LaST: Latent State Tracking for Persistent Object Memory in Video World Models

Anonymous authors
Paper under double-blind review

Abstract

Video world models must preserve object identity across interruptions in visibility. Yet the observations establishing that identity can leave the local attention window long before an object reappears. Persistent memory must retain relevant visual information and make it available to later generation. We introduce LaST, Latent State Tracking, which separates persistent object memory from local video generation. A recurrent writer maintains scene context, object geometry, and appearance across context windows. A learned reader uses geometry to retrieve appearance at the requested locations and combines decoded scene and object fields to correct a frozen generator's denoising predictions. We pretrain memory through future reconstruction, then jointly optimize state construction and reader corrections over multiple generation windows. Parameter-free rescaling regulates spatial contrast. On the full WRBench evaluation set, LaST improves all five metrics over both KV baselines at both 1.3B and 5B. A capacity study on MOSEv2 relates reconstruction quality and object coverage to the number of retained slots. These results support retaining and reusing object history beyond the generator's local attention window.

1 Introduction

Video generators can now synthesize detailed scenes and support interactive environments (Wan et al., 2025; Bruce et al., 2024). Maintaining a coherent world over time also requires objects to retain their identities when the camera revisits them or an occluder moves away. Yet the observations establishing an object's identity can leave the local attention window before it reappears. Recent evaluations reveal failures of visual memory under such interruptions (He et al., 2026). Persistent object memory therefore requires access to relevant visual history beyond the generator's active context.

Long-video methods improve rollout length and temporal conditioning. Diffusion Forcing extends generation beyond the training horizon, while Self Forcing reduces the mismatch between training and inference histories (Chen et al., 2024; Huang et al., 2025). For streaming generation, LongLive combines a rolling window with an initial-frame anchor (Yang et al., 2025), although later observations can still fall outside the retained context. Full-history key–value (KV) caching retains earlier features at increasing storage and attention cost (Wu et al., 2026). WorldMem and Context as Memory retrieve historical views to support scene consistency (Xiao et al., 2026; Yu et al., 2025). Context packing and selective KV retention bound the context supplied to generation (Zhang et al., 2025; Zhao et al., 2026). Within a bounded budget, the central question becomes which object information to retain and how to use it after the original observations are no longer accessible.

Recurrent and object-centric representations offer complementary ways to organize history. State-space and hybrid-attention models combine recurrent memory with local video processing (Po et al., 2025; Li et al., 2026), and Ripple compresses evicted context into fixed-size modality memories (Ding et al., 2026). At scene level, AlayaVista separates panoramic-state evolution from perspective synthesis (Tan et al., 2026). Object-centric representations organize observations by entity (Locatello et al., 2020; Kipf et al., 2022). SlotMemory uses object slots to address retained KV tokens (Dou et al., 2026), while SlotMem learns compact character memories with LoRA-based generator adaptation (Liu et al., 2026). HyDRA and ReMind address dynamic subjects and out-of-sight state evolution through memory retrieval (Chen et al., 2026; Xu et al., 2026).

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.

For a frozen short-context generator, two coupled challenges remain. Object history must survive context replacement within a fixed state budget, and the stored state must guide later denoising. A query may miss an object even when its appearance is retained. Reconstruction error can therefore reflect either forgetting or localization error.

To address these challenges, we introduce LaST, Latent State Tracking, which decouples persistent object memory from local video generation (Figure 1). A learned writer integrates observations into separate scene, geometry, and appearance records. Scene records retain background context, while paired geometry and appearance records make an object's visual details retrievable as its position changes. We train memory to reconstruct future observations, using teacher-supplied geometry to learn appearance retention before introducing reads based on predicted geometry. A learned reader queries the stored records at the requested space–time locations and uses their decoded fields to correct the frozen generator's denoised latent estimate. Parameter-free rescaling regulates spatial contrast in the affected region using the uncorrected prediction as a reference.

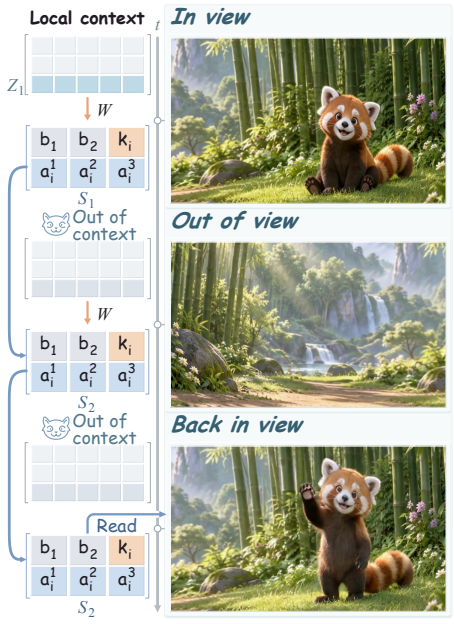

Figure 1: **Persistent object memory.** Left: recurrent updates to *scene, geometry, and appearance* records (*gray, peach, and blue*). Right: LaST (5B) at 0, 2.5, and 5 s with camera control during generation. *Prompt (abridged): stand up and wave one front paw.*

We evaluate LaST against Short-KV and Full-KV on the full WRBench evaluation set, measuring spatial relations and event states while visible and after re-observation. LaST improves all five metrics at both 1.3B and 5B. On MOSEv2, varying the number of object slots shows how reconstruction quality and object coverage change with memory capacity. These evaluations characterize generation consistency and the capacity needed to retain object information.

Our main contributions are as follows:

- **Persistent object memory.** LaST maintains separate scene, geometry, and appearance records that preserve object history beyond a frozen generator's local context.
- **Learning to retain and retrieve.** We connect teacher-guided reconstruction to denoising through staged reader training and joint optimization over generated histories.
- **Consistency with lower history cost.** LaST improves all five WRBench metrics at both scales. Seven-scene 1.3B measurements show 70.0% lower history storage and 21.5% lower continuation latency after history preparation.

2 Related Work

Long-horizon video generation. Video diffusion has progressed from extensions of image diffusion (Ho et al., 2022) to large-scale latent models and diffusion transformers, exemplified by Stable Video Diffusion and Wan (Blattmann et al., 2023; Wan et al., 2025). Extending these generators to long sequences requires both stable rollouts and access to earlier observations. Diffusion Forcing supports generation beyond the training horizon through independent token-wise noise levels (Chen et al., 2024). Self Forcing trains on self-generated histories to reduce the training–inference mismatch that contributes to error accumulation (Huang et al., 2025). For temporal conditioning, StreamingT2V combines preceding-chunk features with a fixed appearance anchor, while LongLive uses long-rollout training and retains the initial frame chunk alongside a rolling KV window (Henschel et al., 2025; Yang et al., 2025). FramePack allocates context by frame importance, and DensityKV selectively retains original KV pairs within a fixed storage budget (Zhang et al., 2025; Zhao et al., 2026). These approaches improve rollout training or the allocation of visual con-

text. LaST complements local generation with recurrent object state, preserving accumulated history beyond the lifetime of the frames used to construct it.

Visual memory for world models. WorldMem, Context as Memory, and spatial-memory world models reuse historical views through pose-aware conditioning or retrieval (Xiao et al., 2026; Yu et al., 2025; Wu et al., 2025). A fixed retrieval budget need not bound total storage. Context as Memory reads a fixed number of frames from a growing history (Yu et al., 2025). Matrix-Game 3.5 aligns memory patches to the target viewpoint (Qian et al., 2026). Alternatives include geometry-indexed latent caches (Wang et al., 2026), geometry-supervised summaries (Wei et al., 2026), and panoramic states with a separate perspective renderer (Tan et al., 2026). WorldTrace keeps historical summaries addressable through in-distribution temporal positions, whereas SlotMemory uses object slots to retrieve retained KV tokens (Wu et al., 2026; Dou et al., 2026). For dynamic subjects, HyDRA retrieves motion cues from compressed latents, while ReMind trains temporally indexed retrieval for out-of-sight state evolution (Chen et al., 2026; Xu et al., 2026).

Recurrent and object-centric state. PlaNet combines deterministic memory and stochastic transitions to infer latent states from observations (Hafner et al., 2019). Video state-space and hybrid-attention models combine recurrence with local processing (Po et al., 2025; Li et al., 2026), while Ripple compresses evicted KV into modality memories (Ding et al., 2026). Within a video clip, Video DeltaNet combines local attention with bidirectional linear memory updated from each frame's spatial tokens (Xi et al., 2026).

Object-centric models organize state by entity. Slot Attention extracts object slots, while SAVi extends this organization to video through recurrent updates (Locatello et al., 2020; Kipf et al., 2022). SlotFormer and SOLD predict slot dynamics for video simulation and planning, using a frozen reconstruction decoder and jointly trained world model, respectively (Wu et al., 2023a; Mosbach et al., 2025). SlotDiffusion conditions a learned diffusion decoder on slots (Wu et al., 2023b), and Temporal Slot Activation uses continuous gates to regulate slot updates and decoder participation (Nguyen et al., 2026). For long-video generation, SlotMem selectively updates character-addressed slots and injects them into matching regions, training LoRA adapters with pretrained base weights frozen (Liu et al., 2026). LaST instead learns a writer and denoised-latent readout around an unchanged generator. Unlike SlotMemory's retained KV, its recurrent object records accumulate history independently of the local attention window. This gives object memory a separate lifetime and storage budget without adapting the generator.

Evaluating persistent memory. Video benchmarks assess appearance consistency, physical plausibility, and commonsense faithfulness (Huang et al., 2024; Zheng et al., 2025; Li et al., 2025). Memory-specific evaluations test out-of-view recall and camera returns (He et al., 2026; King et al., 2026), alongside hidden-state tracking (Shin et al., 2026; Miao et al., 2026). Visual similarity alone does not establish dependence on the correct history. Read-time substitution tests this dependence by changing memory values while holding the preceding computation and read interface fixed (Tiwari et al., 2026). We measure visible and re-observed consistency with WRBench and examine reconstruction and object coverage separately through a capacity study on MOSEv2.

3 METHOD

LaST separates persistent object memory from the generator's local context. A writer updates scene, geometry, and appearance records from each completed segment (Figure 2). To generate the next segment, a reader uses stored geometry to query appearance records and correct the generator's denoising predictions.

3.1 SCENE, GEOMETRY, AND APPEARANCE MEMORY

An object's location changes over time, but its appearance must remain retrievable after it moves. We therefore store geometry separately from visual content (Figure 2a). The memory after segment t is $S_t = [B_t; K_t; A_t]$, where *scene* records B_t capture background context, *geometry* records K_t describe object position, size, and motion, and *appearance* records A_t retain visual content. Each of

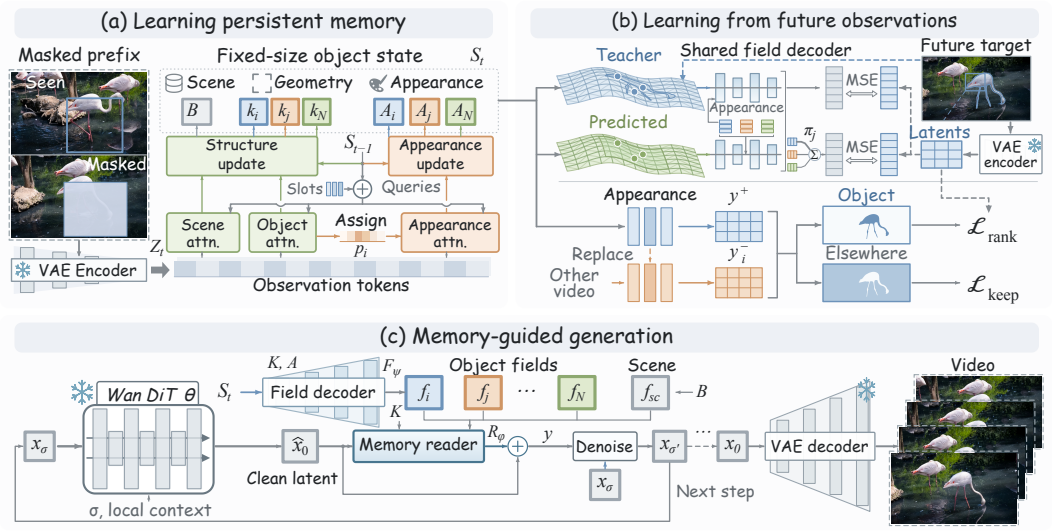

Figure 2: **Overview of LaST.** (a) Observations update scene, geometry, and appearance records. (b) Future reconstruction pretrains the state (top); appearance replacement supervises reader warm-up through $\mathcal{L}_{\text{rank}}$ and $\mathcal{L}_{\text{keep}}$ (bottom). Joint training subsequently updates the memory and reader together. (c) Decoded fields correct Wan's clean prediction before sampling; the frozen VAE decodes video. Snowflakes denote modules fixed in the depicted stage. Video frames illustrate the workflow.

the N object slots pairs one geometry record with a bank of appearance records. The record counts and widths are fixed, making state size independent of video length.

Memory update. A frozen VAE encodes the completed segment into latent observations Z_t, augmented with space–time coordinates. Starting from the preceding state S_{t-1}, the structural writer updates scene and geometry records and produces object–observation assignments P_t. The appearance writer uses these assignments to update the corresponding visual content. With $S_0 = 0$, the two updates are

$$(B_t, K_t, P_t) = \mathcal{W}_{\text{str}}(S_{t-1}, Z_t), \qquad A_t = \mathcal{W}_{\text{app}}\big(S_{t-1}, Z_t; \text{sg}(P_t)\big), \tag{1}$$

where sg denotes stop-gradient and P_t contains the per-frame assignments. Both writers process frames sequentially from S_{t-1}, using learned slot embeddings together with preceding records to query observations. Weak assignments reduce the replacement of stored appearance. Record-wise and cross-object updates then evolve the state, including objects with no new observations. Only the merged state S_t is carried to the next segment. Appendix A.1 gives the attention and update equations.

3.2 Memory-Guided Video Generation

Field decoding. To condition generation, we decode the stored records into latent features at the requested space–time locations (Figure 2c). Let $q = (\tau, \mathbf{u})$ specify a query time and position, and write S for the state held fixed during a read. For object i, geometry records K predict its center from the stored position and velocity. The transform $T_i(q; K)$ expresses the query relative to this center, object size, and reference time. We use these coordinates to retrieve the object's appearance from A_i:

$$f_i(q; S) = \mathcal{F}_\psi\big(\gamma(T_i(q; K)), A_i\big), \tag{2}$$

where γ is a Fourier encoding and $\mathcal{F}_\psi$ is a cross-attention decoder (Vaswani et al., 2017) with parameters ψ. Evaluating f_i across locations yields an object field, a spatial map of predicted latent features. Spatial interpolation of B gives the scene field. Geometry also supplies weights $\pi_i(q; K)$ indicating where each object field contributes in the video. Appendix A.1 specifies the transform, weights, and decoders.

Memory-conditioned denoising. At noise level σ, the frozen Wan generator (Wan et al., 2025) predicts flow v_θ from noisy latent x_σ, giving the clean estimate $b = \widehat{x}_0 = x_\sigma - \sigma v_\theta$. The reader

compares each scene or object field with this estimate and combines the candidate corrections $f_j - b$ according to content compatibility and spatial priors. A zero-valued candidate lets the reader reduce unsupported corrections. We then rescale the corrected estimate using b's spatial standard deviation and retain changes only where the read is nonzero. Denoting the complete correction by $\mathcal{R}_\phi(b, S)$, with reader parameters ϕ, the corrected prediction and sampling flow are

$$\mathcal{Y}_\phi(b, S) = b + \mathcal{R}_\phi(b, S), \qquad v^{\text{mem}} = v_\theta - \frac{\mathcal{R}_\phi(b, S)}{\sigma}, \tag{3}$$

for $\sigma > 0$. Sampling uses v^{mem} in place of v_θ, and the frozen VAE decodes the final latent into video. The conditioning frame and memory state remain fixed throughout denoising. The completed segment then provides the observations for the next memory update. Appendix A.1–A.2 details the reader and parameter-free rescaling.

3.3 Learning to Retain and Use Object History

The state must retain object details that recent observations no longer contain, and the reader must use them during generation. We pretrain the state through future reconstruction (Assran et al., 2023; Baevski et al., 2022) and warm up the reader before joint optimization on progressively longer generated histories (Section 3.4). Given an observed prefix H and its continuation U, we retain an object's earliest observation in H while masking its later observations. Reconstructing the object in U trains the writer to carry its earlier appearance forward in the prefix state S. Future annotations and clean target latents supervise training but are absent from inference inputs.

Teacher-guided reconstruction. Reconstruction errors can arise from either misplaced geometry or forgotten appearance. To supervise appearance independently of predicted geometry, we use future annotations to specify the object and its coordinate transform T_i^{teach} (Figure 2b, top). The teacher-guided read retrieves appearance from S at these supplied coordinates, while the predicted read uses the state's own geometry. For a query assigned to object i, the two reads are

$$\mathcal{D}^{\text{teach}}(q; S) = \mathcal{F}_\psi\big(\gamma(T_i^{\text{teach}}(q)), A_i\big), \qquad \mathcal{D}^{\text{pred}}(q; S) = \sum_{j=1}^{N} \pi_j(q; K) f_j(q; S). \tag{4}$$

Both reads retrieve the same appearance records through $\mathcal{F}_\psi$. Teacher geometry and assignments are detached, and weight not assigned to an object contributes zero. We supervise either read $\mathcal{D}$ at valid future object queries $\mathcal{Q}_U$, using the corresponding clean latents $z_q \in \mathbb{R}^C$ as targets. The reconstruction objective averages the squared error over queries and channels:

$$\mathcal{L}_{\text{rec}}(\mathcal{D}) = \mathbb{E}_{H,U}\left[\frac{1}{C|\mathcal{Q}_U|}\sum_{q\in\mathcal{Q}_U} \|\mathcal{D}(q; S) - z_q\|_2^2\right], \tag{5}$$

where the expectation is over training videos. The teacher-guided objective trains appearance recall at supplied locations, while the predicted read also has to locate the object.

State pretraining. We pretrain the state in stages so that appearance learning does not initially depend on predicted geometry. Training first establishes scene, geometry, and observation assignments, then holds them fixed while teacher-guided reconstruction trains the appearance writer and field decoder. With the writers fixed, both reads and object/null classification subsequently train the field and geometry decoders, followed by joint representation refinement. To retain video-specific details, an auxiliary loss matches differences between two videos' appearance records to differences between their future-crop targets, without assuming shared identities. A second loss separates correct-history and unrelated-history reconstruction errors by a margin proportional to the benefit over a zero state. Appendix A.3 defines these representation objectives.

Reader warm-up. To make the reader sensitive to relevant object history, we hold the pretrained memory fixed and initialize the reader to produce zero output. Replacing one object's appearance records with those from another video yields a paired read at the same noisy input, timestep, and base prediction (Figure 2b, bottom). Residual ranking favors the original-memory correction within the object's visible region, while consistency outside that region discourages unrelated changes. An additional loss penalizes replacement errors above the larger of the original-memory and no-memory errors, discouraging the reader from satisfying the ranking by damaging the replaced prediction. Appendix A.3 gives these losses and their normalization. Inference uses the original state without replacements or visibility annotations.

3.4 JOINT TRAINING WITH GENERATED HISTORY

After warm-up, we optimize the memory for the corrections needed during generation. Generation losses jointly train the writers, state updates, field decoder, and reader, while Wan, the VAE, and the text encoder remain frozen. Gradients pass through the current state and decoded fields to the two most recent state updates, allowing the output objective to change which observations are retained. Earlier states and preceding denoising steps are detached.

Generation and memory supervision. For a clean target latent y and Gaussian noise ϵ, we form $x_\sigma = (1-\sigma)y + \sigma\epsilon$. The memory-conditioned flow from Eq. (3) is trained against $u = \epsilon - y$:

$$\mathcal{L}_{\text{flow}} = \mathbb{E}_{y,\epsilon,\sigma}\left[\frac{1}{C|\mathcal{V}|}\sum_{i\in\mathcal{V}} \|v_i^{\text{mem}}(x_\sigma, S) - u_i\|_2^2\right], \tag{6}$$

where $\mathcal{V}$ contains valid latent locations excluding the conditioning frame. When valid targets are available, we sample forward-noised latents and student-rollout queries in equal proportion, without treating a rollout query as a forward-noised version of the recorded future.

We use a long-history reference to identify locations where earlier observations improve the short-context prediction. Let p_s, p_f, p denote the Short-KV, long-history, and LaST flow predictions at the same noisy latent, timestep, and conditions. For target-aligned forward-noising samples, the channel-averaged error $\ell(a,u) = \|a-u\|_2^2/C$ defines the detached support

$$m_i = \text{sg}(\mathbf{1}[\ell(p_{f,i}, u_i) + \delta < \ell(p_{s,i}, u_i)]), \tag{7}$$

with margin $\delta > 0$ selected on the development set. LaST matches the long-history prediction on this support:

$$\mathcal{L}_{\text{memory}} = \mathbb{E}\left[\frac{1}{C|\mathcal{V}|}\sum_{i\in\mathcal{V}} m_i \|p_i - \text{sg}(p_{f,i})\|_2^2\right]. \tag{8}$$

The loss is normalized over all valid locations, so sparse support is not amplified and empty support contributes zero. Both reference predictions are detached, and the teacher receives earlier observations rather than future targets as inputs.

The joint objective combines these terms with local preservation, identity supervision, and the state-pretraining objectives:

$$\mathcal{L}_{\text{joint}} = \mathcal{L}_{\text{flow}} + 0.5\mathcal{L}_{\text{memory}} + \mathcal{L}_{\text{local}} + 0.1\mathcal{L}_{\text{id}} + 0.1\mathcal{L}_{\text{repr}}. \tag{9}$$

Local preservation protects regions already represented correctly by Short-KV and encourages the reader to abstain under unrelated memory, replacing the warm-up ranking objective. Identity supervision uses objects with reliable correspondence, while $\mathcal{L}_{\text{repr}}$ retains the reconstruction and state objectives above. Each term is normalized over its valid support before weighting.

Multi-window histories. To expose memory to generated context (Huang et al., 2025), we train on episodes of two, four, and eight windows. Each window contains 41 frames with stride 40, giving 81, 161, and 321 frames per episode. Completed clean outputs update the state across windows without resets; noisy intermediate latents do not enter the observation history. Joint training starts with real histories, then increases student-history episodes from 25% to 50%, retaining at least half with real histories and reliable targets.

Generated histories can shift object positions or event timing relative to the source video, so source-video pixel targets and Eq. (7) apply only where correspondence remains reliable. Teacher matching on generated histories requires a separately validated same-history reference and is otherwise inactive. Training batches combine 75% of the fixed base distribution with 25% deduplicated failure replay from training episodes. Appendix A.4 details optimization and replay.

4 EXPERIMENTS

We compare persistent object memory with longer local KV history through WRBench evaluation, state interventions, repeated continuation, and capacity and efficiency measurements.

4.1 Experimental Setup

Models and memory settings. We compare LaST, Short-KV, and Full-KV on Wan2.1-Fun-V1.1-1.3B-Control-Camera and Wan2.2-Fun-5B-Control-Camera (Wan et al., 2025). Short-KV retains three historical latent frames; Full-KV retains ten at 1.3B and fifteen at 5B, excluding the segment-boundary conditioning frame. Both retain full attention within the current window. LaST adds persistent scene, geometry, and appearance records to Short-KV, with the generator and VAE frozen. Its four-slot state uses 12 KiB in BF16, excluding local KV and temporary buffers (Figure 3).

Evaluation. We evaluate all three memory settings at both scales on the full WR-Bench evaluation set using its official evaluator (Lu et al., 2026). The five metrics measure visual integrity, visible spatial and state consistency, and re-observed spatial and state consistency. Higher is better; re-observed scores require judgeable return evidence. Table 1 bolds the best of the three settings at each scale. Selected comparisons from WRBench Table 2 have lower five-score means than LaST (5B). A dagger marks sparse re-observation support.

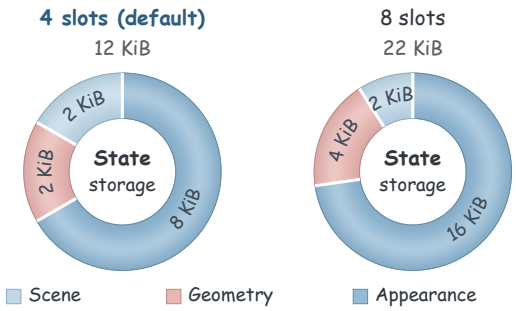

Figure 3: **State storage (KiB).** Four slots (left), eight (right); excludes KV and temporary fields.

The 1.3B results use eight sampling steps per segment at 832×480: a 41-frame prefix and 41-frame continuation yield 81 frames at 16 fps after removing one overlapping frame. At 5B, LaST and both KV baselines use 50 sampling steps per segment, 1280×704 resolution, and 121 frames at 24 fps. Camera trajectories condition generation, directing the view away from the subject and back for re-observation.

Training. We mix the official training splits of MOSEv2 (Ding et al., 2025), VOST (Tokmakov et al., 2023), and YouTube-VOS 2019 (Yang et al., 2019) with episode sampling weights of 50%, 20%, and 30%, using dataset adapters and a unified episode format. Each cycle comprises 20,000 draws with replacement and 2,500 updates at an effective batch of eight. Appendix A.4 details data preparation and optimization.

4.2 Quantitative and Qualitative Results

LaST exceeds both KV settings on all five metrics at both scales while retaining the Short-KV history length (Table 1). Re-observed spatial consistency rises from 0.613 to 0.659 at 1.3B and from 0.608 to 0.745 at 5B relative to Short-KV. Longer local history has mixed effects: Full-KV falls below Short-KV on all five metrics at 1.3B; at 5B it improves visible spatial, visible state, and re-observed spatial consistency, but not integrity or re-observed state consistency. The gains extend to visible observations, although the 5B visible-state gain over Full-KV is small (0.576 versus 0.574). These score differences do not establish statistical significance.

Object state after re-observation. LaST scores 0.639 at 1.3B, compared with 0.611 for Short-KV and 0.503 for Full-KV; at 5B the scores are 0.637, 0.589, and 0.587. Together with spatial consistency, these scores indicate improved location and state when return evidence is available.

Figure 4 compares four 5B examples under action prompts and generation-time camera control. The camera leaves the subject and returns, allowing comparison of appearance and pose across interrupted visibility. Equally spaced columns sample unequal time intervals, as the timestamps indicate.

4.3 Using and Updating Persistent State

These 5B diagnostics on MOSEv2 and VOST test memory dependence, updating, and repeated use separately from WRBench. Object similarity measures spatially matched DINOv2 (Oquab et al., 2024) feature agreement; full-frame MSE also responds to displacement and viewpoint. We average valid frames within each sequence and then across sequences.



Figure 4: **Visual comparison of Short-KV, Full-KV, and LaST (5B).** Each scene follows this row order. Right panels show action prompts and LaST crops at 0 and 5 s.

Table 1: **WRBench results.** *F: Full-KV; S: Short-KV.*

Method	Integrity↑	Visible spatial↑	Visible state↑	Reobs. spatial↑	Reobs. state↑
Wan-Fun 2.1-14B	0.846	0.733	0.530	0.659	0.621
LiveWorld	0.775	0.703	0.541	0.661	0.600
Hunyuan GameCraft	0.705	0.672	0.440	0.554†	0.490†
Hunyuan WorldPlay	0.870	0.737	0.523	0.640	0.603
MagicWorld	0.543	0.623	0.458	0.584	0.574
Wan2.1-Fun-Cam (1.3B)^{F}	0.405	0.706	0.561	0.509	0.503
Wan2.1-Fun-Cam (1.3B)^{S}	0.450	0.741	0.565	0.613	0.611
LaST (1.3B)	**0.494**	**0.783**	**0.604**	**0.659**	**0.639**
Wan2.2-Fun-Cam (5B)^{F}	0.342	0.769	0.574	0.622	0.587
Wan2.2-Fun-Cam (5B)^{S}	0.364	0.765	0.549	0.608	0.589
LaST (5B)	**0.741**	**0.812**	**0.576**	**0.745**	**0.637**

History content and state updates. With the current observation fixed, LaST exceeds Full-KV in object similarity and reduces pixel error on four sequences per dataset (Table 2). Interventions on stored content alter continuation quality, linking the retained history to subsequent generation. Under the same new observation, updating LaST's state improves object similarity and reduces pixel error relative to retaining the earlier state on four sequences per dataset.

Repeated continuation. Across two sequences per dataset, fourth-round mean similarity is 0.700 for LaST, 0.176 for Short-KV, and 0.133 for Full-KV. LaST leads in all four rounds on both datasets, with residual drift from 0.800 to 0.732 on MOSEv2 and from 0.905 to 0.667 on VOST. Each method continues from its own preceding output. Appendix A.7 provides all interventions, per-round curves, pixel errors, support counts, and camera-return examples. In the separate camera-return examples (Appendix A.7, Figure 11), fourth-continuation mean historical-object ROI similarity is 0.770 for LaST, versus 0.145 for Short-KV and 0.098 for Full-KV across two sequences. Full-frame MSE falls to 0.0049, compared with 0.0885 and 0.0697. The appendix shows every round, distinguishing object-region agreement from whole-frame error.

Table 2: **Object similarity** ↑ (5B).

Method / state	MOSEv2	VOST
History content		
Full-KV	0.331	0.348
LaST	**0.840**	**0.816**
Re-observation		
Earlier state	0.409	0.239
Updated state	**0.600**	**0.343**
Fourth continuation		
Short-KV	0.145	0.206
Full-KV	0.135	0.131
LaST	**0.732**	**0.667**

Memory content and state updates both affect continuation quality. Mismatched state reduces similarity on both datasets (Appendix A.7), and updating state improves both metrics under the same new observation. The effects are less object-specific on VOST: removing target or other-object records yields similar scores, while omitting state improves feature similarity at higher pixel error. This discrepancy motivates reporting both feature agreement and pixel error. LaST remains ahead of the KV baselines across rounds, despite the overall decline in similarity.

Object appearance under motion. Figure 5 follows a bear across three LaST frames. Its ears, face, and yellow fur remain recognizable as it stands and raises its paws. Enlarged crops expose these details, while full frames provide scene context. Dashed boxes mark the same crop region; external SAM contours delineate the subject. Pose changes are visible in the paws, while the head provides a stable appearance reference. These masks support post-hoc inspection of appearance across pose changes, rather than representing learned LaST tracks.

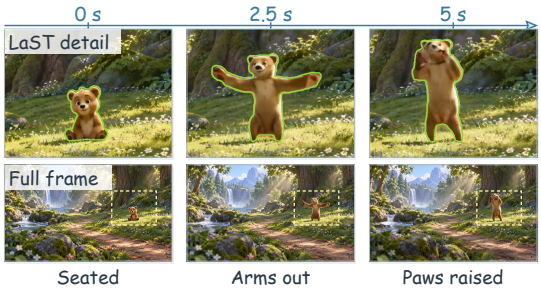

Figure 5: **Appearance across motion.** LaST frames with post-hoc masks and crop boxes.

4.4 Memory Efficiency and Capacity

Comparison with Full-KV. On seven 1.3B scenes (832×480, eight sampling steps, one NVIDIA H100 PCIe 80 GB per video), LaST stores 1.607 GiB local KV plus 12 KiB state, versus 5.356 GiB for Full-KV: **70.0%** less history storage. Mean peak process-allocated GPU memory falls from 27.40 to 23.63 GiB (13.8%).

After history preparation, LaST continuation takes 15.37 ± 0.59 s versus 19.57 ± 0.96 s for Full-KV (mean$\pm$standard deviation): **21.5%** lower latency and a $1.27\times$ ratio of mean times. Timing includes all eight query readouts and residual injections, VAE decoding, and CPU transfer. It excludes history encoding, state writing and predecoding, both methods' KV construction, and video-file writing. The ratio therefore applies to continuation with prepared memory. End-to-end latency would require timing the excluded preparation steps.

Reader activity during denoising. In seven saved 1.3B runs, the reader produces a nonzero correction at each of the eight calls. The final maximum absolute correction is smaller than the first in every run, although intermediate corrections can increase (Appendix A.5, Figure 7). These traces quantify readout activity rather than output quality.

State capacity. On the full MOSEv2 dataset (Ding et al., 2025), one to eight object slots use 4.5–22 KiB. Figure 6 reports normalized reconstruction error and the fraction of windows whose objects fit within the slot budget. Evaluation includes all objects, including those beyond the budget.

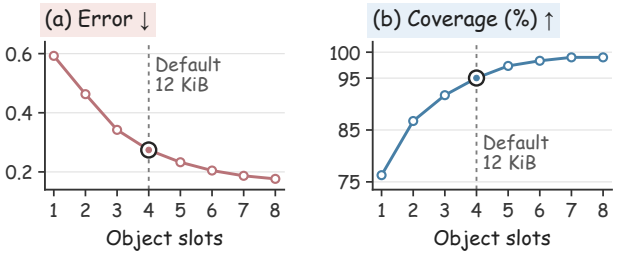

Figure 6: **Capacity on MOSEv2.** Reconstruction error and object coverage. Circled points mark the four-slot default.

The four-slot default uses 12 KiB, with 0.274 error and 95.02% coverage. Eight slots reach 99.00% coverage and 0.176 error. Coverage saturates at seven slots; an eighth reduces error from 0.187 to 0.176. The default therefore trades reconstruction accuracy for a smaller state. Coverage measures whether objects fit within the slot budget, whereas reconstruction error measures how accurately their content is reconstructed.

5 Conclusion

LaST retains scene, geometry, and appearance information beyond local attention and retrieves it to guide a frozen generator. Future reconstruction and multi-window generation training connect history retention to its use during denoising. Evaluation on the full WRBench set shows improvements in all five metrics at both 1.3B and 5B over both KV settings. Seven-scene efficiency measurements show 70.0% lower history storage and 21.5% lower continuation latency after memory preparation. The MOSEv2 ablation relates reconstruction quality and object coverage to memory capacity. These findings support a persistent state that makes earlier object information available to later video generation.

AI USE STATEMENT

We used generative AI tools to discuss and refine the memory architecture, training objectives, and mathematical formulation. AI assistance also supported literature discovery, manuscript drafting and revision, translation, and scientific figure design and preparation. The authors take responsibility for the final content of this work, including the validity and originality of the methods, the accuracy of the references and mathematical statements, and the implementation and experimental evidence supporting its claims.

REFERENCES

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15619–15629, 2023. URL https://arxiv.org/abs/2301.08243.

Alexei Baevski, Wei-Ning Hsu, Qiantong Xu, Arun Babu, Jiatao Gu, and Michael Auli. data2vec: A General Framework for Self-supervised Learning in Speech, Vision and Language, 2022. URL https://arxiv.org/abs/2202.03555.

Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, Varun Jampani, and Robin Rombach. Stable Video Diffusion: Scaling Latent Video Diffusion Models to Large Datasets, 2023. URL https://arxiv.org/abs/2311.15127.

Jake Bruce, Michael Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, Yusuf Aytar, Sarah Bechtle, Feryal Behbahani, Stephanie Chan, Nicolas Heess, Lucy Gonzalez, Simon Osindero, Sherjil Ozair, Scott Reed, Jingwei Zhang, Konrad Zolna, Jeff Clune, Nando de Freitas, Satinder Singh, and Tim Rocktäschel. Genie: Generative Interactive Environments, 2024. URL https://arxiv.org/abs/2402.15391.

Boyuan Chen, Diego Marti Monso, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion Forcing: Next-token Prediction Meets Full-Sequence Diffusion, 2024. URL https://arxiv.org/abs/2407.01392.

Kaijin Chen, Dingkang Liang, Xin Zhou, Yikang Ding, Xiaoqiang Liu, Pengfei Wan, and Xiang Bai. Out of Sight but Not Out of Mind: Hybrid Memory for Dynamic Video World Models, 2026. URL https://arxiv.org/abs/2603.25716.

Henghui Ding, Kaining Ying, Chang Liu, Shuting He, Xudong Jiang, Yu-Gang Jiang, Philip HS Torr, and Song Bai. MOSEv2: A more challenging dataset for video object segmentation in complex scenes. *arXiv preprint arXiv:2508.05630*, 2025. URL https://arxiv.org/abs/2508.05630.

Yanbo Ding, Zhizhi Guo, Quanyue Song, Yishan He, Zhixiang He, Yongxiang Li, and Yali Wang. Ripple: Real-Time Streaming Audio-Video Generation With Cross-Modal Recurrent Memory, 2026. URL https://arxiv.org/abs/2607.26818.

Weijia Dou, Hui Li, Jiahao Cui, Lei Zhou, Jingdong Wang, and Siyu Zhu. SlotMemory: Object-Centric KV Memory for Streaming Long-Video Generation, 2026. URL https://arxiv.org/abs/2605.31033.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning Latent Dynamics for Planning from Pixels, 2019. URL https://arxiv.org/abs/1811.04551.

Guangzhao He, Hadar Averbuch-Elor, and Wei-Chiu Ma. Can 4D Foundation Models Remember?, 2026. URL https://arxiv.org/abs/2609.20819.

Roberto Henschel, Levon Khachatryan, Hayk Poghosyan, Daniil Hayrapetyan, Vahram Tadevosyan, Zhangyang Wang, Shant Navasardyan, and Humphrey Shi. StreamingT2V: Consistent, Dynamic, and Extendable Long Video Generation from Text, 2025. URL https://arxiv.org/abs/2403.14773.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video Diffusion Models, 2022. URL https://arxiv.org/abs/2204.03458.

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self Forcing: Bridging the Train-Test Gap in Autoregressive Video Diffusion, 2025. URL https://arxiv.org/abs/2506.08009.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. VBench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21807–21818, June 2024.

Wayne King, Zeyue Xue, Yuxuan Bian, Jie Huang, Haoran Li, Yaowei Li, Yaofeng Su, Yuming Li, Haoyu Wang, Shiyi Zhang, Songchun Zhang, Yuwei Niu, Sihan Xu, Junhao Zhuang, Haoyang Huang, and Nan Duan. Echo-Memory: A Controlled Study of Memory in Action World Models, 2026. URL https://arxiv.org/abs/2606.09803.

Thomas Kipf, Gamaleldin F. Elsayed, Aravindh Mahendran, Austin Stone, Sara Sabour, Georg Heigold, Rico Jonschkowski, Alexey Dosovitskiy, and Klaus Greff. Conditional Object-Centric Learning from Video, 2022. URL https://arxiv.org/abs/2111.12594.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4015–4026, 2023. URL https://openaccess.thecvf.com/content/ICCV2023/html/Kirillov_Segment_Anything_ICCV_2023_paper.html.

Dacheng Li, Yunhao Fang, Yukang Chen, Shuo Yang, Shiyi Cao, Justin Wong, Michael Luo, Xiaolong Wang, Hongxu Yin, Joseph E. Gonzalez, Ion Stoica, Song Han, and Yao Lu. WorldModelBench: Judging video generation models as world models, 2025. URL https://arxiv.org/abs/2502.20694.

Kunyang Li, Mubarak Shah, and Yuzhang Shang. Attend Locally, Remember Linearly: Linear Attention as Cross-Frame Memory for Autoregressive Video Diffusion, 2026. URL https://arxiv.org/abs/2605.16579.

Yilai Liu, Xin Zhang, Shiyuan Zhang, and Hongyang Du. SlotMem: Character-Addressable Internal Memory for Narrative Long Video Generation, 2026. URL https://arxiv.org/abs/2607.15772.

Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-Centric Learning with Slot Attention, 2020. URL https://arxiv.org/abs/2006.15055.

Jinpeng Lu, Dexu Zhu, Haoyuan Shi, Linghan Cai, Guo Tang, Yinda Chen, Jie Cao, Duyu Tang, Yi Zhang, Yong Dai, and Xiaozhu Ju. Current world models lack a persistent state core, 2026. URL https://arxiv.org/abs/2606.20545.

Yingmao Miao, Pengfei Zhang, Chaoran Xu, Meng Yu, Jing Tang, Xiangxiang Chu, Chao Shen, and Chenhao Lin. Do Video Generators Track the World Across Segments? A Benchmark and Method for World-State Reasoning in Video Continuation, 2026. URL https://arxiv.org/abs/2609.03673.

Malte Mosbach, Jan Niklas Ewertz, Angel Villar-Corrales, and Sven Behnke. SOLD: Slot Object-Centric Latent Dynamics Models for Relational Manipulation Learning from Pixels, 2025. URL https://arxiv.org/abs/2410.08822.

Duc Nguyen, Sieu Tran, Hao Vo, Khoa Vo, Duy Minh Ho Nguyen, Nghi D. Q. Bui, Anh Nguyen, Long Mai, and Ngan Le. TSA: Temporal Slot Activation for Persistent Object-Centric Video Representation, 2026. URL https://arxiv.org/abs/2606.13714.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision, 2024. URL https://arxiv.org/abs/2304.07193.

Ryan Po, Yotam Nitzan, Richard Zhang, Berlin Chen, Tri Dao, Eli Shechtman, Gordon Wetzstein, and Xun Huang. Long-Context State-Space Video World Models, 2025. URL https://arxiv.org/abs/2505.20171.

Runjia Qian, Zile Wang, Jihai Zhang, Kai Zou, Wei Yu, Jiaxing Li, Zexiang Liu, Yaokun Li, Fei Kang, Kaichen Huang, Mengyin An, Haobo Zhang, Biao Jiang, Jiahua Wang, Haofeng Sun, Yang Liu, and Yangguang Li. Matrix-Game 3.5: Enhancing Real-Time Streaming Interactive World Models with Patch Memory, 2026. URL https://arxiv.org/abs/2608.29910.

Joonghyuk Shin, Yicong Hong, Jaesik Park, and Xun Huang. Can Video World Models Track Unobserved World States?, 2026. URL https://arxiv.org/abs/2608.30692.

Jiaming Tan, Mingliang Zhai, Zhen Li, Yuwei Wu, Chuanhao Li, and Kaipeng Zhang. AlayaVista: Streaming World Modeling from Panoramic States to Perspective Video, 2026. URL https://arxiv.org/abs/2609.14462.

Aditi Tiwari, Akshit Bhalla, Darshan Prasad, and Heng Ji. Does Video Memory Use What It Retrieves? A Causal Audit of Memory Specificity, 2026. URL https://arxiv.org/abs/2609.12090.

Pavel Tokmakov, Jie Li, and Adrien Gaidon. Breaking the "object" in video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22836–22845, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://papers.nips.cc/paper/7181-attention-is-all-you-need.

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

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, 2026. URL https://arxiv.org/abs/2606.09828.

Zhengxuan Wei, Xu Guo, Xinghui Li, Xunzhi Xiang, Min Wei, Yiran Zhu, Qiulin Wang, Xintao Wang, Pengfei Wan, Xiangwang Hou, and Qi Fan. Geometry-Aware Implicit Memory for Video World Models, 2026. URL https://arxiv.org/abs/2606.02436.

Tong Wu, Shuai Yang, Ryan Po, Yinghao Xu, Ziwei Liu, Dahua Lin, and Gordon Wetzstein. Video World Models with Long-term Spatial Memory, 2025. URL https://arxiv.org/abs/2506.05284.

Xindi Wu, Sven Elflein, James Lucas, Olga Russakovsky, Laura Leal-Taixé, Despoina Paschalidou, Jonathan Lorraine, and Aljoša Ošep. Addressable Memory for Video World Models, 2026. URL https://arxiv.org/abs/2608.07408.

Ziyi Wu, Nikita Dvornik, Klaus Greff, Thomas Kipf, and Animesh Garg. SlotFormer: Unsupervised Visual Dynamics Simulation with Object-Centric Models, 2023a. URL https://arxiv.org/abs/2210.05861.

Ziyi Wu, Jingyu Hu, Wuyue Lu, Igor Gilitschenski, and Animesh Garg. SlotDiffusion: Object-Centric Generative Modeling with Diffusion Models, 2023b. URL https://arxiv.org/abs/2305.11281.

Haocheng Xi, Yiming Xie, Hexu Zhao, Yiwen Zhang, Michael Liu, Thomas Creavin, Kurt Keutzer, Xiuyu Li, Zhaoyang Lv, Chenfeng Xu, and Haiwen Feng. Video DeltaNet: A Video-Native Hybrid Attention for Livestream Video Generation, 2026. URL https://arxiv.org/abs/2609.20744.

Zeqi Xiao, Yushi Lan, Yifan Zhou, Wenqi Ouyang, Shuai Yang, Yanhong Zeng, and Xingang Pan. WorldMem: Long-term Consistent World Simulation with Memory, 2026. URL https://arxiv.org/abs/2504.12369.

Tianshuo Xu, Yichen Xie, Depu Meng, Chensheng Peng, Quentin Herau, Bo Jiang, Yihan Hu, and Wei Zhan. Teaching Video Generators to Remember: Eliciting Dynamic Memory for Out-of-Sight State Evolution, 2026. URL https://arxiv.org/abs/2605.25333.

Linjie Yang, Yuchen Fan, and Ning Xu. The 2nd large-scale video object segmentation challenge - video object segmentation track, October 2019. URL https://youtube-vos.org/dataset/vos/.

Shuai Yang, Wei Huang, Ruihang Chu, Yicheng Xiao, Yuyang Zhao, Xianbang Wang, Muyang Li, Enze Xie, Yingcong Chen, Yao Lu, Song Han, and Yukang Chen. LongLive: Real-time Interactive Long Video Generation, 2025. URL https://arxiv.org/abs/2509.22622.

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, 2025. URL https://arxiv.org/abs/2506.03141.

Lvmin Zhang, Shengqu Cai, Muyang Li, Gordon Wetzstein, and Maneesh Agrawala. Frame Context Packing and Drift Prevention in Next-Frame-Prediction Video Diffusion Models, 2025. URL https://arxiv.org/abs/2504.12626.

Wenqu Zhao, Xuemin Chi, Xin Zhang, Guoqing Ma, Baorun Li, Jianjie Fang, Peizhi Tang, Chen Gao, and Wei Wu. DensityKV: Density-Guided KV Cache Compression for Long Video Generation, 2026. URL https://arxiv.org/abs/2608.27922.

Dian Zheng, Ziqi Huang, Hongbo Liu, Kai Zou, Yinan He, Fan Zhang, Lulu Gu, Yuanhan Zhang, Jingwen He, Wei-Shi Zheng, Yu Qiao, and Ziwei Liu. VBench-2.0: Advancing video generation benchmark suite for intrinsic faithfulness, 2025. URL https://arxiv.org/abs/2503.21755.

A Appendix

A.1 State decoding and readout details

The memory comprises scene records $B_t \in \mathbb{R}^{G\times d}$, geometry records $K_t \in \mathbb{R}^{N\times d}$, and appearance records $A_t \in \mathbb{R}^{NR\times d}$. Each object i pairs one geometry record $k_i \in \mathbb{R}^d$ with an appearance bank $A_i \in \mathbb{R}^{R\times d}$. We use $(N, R, G, d) = (4, 4, 4, 256)$, so the persistent state contains 24×256 values, or 12,288 bytes (12 KiB) in BF16. Local KV, model parameters, and temporary decoded fields are stored separately. A segment $Z_t \in \mathbb{R}^{L\times C\times H\times W}$ provides L latent frames with C channels on an $H \times W$ grid.

Writing observations to memory. Both writers start from the preceding state and process the segment frame by frame. The structural branch supplies scene and geometry records, while the appearance branch uses the structural branch's detached assignments to write visual content. After merging their outputs as in Section 3.1, we discard intermediate states and assignment maps.

For a latent token z_j at space–time coordinate c_j, we form the addressing feature $o_j = W_z \,\mathrm{LN}(z_j) + W_c c_j$. Each record queries these features using a learned slot embedding and its preceding value. Scene attention is normalized over observations. Geometry attention first assigns observations across object slots and then normalizes each slot's weights over observations. Appearance attention also uses the object/background ownership probabilities p_{ij} to select content for object i:

$$\omega_{h,j} = \rho_i \,\mathrm{softmax}_j\big(\ell_{h,j} + \log\max(p_{ij}, 10^{-8})\big)\,, \qquad \rho_i = \max_{j\in\mathcal{O}} p_{ij}, \tag{10}$$

Here $\ell_{h,j}$ is the attention logit for head h, and $\mathcal{O}$ is the set of available observations. Unavailable observations are masked before normalization. For appearance records, ownership therefore determines both which observations are read and how strongly they replace stored content. Scene and geometry writes use their respective attention weights without this ownership factor.

Within either branch, the attended observations form a candidate for each record. Let s denote its preceding value and $v_{h,j}$ a projected observation. We compute $\widetilde{s} = b + \mathrm{MLP}(b)$ from $b = \mathrm{LN}(s + W_{\mathrm{write}}\,\mathrm{Concat}_h[\sum_j \omega_{h,j} v_{h,j}])$. The total attention mass, averaged over H_a heads and capped at one, controls replacement:

$$\rho = \min\big(1, H_a^{-1}\textstyle\sum_h \sum_j \omega_{h,j}\big), \qquad s^+ = (1-\rho)s + \rho\widetilde{s}. \tag{11}$$

Records start at $s = 0$, with learned queries serving to select observations rather than initialize stored content. For appearance records, $\rho = \rho_i$, so weak ownership favors the preceding value. Stacking the updated records into $\widetilde{S}$, we obtain the state for the next frame:

$$S' = P_{\mathrm{anchor}}\widetilde{S} + P_{\mathrm{self}}\,\mathrm{LN}(\widetilde{S}) + \mathrm{Attn}_{\mathrm{cross}}(\mathrm{LN}(\widetilde{S})). \tag{12}$$

The first two terms transform each record independently, providing an observation anchor and a self residual. The third term exchanges information across objects: an object's query excludes its own geometry and appearance records, while scene queries can attend to all records. A record with zero write mass is unchanged by Eq. (11), but can still evolve through Eq. (12).

Decoding geometry and appearance. The core decoder maps each object record to $\kappa_i = (\tau_i, \mathbf{p}_i, \mathbf{e}_i, \mathbf{v}_i, o_i)$. A normalized MLP predicts a residual relative to the record's repeated-core representation. The input record is detached during address alignment, whereas joint generation training allows gradients through the current state and decoded field. At query time τ, the object center is extrapolated as $\mathbf{p}_i(\tau) = \mathbf{p}_i + (\tau - \tau_i)\mathbf{v}_i$, while the decoded observation remains fixed.

Appearance is queried relative to this center and the object's size. We bound the absolute half-extents elementwise by $\widetilde{\mathbf{e}}_i = \max(|\mathbf{e}_i|, 10^{-6})$. For $q = (\tau, \mathbf{u})$, the relative coordinates in Eq. (2) are $q_i = (\tau - \tau_i, (\mathbf{u} - \mathbf{p}_i(\tau)) \oslash \widetilde{\mathbf{e}}_i)$, where $\oslash$ denotes elementwise division. The field $f_i(q) = \mathcal{F}_\psi(\gamma(q_i), A_i)$ uses coordinates as attention queries and appearance records as values. Cross-attention updates accumulate in a separate content stream initialized at zero, keeping query coordinates separate from the retrieved appearance. Geometry also determines where each object field contributes, through the logit $b_i(p) = \beta \min_{a\in\{x,y\}}(|e_{i,a}| - |u_a - p_{i,a}(\tau)|) + o_i$ and weights

$$\pi_i(p) = \frac{\exp b_i(p)}{1 + \sum_{k\in\mathcal{J}} \exp b_k(p)}, \qquad \pi_{\mathrm{sc}}(p) = \frac{1}{1 + \sum_{k\in\mathcal{J}} \exp b_k(p)}. \tag{13}$$

Here $\mathcal{J}$ contains the active object slots and β controls the address temperature. Invalid slots receive zero weight. The remaining mass is assigned to null in the object field. In the generator's reader, it instead supplies the prior for the scene field, obtained by spatial interpolation of the scene atlas. The reader also has a separate zero-valued null candidate.

Reading a correction. The reader compares scene and object fields with the generator's current clean estimate. At position $p = (\tau, \mathbf{u})$, each field proposes the residual $r_j(p) = f_j(p) - \widehat{x}_0(p)$, while the null candidate proposes $r_\varnothing = 0$. The combined correction is

$$\Delta(p) = m(p) \sum_h W_{O,h}\big[\textstyle\sum_j a_{h,j}(p) W_{V,h} r_j(p)\big]. \tag{14}$$

The mask m excludes the conditioning frame and inactive reads. Each head has a value projection $W_{V,h}$ and an output projection $W_{O,h}$, the corresponding column block of W_O. Summing the projected outputs is equivalent to concatenating the heads and applying W_O; the value and output projections are learned separately.

Attention combines content compatibility with the geometric priors. For head width d_h, the logit is $\ell_{h,j} = q_h^\top k_{h,j}/\sqrt{d_h}$ and $\mathbf{a}_h = \mathrm{softmax}(\boldsymbol{\ell}_h + \log \boldsymbol{\pi})$. Candidates with zero prior are masked, and the extra null candidate has unit prior. Queries use $q_h = W_{Q,h}\,\mathrm{LN}(\widehat{x}_0)$, while keys use $k_{h,j} = W_{K,h}\,\mathrm{LN}(f_j) + e_{h,j}$, with a scene or object role vector $e_{h,j}$. The null field and role are zero. Non-affine normalization and bias-free projections therefore give $k_{h,\varnothing} = 0$ and preserve its zero value.

With optional causal-anchor conditioning, the query input is the average of the normalized current estimate and a normalized anchor feature. The anchor is the spatial mean of the conditioning latent, repeated at each query position. In either case, keys describe the decoded fields and values express their residuals relative to $\widehat{x}_0$. The conditioning frame always has zero read support. Reads at other denoising steps may be disabled by the schedule without changing the stored state.

A.2 Readout rescaling

The reader's correction can change the spatial scale of a latent prediction. We rescale the corrected estimate $y = \widehat{x}_0 + \Delta$ using the base prediction's standard deviation, computed separately over spatial positions in each frame and channel. For $s(y) > \varepsilon$, this gives

$$\bar{y} = \mathcal{C}_{\widehat{x}_0}(y) = \mu(y)\mathbf{1} + \frac{s(\widehat{x}_0)}{s(y)}\big(y - \mu(y)\mathbf{1}\big). \tag{15}$$

We apply the rescaled prediction only where the reader produces a nonzero correction: $\widehat{x}_0^{\mathrm{mem}} = \widehat{x}_0 + \chi \odot (\bar{y} - \widehat{x}_0)$, with $\chi = \mathbf{1}[\Delta \neq 0]$. The complete correction in Eq. (3) is therefore $\mathcal{R}_\phi(b, S) = \chi \odot (\mathcal{C}_b(b + \Delta_\phi) - b)$, where $b = \widehat{x}_0$. This notation includes the parameter-free rescaling. Each memory state has its own correction Δ_ϕ and support χ. Substituting the result into $v^{\mathrm{mem}} = (x_\sigma - \widehat{x}_0^{\mathrm{mem}})/\sigma$ recovers the flow relation for $\sigma > 0$.

For a frame–channel slice $u \in \mathbb{R}^P$, the statistics are $\mu(u) = P^{-1}\mathbf{1}^\top u$ and $s(u) = \|u - \mu(u)\mathbf{1}\|_2/\sqrt{P-1}$. Before masking, Eq. (15) preserves the corrected mean $\mu(y)$ and matches the base standard deviation $s(\widehat{x}_0)$ whenever $s(y) > \varepsilon$.

Masking can change these global moments, but preserves the base prediction at every coordinate with $\chi_p = 0$. Consequently,

$$\mathrm{supp}(\widehat{x}_0^{\mathrm{mem}} - \widehat{x}_0) \subseteq \mathrm{supp}(\Delta), \qquad \Delta = 0 \implies \widehat{x}_0^{\mathrm{mem}} = \widehat{x}_0. \tag{16}$$

We set $\varepsilon = 10^{-6}$ and use a scale factor of one when $s(y) \leq \varepsilon$, giving $\bar{y} = y$. Disabled and all-zero reads return the original prediction directly, avoiding an unnecessary conversion between flow and latent predictions.

A.3 Training objectives

State pretraining (R). We train the state to reconstruct future object content from a partially observed history. The writer sees a causal prefix, while subsequent clean frames provide training targets. To simulate an object becoming unavailable, we retain its earliest visible observation and mask its annotated latent locations in later prefix frames, leaving the target frames unchanged. Object IDs determine slot assignments during training and are absent at inference.

For each object visible in the prefix, target masks provide normalized centers and half-extents. We fit velocity by least squares from the anchor center to visible future centers and define future visibility as the fraction of target frames containing the object. To form appearance targets $A^\star$, we resize visible object-relative latent crops to a common grid, average them, and pack the result into records. Background targets pool latent values over the scene grid outside object masks. These targets are detached. A state built from the last available observations supplies a causal anchor target, with smooth-L_1 supervision on the scene and geometry components of the anchor, the predicted residual, and their sum.

The teacher-guided read isolates appearance learning from errors in predicted geometry. For a query belonging to object i, the teacher supplies $T_i^{\mathrm{teach}}(q)$ from target geometry and assigns unit weight to

that object. All other objects receive zero weight, as do all objects at a null query. The read $\mathcal{D}^{\text{teach}}$ in Eq. (4) retrieves the student's appearance records at these supplied coordinates. By contrast, $\mathcal{D}^{\text{pred}}$ combines fields using the state's predicted geometry. For either read $\mathcal{D}$, the reconstruction error on continuation U is

$$E(S;U) = \frac{1}{C|\mathcal{Q}_{\text{obj}}|} \sum_{q \in \mathcal{Q}_{\text{obj}}} \|\mathcal{D}(q;S) - z_q\|_2^2. \tag{17}$$

Here $\mathcal{Q}_{\text{obj}}$ denotes the main text's $\mathcal{Q}_U$: valid object queries at interior and boundary locations, each paired with its annotated identity. Queries are weighted equally within a video, and averaging $E(S;U)$ across clips gives $\mathcal{L}_{\text{rec}}(\mathcal{D})$ in Eq. (5). Object/null cross-entropy supervises query addressing, while prefix ownership supervises the shared write assignments. Teacher geometry and assignments remain detached, so the teacher-guided reconstruction trains the student's appearance writer and field decoder.

Reconstruction alone can favor appearance common to many videos. To retain details specific to each history, we also match differences between appearance records to differences between their targets. For two histories with records A, A^- and future-crop targets A^*, A^{*-}, we use

$$\mathcal{L}_{\text{pair}} = \text{SmoothL1}_{\mathcal{I}}\big(A - A^-, \text{sg}(A^* - A^{*-})\big). \tag{18}$$

The set $\mathcal{I}$ selects record elements valid in both videos, and smooth-L_1 has transition point one. Subtraction removes components common to the two states, focusing supervision on their differences. The comparison pairs record positions without requiring shared object identities. This term has unit weight during joint representation refinement in stage R.

To make these differences matter at readout, we compare reconstruction under the correct and an unrelated history. Holding U fixed, we replace S with S^-. For either read path, let $E^+ = E(S;U)$ and $E^- = E(S^-;U)$ denote their errors, with $E^0 = E(0;U)$ as the no-memory reference. The history loss requires separation proportional to the benefit of correct memory:

$$\mathcal{L}_{\text{hist}} = \left[\text{sg}\big(E^+ + \eta[E^0 - E^+]_+\big) - E^-\right]_+. \tag{19}$$

Here $[a]_+ = \max(a, 0)$ and $\eta = 0.8$. The correct-memory and zero-state errors define a detached threshold. This penalty therefore updates the unrelated read, while reconstruction trains the correct read. We compute the losses per video and average over valid samples.

A learned appearance bank shared across videos provides a second reference, capturing content available without a specific history. Let E^{global} be its error and E^{teach} the teacher-guided error. The term $[E^{\text{teach}}(S;U) - 0.9\,\text{sg}(E^{\text{global}})]_+$ encourages the video-conditioned state to improve on this shared bank. Stage R first learns structure and ownership, then trains appearance with structure fixed. Address alignment holds the writers fixed, followed by joint refinement of the representation modules. Stages J and L subsequently train state construction and readout with the generation objective in Eq. (9).

Reader warm-up (G). We train the reader to distinguish relevant object memory from unrelated appearance. For each clip, we select an object observed in the prefix and visible at valid target locations. Its appearance bank A_i is replaced by a valid bank of the same shape from another video, whose object identity may differ. Scene records, geometry records, and all other appearance banks remain fixed. Clips without an eligible object contribute original-memory reconstruction losses only. Both memory states are detached during stage G, so optimization is restricted to the reader. Joint training in stages J and L subsequently allows gradients through the fields and state construction (Section 3.4).

At a noisy input sampled from the original-memory rollout, let b be the frozen generator's clean prediction. Reading the original and replaced memories gives y^+ and y_i^-, respectively, each with its own rescaling and support restriction from Appendix A.2. The two reads share the noisy input, timestep, base prediction, and conditioning. We detach both the sampled input and b, retaining gradients through the current reader evaluations. Relative to the clean target x^*, the missing correction is $e = \text{sg}(x^* - b)$, and the two reader residuals are $r^+ = y^+ - \text{sg}(b)$ and $r_i^- = y_i^- - \text{sg}(b)$. Replacement supplies a comparison for memory use, not a target for transferring the donor's appearance.

Let $\mathcal{V}$ contain valid latent locations excluding the conditioning frame. Annotations define a detached visibility mask M_i: a latent location belongs to the object if any pixel in its annotation footprint is

foreground. The remaining locations form $\Omega_i^{\text{out}} = \{p \in \mathcal{V} : M_i(p) = 0\}$. For a binary spatial mask M, we use $\|z\|_M^2 = \|M \odot z\|_2^2/(C\|M\|_1)$, broadcasting M across the C channels and omitting empty-mask terms. Losses outside the object are computed per video and averaged over valid samples. They are applied after rescaling, so they also penalize nonlocal changes caused by shared spatial statistics, without enforcing exact spatial independence.

The loss $\mathcal{L}_{\text{fit}} = \text{mean}_{\mathcal{V}} |y^+ - x^*|^2$ fits the original-memory prediction to the target. We also measure how well a reader residual points toward the missing correction. At location p, define the channel-averaged energies $E_p = C^{-1}\|e_p\|_2^2$ and $R_p = C^{-1}\|r_p\|_2^2$, and let $\overline{E} = \text{mean}_{p\in\mathcal{V}} E_p$. A stabilized alignment score and its object-region average are

$$\begin{aligned} a_p(r,e) &= \frac{C^{-1}\langle r_p, e_p\rangle}{\sqrt{\max(R_p + 0.01E_p, \epsilon_a)}\sqrt{\max(E_p, \epsilon_a)}}, \\ \mathcal{A}_i(r,e) &= \frac{\sum_{p\in\mathcal{V}} M_i(p) w_p a_p(r,e)}{\sum_{p\in\mathcal{V}} M_i(p) w_p}. \end{aligned} \tag{20}$$

where $w_p = \text{clip}_{[0.05,4]}(E_p/\max(\overline{E}, \epsilon_a))$ gives more weight to larger base errors, and $\epsilon_a = 10^{-12}$ stabilizes zero residuals. With b and x^* fixed, write $a_i^+ = \mathcal{A}_i(r^+, e)$ and $a_i^- = \mathcal{A}_i(r_i^-, e)$. We favor the original memory's alignment inside the object and preserve its prediction outside the object:

$$\mathcal{L}_{\text{rank}} = [\delta_a - a_i^+ + \text{sg}(a_i^-)]_+ + [\delta_a - \text{sg}(a_i^+) + a_i^-]_+, \tag{21a}$$

$$\mathcal{L}_{\text{keep}} = \frac{1}{C|\Omega_i^{\text{out}}|} \sum_{p\in\Omega_i^{\text{out}}} \left\| y_i^-(p) - \text{sg}(y^+(p)) \right\|_2^2. \tag{21b}$$

Let $\mathcal{A}(r,e)$ denote the same weighted alignment average over all of $\mathcal{V}$. The residual objective combines whole-clip fitting and alignment with the object-specific ranking:

$$\begin{aligned} \mathcal{L}_{\text{residual}} = \text{mean}_{p\in\mathcal{V}} &\frac{C^{-1}\|r_p^+ - e_p\|_2^2}{\max(E_p, 0.05\max(\overline{E}, \epsilon_a))} \\ &+ [a_0 - \mathcal{A}(r^+, e)]_+ + \mathcal{L}_{\text{rank}}. \end{aligned} \tag{22}$$

The normalized error fits the original-memory residual, while $a_0 = 0.2$ sets a minimum alignment target. In Eq. (21a), the first hinge term raises original-memory alignment against a fixed replaced-memory reference; the second lowers replaced-memory alignment against a fixed original-memory reference. Both use margin $\delta_a = 0.1$ and are summed without an averaging factor. Paired terms are omitted when $\sum_p M_i(p)E_p \le \epsilon_a\|M_i\|_1$, since a zero target residual carries no directional information.

Ranking alone could be satisfied by damaging the replaced prediction. We discourage this using the object-region errors $\mathcal{E}_i^+ = \text{mean}_{M_i} |y^+ - x^*|^2$, $\mathcal{E}_i^- = \text{mean}_{M_i} |y_i^- - x^*|^2$, and $\mathcal{E}_i^0 = \text{mean}_{M_i} |b - x^*|^2$, averaged over locations and channels. We use

$$\mathcal{L}_{\text{bound}} = \left[\mathcal{E}_i^- - \text{sg}\big(\max(\mathcal{E}_i^0, \mathcal{E}_i^+)\big)\right]_+. \tag{23}$$

This term penalizes replacement errors above the larger of the original-memory and base errors. It constrains reconstruction error without imposing a donor-appearance target.

We supplement these losses with latent-structure supervision: spatial correlation error $\mathcal{L}_{\text{corr}}$, smooth-$L_1$ error after 2×2 spatial pooling $\mathcal{L}_{\text{coarse}}$, the average of first-order temporal and spatial derivative errors $\mathcal{L}_{\text{diff}}$, and log spectral-magnitude error $\mathcal{L}_{\text{spec}}$. Excluding the conditioning frame, they form $\mathcal{L}_{\text{structure}} = \mathcal{L}_{\text{corr}} + 0.25\mathcal{L}_{\text{coarse}} + 0.5\mathcal{L}_{\text{diff}} + 0.25\mathcal{L}_{\text{spec}}$. The full warm-up objective is

$$\mathcal{L}_{\text{read}} = \mathcal{L}_{\text{fit}} + \lambda_r\mathcal{L}_{\text{residual}} + \lambda_s\mathcal{L}_{\text{structure}} + \lambda_k\mathcal{L}_{\text{keep}} + \lambda_b\mathcal{L}_{\text{bound}}. \tag{24}$$

Grouping the original-memory terms into $\mathcal{L}_{\text{base}} = \mathcal{L}_{\text{fit}} + \lambda_r(\mathcal{L}_{\text{residual}} - \mathcal{L}_{\text{rank}}) + \lambda_s\mathcal{L}_{\text{structure}}$ separates them from the intervention losses:

$$\mathcal{L}_{\text{read}} = \mathcal{L}_{\text{base}} + \lambda_r\mathcal{L}_{\text{rank}} + \lambda_k\mathcal{L}_{\text{keep}} + \lambda_b\mathcal{L}_{\text{bound}}. \tag{25}$$

We use $\lambda_r = 1$ and $\lambda_s = 0.1$, with λ_k and λ_b weighting the intervention penalties. Sampling one eligible object per clip adds one reader evaluation at the shared base prediction. Training queries are drawn from denoising steps where the reader is active at inference. After warm-up, stages J and L use Eq. (9); its local-preservation term encourages the reader to abstain from unsupported corrections under unrelated memory. Inference uses the original state with the same reader and state size, requiring neither a donor nor annotation masks.

A.4 Training Schedule and Reproducibility

Training sources and splits. MOSEv2 provides complex occlusion and reappearance, VOST provides object transformations, and YouTube-VOS 2019 broadens appearance and category coverage. We construct training and internal development sets from their official training splits. Original recordings, subject or kitchen groups where available, and near-duplicate clips define source families. A deterministic family-level split targets 90% training and 10% development, keeping related windows and derivatives together. Families with unresolved membership or overlap with evaluation sources are excluded from training. Official validation and test splits retain their original identities.

Unified episodes. Dataset-specific adapters produce a common JSONL record with source identity, split, original frame indices, timestamps, object annotations, and preprocessing metadata. The loader returns observed inputs and supervision separately. An anchor contains 81 consecutive RGB frames: offsets 0–40 form the history, and 41–80 form the prediction targets. The two 41-frame generation windows share frame 40, which is excluded from the future-target loss. Candidate anchors have stride 40, with at most eight per video. RGB frames and annotations share an aspect-preserving spatial transform to 832×480, with padded pixels excluded from the loss. Each generator uses its own frozen VAE. Original timing is retained when known; missing timestamps and masks remain unknown rather than being replaced by a common frame rate or object-absence labels. Unknown camera poses are handled separately from identity poses. Object IDs are scoped by dataset and source video and supply supervision rather than inference-time assignments.

Sampling and optimization. Each block of 40 anchor draws contains 20 MOSEv2, eight VOST, and 12 YouTube-VOS episodes. Within each source, sampling proceeds uniformly over families, then videos, then eligible windows, with replacement. A mixed cycle comprises 20,000 anchor draws, or 2,500 updates with an effective batch of eight; it does not denote one complete traversal of the distinct windows. The window-construction caps are 5,000, 2,000, and 3,000 for the three sources. These caps and the stage budgets in Table S1 specify upper limits, not realized sample counts or stopping steps. We use AdamW with $(\beta_1, \beta_2) = (0.9, 0.95)$ and $\epsilon = 10^{-8}$. Learning rates warm up over the first 5% of each stage, then follow cosine decay to 10% of their peak values. Each parameter group's gradient norm is clipped at 1.0, and bias and normalization parameters are excluded from weight decay. Trainable parameters and optimizer states use FP32. The frozen generator, state forward computation, and persistent storage use BF16. Development checks occur every 1,250 updates, with a patience of six checks.

Table S1: Stage-specific optimization. R: state pretraining; G: reader warm-up; J: joint training; L: long-rollout training. Cycle and update counts are budget caps; learning rates are peak values.

Stage	Cycles	Updates	Memory LR	Reader LR	Decay
R	10	25,000	3×10^{-4}	–	0.01
G	2	5,000	–	3×10^{-4}	0.00
J	10	25,000	1×10^{-5}	1×10^{-4}	0.01
L	5	12,500	1×10^{-5}	1×10^{-4}	0.01

Gradient flow and supervision. In stages J and L, generation gradients pass through the current reader, field decoder, and two most recent state updates. Earlier states and preceding denoising steps are detached, while Wan, the VAE, and the text encoder remain frozen. We sample training queries from steps where the reader is active at inference and execute the remaining steps without retaining their computation graph. The conditioning latent receives neither read corrections nor generation-loss supervision.

Teacher and student predictions share the noisy latent, timestep, controls, and history, differing in how they access that history. The detached mask in Eq. (7) selects where prediction matching applies. Equation (8) normalizes over all valid locations and latent channels, so an empty mask contributes zero loss and sparse support is not amplified. Matching takes place in flow space: at fixed x_σ and $\sigma > 0$, the relation $x_0 = x_\sigma - \sigma v$ gives $\|\Delta v\|_2^2 = \|\Delta x_0\|_2^2/\sigma^2$. An equivalent loss on clean predictions must include this noise-dependent scaling. The clean-write endpoint $\sigma = 0$ is excluded. When reliable targets are available, we sample forward-noising and rollout queries in equal proportion. For student-generated histories, unaligned future pixels are excluded from supervision, and target-based teacher masks are used only on aligned examples.

Episodes and replay. Long-rollout training uses 81-frame episodes for its first 20% of updates, then samples lengths of 81, 161, and 321 frames in a 50:30:20 ratio within each source. These lengths correspond to two, four, and eight overlapping generation windows. Eligible episodes retain continuous source frames without repetition or cross-shot concatenation. Completed clean outputs update the state carried across windows. Joint training begins with real histories, then increases the fraction of student-generated histories from 25% to 50%. Failure replay occupies 25% of each source's quota, with the remaining 75% drawn from its base training pool. When a source has no replay examples, its base pool fills that quota. The replay pool contains deduplicated generated training episodes and stays fixed within each scheduled round. Negative states come from a different training family. Development and test examples are excluded from both pools.

We group sources, scenes, and near-duplicate content before splitting the data. Adjacent windows, camera variants, and synthetic derivatives of one source stay in the same split, as do all windows of an episode. Future images, masks, and identities supply training targets but never enter observation updates or inference inputs. Text and camera conditions use information available at inference. WRBench evaluation outputs and test scores are excluded from gradient updates, replay selection, and hyperparameter selection.

A.5 History Storage and Reader Activity

State layout. Figure 3 shows how the state is divided among scene, geometry, and appearance records. Each record has 256 BF16 values and uses 0.5 KiB. Four scene records take 2 KiB. Each object has one geometry record and four appearance records, adding 2.5 KiB per slot. The total for N slots is $2 + 2.5N$ KiB, with appearance forming the largest share. New segments update these records in place, so scene and object history persist across generation windows. Adding a segment does not add records: the number of slots determines state storage even as the generated video grows longer.

Doubling the object budget from four to eight slots increases storage from 12 to 22 KiB. The 2 KiB scene component is shared, while geometry and appearance records scale with the represented objects. Capacity can therefore be assigned to additional objects without extending the recent frame history supplied to local attention.

Memory cost and use. Figure 7 summarizes history storage and reader activity. In the efficiency study, LaST uses 1.607 GiB of local KV plus 12 KiB of state, compared with 5.356 GiB for Full-KV. This is a 70.0% reduction in history storage, rather than total GPU memory. The state contribution is shown by annotation because it is too small to distinguish on the GiB axis. The lower panel plots the maximum absolute correction at each of eight reader calls, retaining all 56 values from seven saved runs. Gray curves show individual runs, and the blue curve is their arithmetic mean.

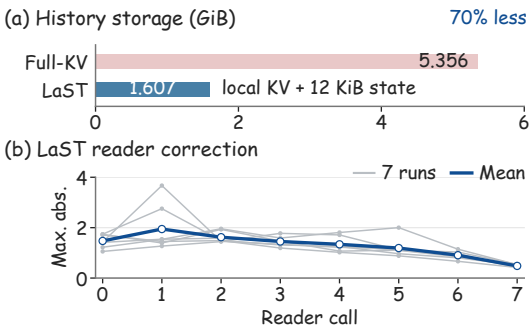

Figure 7: **History storage and readout (1.3B).** Gray: seven saved runs; blue: their mean.

The reader produces nonzero corrections at all recorded calls. In every run, the final maximum absolute correction is smaller than the first, although intermediate corrections can increase. Storage measurements and reader traces were collected separately, preventing a per-scene comparison between history cost and correction magnitude. The traces quantify the reader's changes to the denoising prediction. They describe readout activity rather than attention weights or an independent measure of output quality.

A.6 Object-Level and Internal-State Diagnostics

Masks and external descriptors. SAM ViT-B (Kirillov et al., 2023) annotates the main subject in thirteen sequences from boxes and foreground points. We select the eligible mask with highest predicted IoU and keep the component containing the point. Nearest-neighbor projection into the displayed viewport defines visibility; subject correspondence is manual. Mask-weighted pooling

of the 64×64 SAM feature grid gives a 256-dimensional descriptor f_t. Figure 8 plots its cosine similarity $\langle f_t, f_0 \rangle / (\|f_t\|_2 \|f_0\|_2)$ to the initial descriptor. These source-frame descriptors remain available outside the displayed viewport and are separate from LaST's state. They refer to the original source sequences, not the independently sampled references in Appendix A.8; masks and identities are not transferred between these videos.

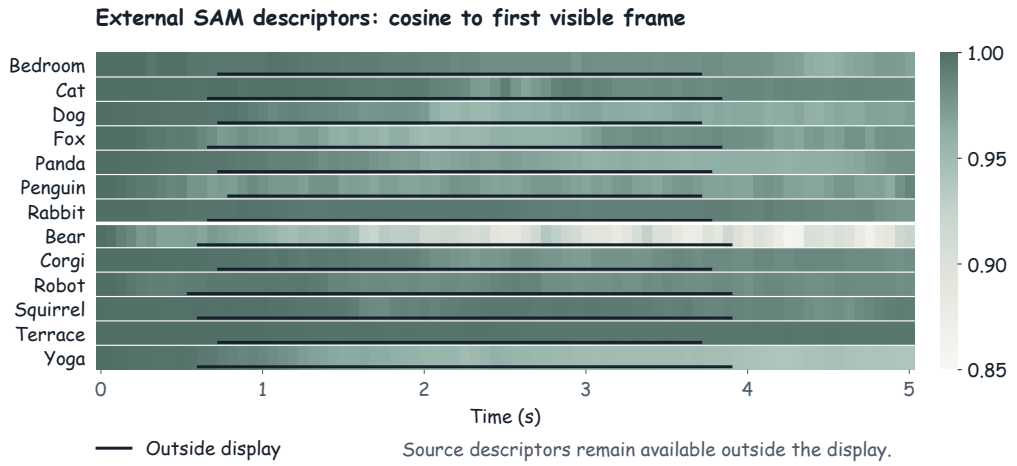

Figure 8: **External descriptor similarity.** Thirteen sequences; color encodes cosine similarity to the initial SAM descriptor. Dark marks indicate intervals outside the displayed viewport.

Saved state. Figure 9 expands the seven 1.3B runs in Figure 7: final 24×256 BF16 states and all eight readout corrections. Records follow storage order, without assigned subject identities.

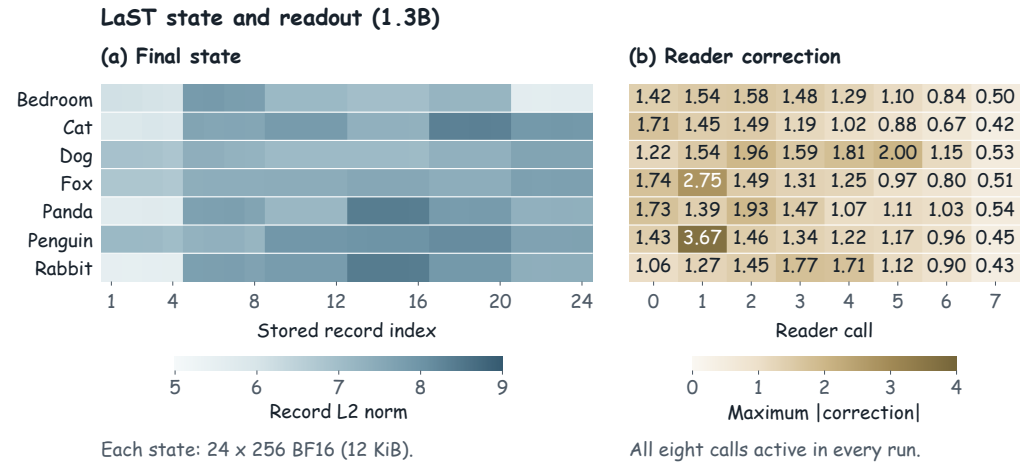

Figure 9: **LaST state and readout (1.3B).** Record norms (left) and maximum read corrections (right), with shared scene rows.

A.7 State Interventions and Multi-round Generation

We test how LaST-5B uses, revises, and carries state across continuations on MOSEv2 and VOST, separately from WRBench. We report spatially matched DINOv2 (Oquab et al., 2024) object-feature cosine similarity and full-frame MSE, averaging valid frames within each sequence before averaging sequences.

Historical state interventions. With the starting observation fixed, Table S2 compares absent, mismatched, and static state and removal of target or other-object records. Object similarity uses five valid frames across four MOSEv2 sequences and eleven across VOST; MSE uses twelve per dataset. Each dataset retains four source sequences. Feature agreement does not verify instance identity, and MSE is sensitive to displacement and viewpoint.

Table S2: **History-conditioned continuation (5B).** Object similarity and full-frame MSE on four sequences per dataset. Bold marks the best value in each column.

	MOSEv2		VOST	
Method / state	Obj. sim.↑	MSE↓	Obj. sim.↑	MSE↓
Short-KV	0.432	0.1123	0.613	0.0381
Full-KV	0.331	0.1126	0.348	0.0504
No state	0.620	0.1169	**0.856**	0.0289
Mismatched state	0.303	0.1199	0.499	0.0430
Static state	0.441	0.1137	0.824	0.0308
Target records removed	0.327	0.1126	0.341	0.0506
Other-object records removed	0.401	0.1117	0.338	0.0508
LaST	**0.840**	**0.0281**	0.816	**0.0126**

LaST exceeds Full-KV on both datasets. Replacing its state with mismatched content lowers similarity to 0.303 and 0.499, indicating that continuation depends on retained history. On VOST, omitting state raises similarity from 0.816 to 0.856 while increasing pixel error from 0.0126 to 0.0289. The two metrics therefore capture different aspects of continuation quality. Removing target or other-object records produces similar scores on VOST, providing weaker evidence for object-specific control.

Re-observation updates. We update LaST's state from the new observation or retain the earlier state under the same current observation. Table S3 reports four sequences per dataset, with eleven valid object frames and twelve MSE frames in each dataset. Updating the state improves object similarity and lowers pixel error on both datasets.

Table S3: **Re-observation (5B).** Updated and retained earlier states share the current observation; four sequences per dataset.

	MOSEv2		VOST	
Method / state	Obj. sim.↑	MSE↓	Obj. sim.↑	MSE↓
Retained earlier state	0.409	0.0894	0.239	0.0512
LaST: updated state	**0.600**	**0.0845**	**0.343**	**0.0458**

Multi-round generation. We generate four successive continuations, carrying and updating state across rounds. Each continuation starts from the preceding output's final frame. From the second round onward, each method conditions on its own generated history. The comparison thus evaluates complete generation chains, whose starting images diverge across methods, rather than isolated reads from a shared observation. A round denotes a continuation segment; camera departures and returns are evaluated separately below.

Table S4: **Four-round continuation (5B).** Two sequences per dataset. Each cell gives object similarity ↑ / full-frame MSE ↓.

Dataset	Method	Round 1	Round 2	Round 3	Round 4
MOSEv2	Short-KV	0.769 / 0.0913	0.254 / 0.1405	0.124 / 0.1602	0.145 / 0.1490
	Full-KV	0.593 / 0.0963	0.239 / 0.1235	0.162 / 0.1443	0.135 / 0.1535
	LaST	**0.800 / 0.0618**	**0.837 / 0.0673**	**0.825 / 0.0678**	**0.732 / 0.0828**
VOST	Short-KV	0.613 / 0.0326	0.334 / 0.0434	0.278 / 0.0721	0.206 / 0.0934
	Full-KV	0.441 / 0.0451	0.216 / 0.1005	0.158 / 0.1552	0.131 / 0.1656
	LaST	**0.905 / 0.0170**	**0.902 / 0.0134**	**0.705 / 0.0274**	**0.667 / 0.0154**

Camera-return examples. Figure 11 shows the final frame of each round in both supplied camera sequences, using the same frame index for every method. At the fourth return, mean historical-object ROI similarity is 0.770 for LaST, 0.145 for Short-KV, and 0.098 for Full-KV; full-frame MSE is 0.0049, 0.0885, and 0.0697, respectively. These two-sequence results characterize appearance at the sampled returns. Interior-frame ROI similarity is 0.179, 0.144, and 0.141, but higher similarity in a fixed historical region is not necessarily desirable while the camera is turned away.

Figure 10 and Table S4 show that LaST retains higher object similarity throughout the four rounds. By round four, scores are 0.732 on MOSEv2 and 0.667 on VOST, versus 0.145/0.206 for Short-KV and 0.135/0.131 for Full-KV. Across the four sequences, the mean is 0.700 for LaST, 0.176 for Short-KV, and 0.133 for Full-KV. LaST still drifts: its VOST similarity falls from 0.905 in the first round to 0.667 in the fourth. There are six valid object frames per dataset and round, except five for MOSEv2 in round two and five for VOST in round three; every MSE cell uses six frames. Each dataset mean retains two source sequences.

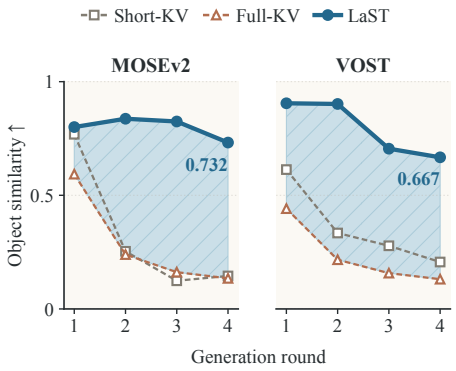

Figure 10: **Four-round continuation (5B).** Shading marks the LaST–Full-KV gap.

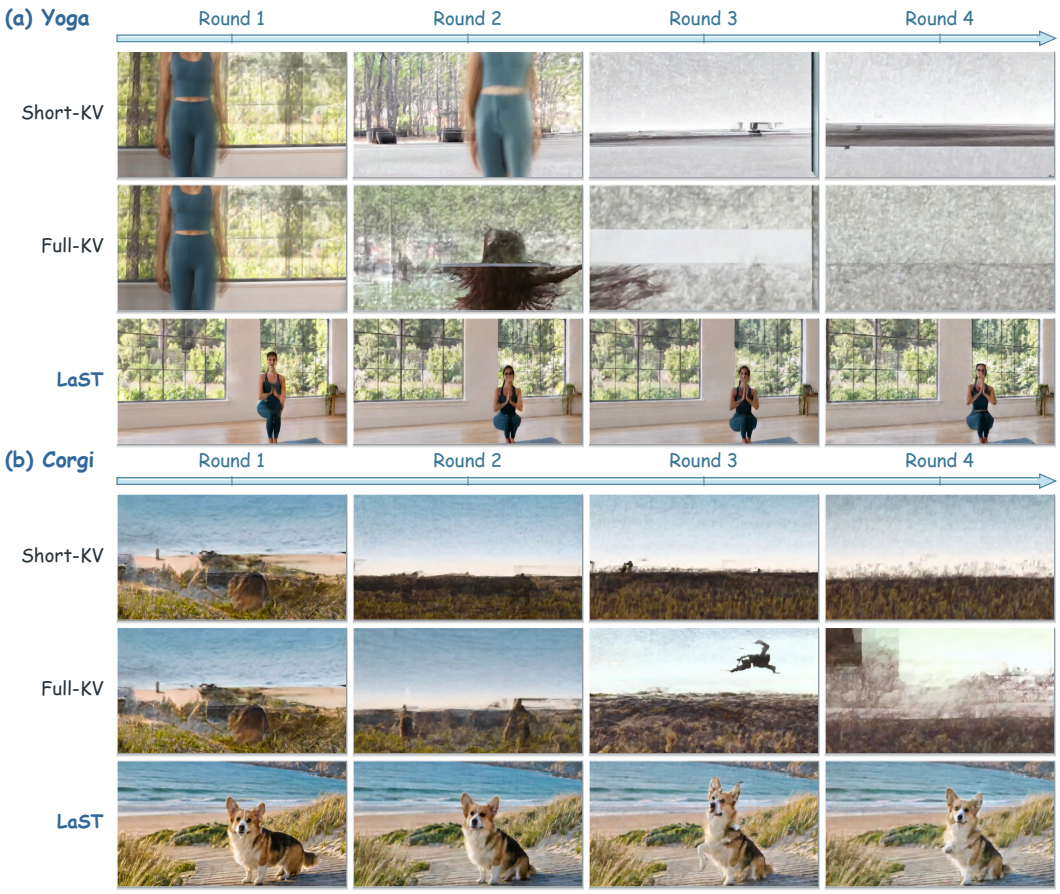

Figure 11: **Four-round camera-return comparison (5B).** Both camera sequences and all four rounds are shown. Rows follow the same method order; each cell is the complete final frame of its continuation.

A.8 Additional Qualitative Examples

Figures 12 and 13 pair camera-controlled LaST generations (upper rows) with independent full-scene references (lower rows). Camera motion is specified during generation. Columns give elapsed time within each separate clip, not synchronized observations.

As in Section 4.2, generation-time camera control takes the subject out of view and brings it back. Independent full-scene references provide context for the requested action. They can depart from the prompt: the terrace reference does not complete the leg crossing, and the robot reference drifts from its requested position.

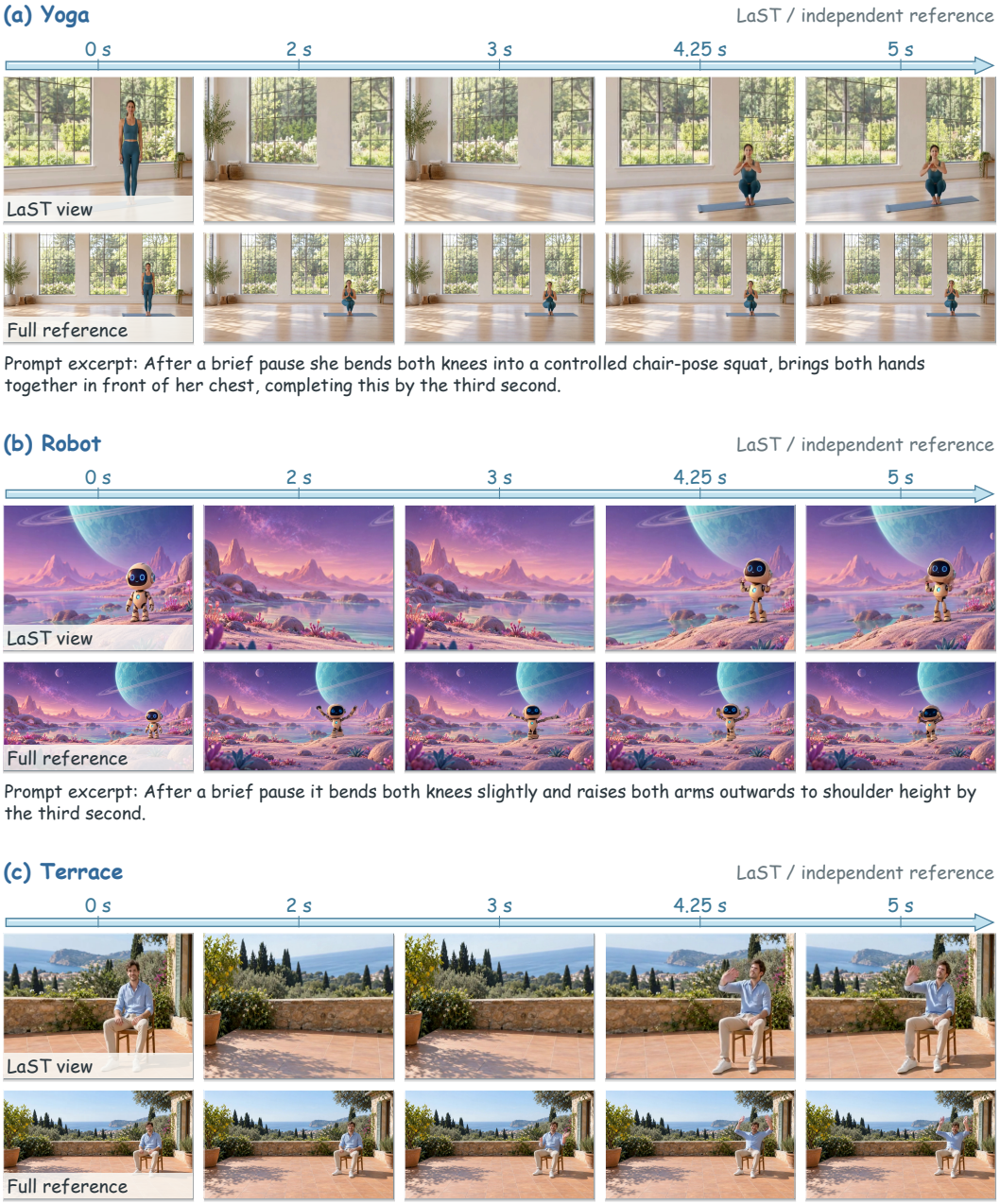

Figure 12: **Additional qualitative examples.** Yoga, robot, and terrace. Full-scene references use the same starting images and event prompts with new seeds in Wan2.2-Fun-5B (50 steps, 1280×736, 81 frames at 16 fps). They illustrate possible scene trajectories, not ground truth or inputs to LaST. They are excluded from benchmark scores.



Figure 13: **Additional animal examples.** Bear, corgi, and squirrel in the same format. The lower rows are independent samples under the same initial-image and prompt conditions, without additional final-frame input. These qualitative clips are separate from the 1280×704, 121-frame main-table evaluation.