonymous supplementary materials.

Ali Behrouz, Meisam Razaviyayn, Peilin Zhong, and Vahab Mirrokni. Nested learning: The illusion of deep learning architectures. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 46968–47002. Curran Associates, Inc., 2025a. doi: 10.52202/085713-1565.

Ali Behrouz, Peilin Zhong, and Vahab Mirrokni. Titans: Learning to memorize at test time. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 113506–113543. Curran Associates, Inc., 2025b. doi: 10.52202/085713-3786.

Ali Behrouz, Zeman Li, Yuan Deng, Peilin Zhong, Meisam Razaviyayn, and Vahab Mirrokni. Memory caching: RNNs with growing memory. In *Forty-third International Conference on Machine Learning*, 2026a.

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. In *Forty-third International Conference on Machine Learning*, 2026b.

Ali Behrouz, Meisam Razaviyayn, Peilin Zhong, and Vahab Mirrokni. It's all connected: A journey through test-time memorization, attentional bias, retention, and online optimization. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 131306–131333, 2026c.

Guibin Chen, Dixuan Lin, Jiangping Yang, Chunze Lin, Junchen Zhu, Mingyuan Fan, Hao Zhang, Sheng Chen, Zheng Chen, Chengcheng Ma, Weiming Xiong, Wei Wang, Nuo Pang, Kang Kang, Zhiheng Xu, Yuzhe Jin, Yupeng Liang, Yubing Song, Peng Zhao, Boyuan Xu, Di Qiu, Debang Li, Zhengcong Fei, Yang Li, and Yahui Zhou. Skyreels-v2: Infinite-length film generative model, 2025. URL https://arxiv.org/abs/2504.13074.

Junsong Chen, Yuyang Zhao, Jincheng YU, Ruihang Chu, Junyu Chen, Shuai Yang, Xianbang Wang, Yicheng Pan, Zhou Daquan, Huan Ling, Haozhe Liu, Hongwei Yi, Hao Zhang, Muyang Li, Yukang Chen, Han Cai, Sanja Fidler, Ping Luo, Song Han, and Enze Xie. Sana-video: Efficient video generation with block linear diffusion transformer. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 38973–38997, 2026a.

Zhifei Chen, Luozhou Wang, Guibao Shen, Dongyu Yan, Shuai Yang, Tianshuo Xu, Yihua Du, Wei Wang, Tianyi Gui, Lianghua Huang, and Yingcong Chen. Reworld: An interactive world model with long-horizon memory, 2026b. URL https://arxiv.org/abs/2608.23565.

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, 2025.

Jusen Du, Weigao Sun, Disen Lan, Jiaxi Hu, Zhang Tao, and Yu Cheng. Mom: Linear sequence modeling with mixture-of-memories. In *International Conference on Learning Representations*, volume 2026, pp. 106613–106631, 2026.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pp. 1126–1135. PMLR, 2017.

Jiaxi Hu, Yongqi Pan, Jusen Du, Disen Lan, Tang Tang, Qingsong Wen, Yuxuan Liang, and Weigao Sun. Improving bilinear rnn with closed-loop control. *Advances in Neural Information Processing Systems*, 38:107480–107506, 2026.

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 167283–167308. Curran Associates, Inc., 2025. doi: 10.52202/085713-5576.

Yuqi Pan, Yongqi An, Zheng Li, Yuhong Chou, Ruijie Zhu, Xiaohui Wang, Mingxuan Wang, Jinqiao Wang, and Guoqi Li. Scaling linear attention with sparse state expansion. *arXiv preprint arXiv:2507.16577*, 2025.

Leigang Qu, Ziyang Wang, Na Zheng, Wenjie Wang, Liqiang Nie, and Tat-Seng Chua. Ttom: Test-time optimization and memorization for compositional video generation. In *International Conference on Learning Representations*, volume 2026, pp. 70225–70247, 2026.

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, 2026. URL https://arxiv.org/abs/2512.14614.

Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *International conference on machine learning*, pp. 9229–9248. PMLR, 2020.

Yu Sun, Xinhao Li, Karan Dalal, Jiarui Xu, Arjun Vikram, Genghan Zhang, Yann Dubois, Xinlei Chen, Xiaolong Wang, Sanmi Koyejo, Tatsunori Hashimoto, and Carlos Guestrin. Learning to (Learn at test time): RNNs with expressive hidden states. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 57503–57522. PMLR, 13–19 Jul 2025.

DreamX Team, Yancheng Bai, Rui Chen, Xiangxiang Chu, Rujing Dang, Hao Dou, Bingjie Gao, Qiwen Gu, Siyu Hong, Jiachen Lei, Geng Li, Jifan Li, Ruimin Lin, Qingfeng Shi, Bingze Song, Lei Sun, Jing Tang, Ruitian Tian, Jun Wang, Jiahong Wu, Pengfei Zhang, Shen Zhang, and Jiashu Zhu. Dreamx-world 1.0: A general-purpose interactive world model, 2026. URL https://arxiv.org/abs/2606.16993.

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

Tong Wu, Shuai Yang, Ryan Po, Yinghao Xu, Ziwei Liu, Dahua Lin, and Gordon Wetzstein. Video world models with long-term spatial memory. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 49371–49393. Curran Associates, Inc., 2025. doi: 10.52202/085713-1651.

Zeqi Xiao, Yushi LAN, Yifan Zhou, Wenqi Ouyang, Shuai Yang, Yanhong Zeng, and Xingang Pan. Worldmem: Long-term consistent world simulation with memory. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 49632–49652. Curran Associates, Inc., 2025. doi: 10.52202/085713-1659.

Jiacong Xu, Hanwen Jiang, Zhixin Shu, Kalyan Sunkavalli, Vishal M. Patel, and Yiqun Mei. Wonder: Video world model done better, 2026. URL https://arxiv.org/abs/2607.26037.

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

Lvmin Zhang, Shengqu Cai, Muyang Li, Gordon Wetzstein, and Maneesh Agrawala. Frame context packing and drift prevention in next-frame-prediction video diffusion models. *Advances in Neural Information Processing Systems*, 38:34436–34456, 2025.

Ruicheng Zhang, Jun Zhou, Zunnan Xu, Zihao Liu, Jiehui Huang, Mingyang Zhang, Yu Sun, and Xiu Li. Zo3t: Zero-shot 3d-aware trajectory-guided image-to-video generation via test-time training. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 12708–12716, 2026.

Zeyun Zhong, Joya Chen, Manuel Martin, Frederik Diederichs, Juergen Gall, and Juergen Beyerer. Rethinking expressivity and efficiency in test-time training. *arXiv preprint arXiv:2608.21308*, 2026.

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, 2026. URL https://arxiv.org/abs/2605.15178.

# A Appendix

## A.1 Related Work

**Long-horizon video generation.** Long-video generation requires efficient temporal processing and control of accumulated errors. Block-wise diffusion and recurrent sequence modeling enable minute-scale generation with reduced attention cost (Chen et al., 2026a; Dalal et al., 2025), while diffusion forcing supports extended video continuation (Chen et al., 2025). Compressing historical frames into a bounded context further reduces processing costs, and modified sampling strategies help mitigate drift (Zhang et al., 2025). Training on self-generated autoregressive histories addresses the discrepancy between clean training context and imperfect inference-time observations (Huang et al., 2025), while causal generation and caching support efficient streaming (Yang et al., 2026). Camera- and action-conditioned models extend these capabilities to explorable environments (Zhu et al., 2026; Wang et al., 2026). NARRA builds on the full-context bidirectional generator in (Zhu et al., 2026), retaining its camera-conditioning mechanism and frozen refiner. We use an initial generated video to adapt selected generator parameters before final sampling, targeting scene persistence within this existing generation pipeline.

**Memory for video world models.** Explicit memory preserves observations that may become relevant when the camera returns to an earlier location. Historical images and poses can be stored in memory banks (Xiao et al., 2025), while spatially organized memories support consistency across viewpoint changes Wu et al. (2025). Camera pose and view overlap guide retrieval in Team et al. (2026); bounded landmark banks limit historical storage in Chen et al. (2026b); and sparse attention selects relevant portions of a growing KV cache in Xu et al. (2026). NARRA uses camera geometry to encourage latent agreement across revisited views during adaptation, while aggregated recurrent segment states provide historical context during final generation.

**Test-time adaptation and meta-learning.** Test-time training layers represent sequence state through parameters updated on the observed context (Sun et al., 2020; Behrouz et al., 2026c), and their use in video backbones enables efficient modeling of minute-long sequences (Dalal et al., 2025). Inference-time optimization has also been applied to trajectory guidance and compositional or textual alignment (Zhang et al., 2026; Qu et al., 2026). Gradient-based meta-learning complements these approaches by learning an initialization suited to adaptation through a small number of updates (Finn et al., 2017). NARRA combines a meta-learned initialization with nested test-time updates, using its initial generated rollout as self-supervision for scene persistence across revisited viewpoints.

**Neural memory and multiple learning timescales.** Neural-memory architectures encode observations in parameters updated at inference time (Sun et al., 2025; Behrouz et al., 2025b), with memory architecture, retention, and optimization determining how information is incorporated and preserved (Behrouz et al., 2026c). To improve the performance of deep memory modules, several studies have focused on improving the update algorithms: E.g., accounting for both current and past observations (Behrouz et al., 2026b; Hu et al., 2026; Zhong et al., 2026), channel-wise learning rate, and decay term (Behrouz et al., 2026c), sparsify the memory (Du et al., 2026; Pan et al., 2025). Nested Learning (Behrouz et al., 2025a) organizes interacting learning processes through distinct update frequencies, while Memory Caching (Behrouz et al., 2026a) supplements recurrence with states retained from earlier sequence segments. NARRA draws on these ideas through coupled fast and slow parameter updates and a channel-weighted read from aggregated segment states, combining adaptation to the generated world with access to earlier temporal context.

## A.2 Effect of Camera-Travel Distance

We further stratify trajectories by camera-travel distance into short (0.4–3.3 m), medium (3.6–7.6 m), and long (8.1–9.6 m) ranges. This analysis examines performance as the spatial extent of the camera excursion increases, independently of the benchmark's simple/hard trajectory labels. Figure 6 compares frozen SANA-WM, NARRA only using the Narra basic TTTcomponent, and NARRA with all components within each range.

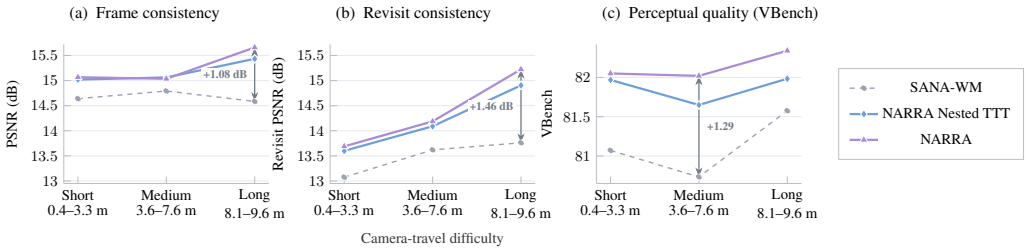



Figure 6: Camera-travel regimes. NARRA consistently improves frame consistency, revisit consistency, and perceptual video quality, with the largest improvements under long camera travel.

Both test-time adaptation methods outperform frozen SANA-WM across all three distance groups on PSNR, N-PSNR, and VBench. The largest consistency gains occur on long trajectories: relative to the frozen model, NARRA improves PSNR by 1.08 dB and N-PSNR by 1.46 dB, compared with gains of 0.43 dB and 0.62 dB on short trajectories. Its gains over the NARRA Narra nested CMSvariant are also largest in the long-distance group, reaching 0.23 dB in PSNR and 0.32 dB in N-PSNR. On medium trajectories, Narra nested CMSretains a small 0.02 dB advantage in PSNR, while NARRA achieves higher N-PSNR and VBench.

NARRA also attains the highest VBench score in every distance group. Its largest improvement over frozen SANA-WM occurs on medium trajectories, where VBench increases by 1.29 points; the corresponding gains on short and long trajectories are 0.98 and 0.76 points. Overall, the strongest revisit-consistency gains occur on the longest camera excursions, while perceptual-quality improvements extend across all distance ranges. Because each range contains different scenes, however, this analysis does not isolate camera-travel distance as the sole source of the observed performance differences.

### A.3 Ablation Studies

| | PSNR | SSIM | MSE | N-PSNR | N-SSIM | N-MSE | VBench |
|---|---|---|---|---|---|---|---|
| (a) Frozen backbone | 14.29 | 0.306 | 2747 | 13.04 | 0.262 | 3727 | 82.04 |
| **TEST-TIME ADAPTATION** | | | | | | | |
| (b) Query-key TTT | 14.77 | 0.317 | 2498 | 13.60 | 0.270 | 3298 | 82.32 |
| **(c) Full-projection TTT** ⋆ | 14.71 | 0.302 | 2539 | 13.81 | 0.270 | 3175 | 82.51 |
| **NESTED ADAPTATION** | | | | | | | |
| (d) Feed-forward nested adaptation | 14.43 | 0.310 | 2683 | 13.19 | 0.268 | 3626 | 82.09 |
| (e) Attention nested adaptation | 14.28 | 0.305 | 2757 | 13.03 | 0.262 | 3728 | 82.01 |
| (f) Joint recurrent-feed-forward adaptation | 14.93 | 0.305 | 2387 | 13.86 | 0.266 | 3133 | 82.16 |
| **(g) Meta-initialized joint adaptation** ⋆ | 14.90 | 0.306 | 2406 | 13.82 | 0.267 | 3171 | 82.93 |
| **CAMERA REGULARIZATION** | | | | | | | |
| (h) Motion-weighted residual regularization | 14.85 | 0.305 | 2432 | 13.79 | 0.267 | 3179 | 82.22 |
| (i) Minimum-return latent consistency | 14.73 | 0.308 | 2554 | 13.74 | 0.274 | 3291 | 82.49 |
| (j) Counterfactual return ranking | 14.89 | 0.304 | 2410 | 13.81 | 0.267 | 3177 | 82.26 |
| (k) Multi-return relative consistency | 14.85 | 0.305 | 2434 | 13.79 | 0.267 | 3185 | 82.28 |
| **(l) Matched-pose feature consistency** ⋆ | 14.97 | 0.309 | 2417 | 13.84 | 0.270 | 3249 | 82.61 |
| **MEMORY CACHING** | | | | | | | |
| (m) Uniform residual memory | 14.79 | 0.298 | 2472 | 13.76 | 0.265 | 3199 | 82.34 |
| (n) Diffusion-scheduled memory | 14.79 | 0.304 | 2487 | 13.67 | 0.264 | 3311 | 82.22 |
| (o) Relevance-weighted memory | 14.87 | 0.304 | 2429 | 13.78 | 0.268 | 3229 | 82.76 |
| **(p) Channel-wise residual memory** ⋆ | 15.06 | 0.305 | 2325 | 13.89 | 0.265 | 3114 | 82.30 |

Figure 7: **Ablation overview on a 12-video subset.** Stars mark the selected configuration from each ablation family.

We first examine which parameters should be updated during test-time adaptation. Figure 7 compares conventional test-time adaptation with localized and joint nested variants on the same 12-video cohort. We evaluate each configuration using standard reconstruction metrics, temporally stratified revisit metrics, and VBench.

**Choice of Adaptation Parameters** The parameters selected for test-time adaptation determine where video-specific information can be incorporated and how it propagates through the model. Figure 7**(a–g)** compares several NARRA adaptation variants on the same 12-video cohort, ranging from conventional test-time training to localized and joint nested updates.

The conventional TTT variants of NARRA apply Equation 2 across all 15 GDN blocks, without the nested fast–slow adaptation mechanism. Query–key TTT **(b)** updates $W_Q$ and $W_K$, whereas full-projection TTT **(c)** additionally updates $W_V$ and $W_O$. Query–key adaptation performs better on the standard reconstruction metrics, while full-projection adaptation yields stronger long-term revisit metrics

and higher VBench. This suggests that adapting $W_V$ and $W_O$ provides additional flexibility in both the content represented by the recurrent blocks and how that content is propagated downstream.

The localized nested variants restrict the fast update in Equation 3 to a terminal component: $\delta \subset W_{\text{FFN}}^{\text{final}}$ in **(d)** and $\delta \subset W_O^{\text{final}}$ in **(e)**. Because these updates act only at the final block, they cannot alter how earlier GDN blocks construct and propagate recurrent state, and their performance remains close to the frozen model **(a)**.

The joint nested variants **(f,g)** use the full optimization in Equations 3–4, combining rapid feed-forward specialization with slower adaptation of the recurrent representation. Configuration **(f)** initializes the fast memory from $\phi^0 = 0$, whereas **(g)** uses the learned initialization $\phi^0 = \Phi^\star$, providing a reusable starting point for rapid adaptation. Relative to full-projection TTT **(c)**, configuration **(g)** improves VBench by 0.42 while maintaining comparable revisit metrics.

**Effect of Segment-Memory Caching** We next compare four memory-caching strategies based on Equation 7. All variants build on the same meta-initialized parent **(g)** and differ only in how previous segment states are aggregated or how the retrieved response is scaled. Figure 7**(m–p)** summarizes the results.

Uniform residual memory **(m)** averages all preceding segment states and applies a constant scalar gain, $\boldsymbol{\lambda} = 1$. Diffusion-scheduled memory **(n)** retains the same uniform aggregation but varies the retrieval strength over denoising as $\boldsymbol{\lambda}_\tau = \lambda(1 - \tau/T)^p \mathbf{1}$, reducing the contribution of cached information at noisier steps and increasing it as the sample becomes cleaner. Relevance-weighted memory **(o)** instead replaces uniform aggregation with query-dependent weights. Given cosine similarities $\rho_{s,j}$ between the current query summary and each preceding state, it uses $\alpha_{s,j} \propto [\rho_{s,j} - \overline{\rho} s - \eta]+$ to suppress incompatible segments. However, the resulting reconstruction gains remain modest.

Channel-wise residual memory **(p)** retains the uniform state average but replaces the scalar gain with $\lambda_c = \lambda\left(\frac{1}{2} + \frac{c-1}{C-1}\right), \qquad c = 1, \ldots, C$. The gains span $[0.5\lambda, 1.5\lambda]$ with mean $\lambda$, redistributing the memory contribution across feature channels without changing its average magnitude. The current queries determine which tokens retrieve cached information, while $\lambda_c$ controls how strongly each feature channel incorporates the retrieved state. This allows different feature channels to rely on memory to different degrees.

As shown in Figure 7, channel-wise memory **(p)** achieves the strongest reconstruction and revisit fidelity among the caching strategies. Although its VBench score is lower on the 12-video subset, it consistently improves both standard and revisit reconstruction metrics, supporting channel-wise modulation as the preferred strategy for preserving scene information. These gains become more pronounced on the complete 50-video evaluation.

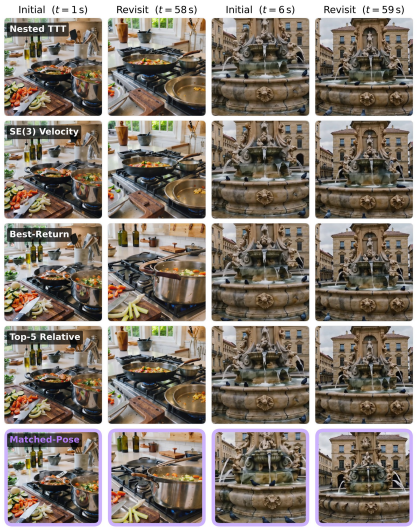



Figure 8: Effect of camera-return regularization. Matched-pose consistency preserve the cooking pot and the fountain's single front ornament.

**Effect of Camera-Return Consistency** Finally, we compare five camera-aware objectives using the same meta-initialized joint parent **(g)**. Figure 7**(h–l)** summarizes the results. Let $\mathbf{e}_t = \widehat{\mathbf{z}} * 0, t - \mathbf{z} * v, t$ denote the latent prediction residual.

Motion-weighted regularization **(h)** minimizes $\langle w_t |\mathbf{e}_{t+1} - \mathbf{e}_t| * 2^2 \rangle_t$, assigning larger weights to frames separated by greater camera motion. Minimum-return consistency **(i)** minimizes $|\widehat{\mathbf{z}} * 0, b - \mathbf{z}_{v,a}| * 2^2$ for the closest qualifying return pair $(a, b)$. Counterfactual ranking **(j)** uses $\langle \log(1 + \exp(d * a, b - d_{a,c} + \mu)) \rangle$, encouraging the selected return $b$ to match the earlier observation $a$ more closely than a shifted alternative $c$. Multi-return consistency **(k)** applies $\langle |\mathbf{e}_b - \mathbf{e}_a| * 2^2 \rangle * (a, b) \in \mathcal{H}_5$ across multiple geometrically ranked returns. Finally,

atched-pose feature consistency **(l)** instead uses the pooled-feature distance $d_{\mathrm{ret}}$ from Equation 5.

The first four objectives operate directly on latent residuals or latent distances. Direct latent matching can be sensitive to small pose errors, parallax, and occlusion, while residual-based objectives do not explicitly enforce recovery of the same scene content upon return. Matched-pose consistency instead compares pooled features at corresponding poses spatially, making the constraint more tolerant to local misalignment while still encouraging preservation of revisited content.

Matched-pose consistency **(l)** achieves the strongest overall performance within this family and is therefore used in the final model. Figure 8 further highlights qualitative differences that are not fully captured by the aggregate metrics. In the kitchen example, minimum-return and matched-pose consistency preserve the deep cooking pot, whereas the other approaches replace it with a shallow pan. In the fountain example, only matched-pose consistency preserves the single front ornament visible in the initial view, while the other displayed methods introduce an additional ornament.

### A.4 60-Second Evaluation of External World Models

We attempted to evaluate *ReWorld* Chen et al. (2026b), *WorldPlay* Sun et al. (2026), and *DreamX-World* Team et al. (2026) using the complete 60-second versions of our departure-and-return trajectories. Our goal was to preserve the same evaluation intent across all methods: each model received the same conditioning image and caption, followed an outbound camera path, and was then instructed to retrace that path to the initial camera pose.

The released methods expose different camera-control representations and were originally evaluated using different temporal horizons. Consequently, an identical low-level camera-control sequence cannot be supplied to every model. For each method, we converted the continuous camera trajectory into the closest native control representation supported by its released implementation while preserving the outbound motion, inverse return, and approximately 60-second duration.

These papers report quantitative results at different temporal horizons, but none evaluates our exact 60-second departure-and-return protocol: *ReWorld* reports a 64-second out-and-back memory experiment, *WorldPlay* reports short- and long-term reconstruction metrics without a matched 60-second return setting, and *DreamX-World*'s principal quantitative camera-control evaluation uses 5-second clips. We therefore evaluated the released *ReWorld*-5B model on our 60-second trajectories and a paper-inspired 64-second return trajectory. We also converted our trajectories to *WorldPlay*'s pose-conditioning format and *DreamX-World*'s native action representation, generating approximately 60 seconds with each method's released long-horizon configuration.

The minute-long outputs frequently exhibited scene drift, scene replacement, loss of visual anchors, or failure to follow the return path. In these cases, the final frames no longer represented a meaningful revisit of the initial scene, making PSNR, SSIM, and MSE difficult to interpret. We therefore omit these methods from the quantitative 60-second comparison. Representative outputs and failure cases are available on the project website. world-models-60s