SLOTMEMORY: OBJECT-CENTRIC KV MEMORY FOR STREAMING LONG-VIDEO GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Streaming video generation models typically rely on temporal-centric memory, organizing historical context as latent queues, compressed context tokens, or unclustered token sets. This often causes identity drift and semantic inconsistency when entities exit the frame or during interactive prompt transitions. To address these limitations, we propose SlotMemory, an object-centric Key-Value (KV) memory mechanism for streaming video diffusion. Our approach shifts memory abstraction from "when" an event occurred to "what" is represented by decomposing the transformer's KV manifold into discrete, reusable semantic slots. Because the slots serve only as routing addresses and the stored content remains the original transformer KV tokens, the mechanism adds entity-level persistence and prompt-aware retrieval across long horizons. Evaluated on 60-second interactive narratives using the Wan2.1-T2V-1.3B backbone, SlotMemory achieves the highest Dynamic Degree among evaluated streaming methods and the second-highest aggregate Quality Score, preserving prompt adherence rather than trading it away for consistency. At the 30-second horizon, SlotMemory leads in Total, Quality, and Dynamic Degree, improving Dynamic Degree by 31.2% over MemFlow, the strongest memory-based baseline. Our results support treating structured semantic representation, rather than raw temporal capacity, as a critical primitive for persistent long-form video synthesis.

1 INTRODUCTION

The frontier of generative video modeling has rapidly evolved from synthesizing isolated, short-duration clips (Wan et al., 2025; HaCohen et al., 2024) toward the more formidable objective of producing long-form, persistent narratives. Current streaming frameworks address this challenge by generating video in chunks and conditioning each new chunk on the past, either by rolling a window of recent KV states (Yang et al., 2026a; Huang et al., 2026; Yesiltepe et al., 2025; Chen et al., 2025) or by injecting a persistent appearance anchor into the generator (Henschel et al., 2025). Both families do carry information beyond the recent window—an appearance anchor is a memory that outlives the local slice—but neither addresses that memory by entity: the retained context is not attached to a specific identity, and no mechanism decides when an identity should be kept, recalled, or retired. However, these architectures typically rely on a temporal-centric memory organization, where history is represented as latent queues (Kim et al., 2024), compressed context tokens (Gu et al., 2025), or unclustered token sets (Yin et al., 2025). This approach introduces a fundamental bottleneck: by treating memory as a linear sequence of temporal slices rather than a structured representation of the scene, the models neglect the semantic hierarchy of the visual field. Consequently, they frequently fail to maintain identity coherence when entities exit and re-enter the frame or when the synthesis requires long-range interaction.

This temporal-centric memory organization produces two recurring failure modes in long-horizon video synthesis. First, it undermines entity persistence; when a character or object exits the local temporal window and re-enters, the model often fails to recover its unique attributes, resulting in visual drift or identity switching. Second, existing architectures are ill-suited for interactive, multi-prompt generation. When a prompt transitions–for example, from *a glaciologist drilling ice* to *an avalanche*, the model must retain persistent entities while discarding transient motion context. Existing approaches decide what to keep either by recency, through fixed decay over the temporal window, or by similarity to the current prompt (Ji et al., 2025; Yang et al., 2026b; Yu et al., 2025), and

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

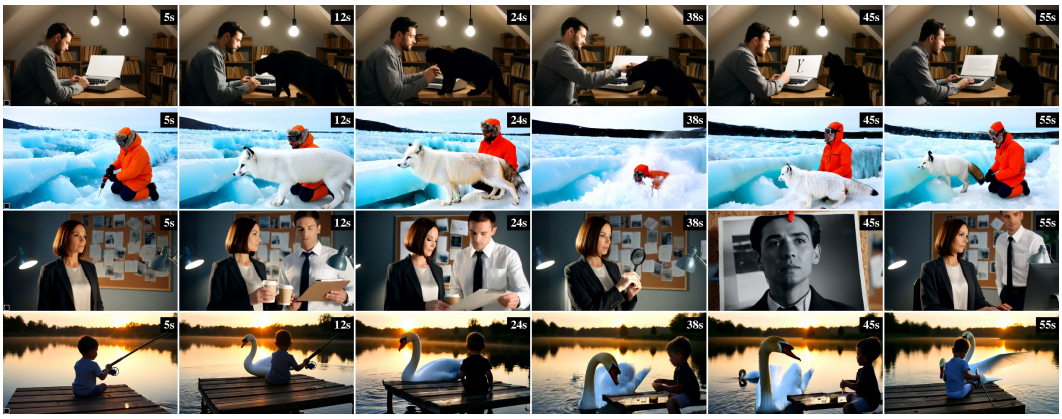


Figure 1: **Teaser examples of persistent, prompt-aware streaming video generation with SlotMemory.** Across extended temporal horizons, SlotMemory maintains coherent entity identity and appearance while supporting newly prompted interactions and scene dynamics, highlighting the advantage of object-centric KV memory over purely temporal context organization.

lack the structural awareness to distinguish persistent entities from transient motion. Consequently, they suffer from cumulative loss of coherence, leading to progressive narrative degradation over extended 60-second rollouts.

To address these limitations, we propose **SlotMemory**, an object-centric KV memory mechanism designed to enforce long-term semantic consistency in streaming video diffusion. Our core insight is a shift in memory abstraction from temporal occurrence ("when") to semantic identity ("what"). We achieve this by decomposing the transformer's KV manifold into discrete, reusable semantic slots via slot-based representation learning (Locatello et al., 2020; Manasyan et al., 2025). Figure 1 presents long-form examples where subject identity, key props, and action flow remain coherent across scene changes and temporary occlusions. Unlike traditional object-centric models, which are trained for unsupervised discovery on simple scenes (Burgess et al., 2019; Greff et al., 2019; Kipf et al., 2021; Elsayed et al., 2022; Zadaianchuk et al., 2023; Manasyan et al., 2025), SlotMemory bridges structured scene decomposition with high-resolution diffusion. Specifically, it utilizes these slots as routing addresses to index and store original, transformer-compatible KV tokens. This hybrid architecture preserves the high-fidelity detail essential for diffusion while providing a structured, entity-level index for efficient retrieval across extended horizons.

Architecturally, SlotMemory centers on an iterative read–write–update cycle that maintains a persistent semantic memory bank within fixed computational constraints. During the writing phase, we leverage a temporally-initialized Slot Attention mechanism (Manasyan et al., 2025) to partition high-dimensional transformer manifolds into discrete, reusable semantic regions. To make the bank usable across prompt switches, we recache the text-conditioning cache at every switch while leaving the entity bank and its slot states intact, so a new prompt reconditions the rollout without erasing the identities it inherits; core entities then survive the transition while obsolete temporal context is evicted. By embedding this module directly into the attention stack of the diffusion transformer, our architecture enables the model to attend to structured entities rather than unstructured frames, fundamentally stabilizing the synthesis of complex, multi-stage narratives.

We evaluate SlotMemory using the Wan2.1-T2V-1.3B backbone across a comprehensive suite of benchmarks, including a 60-second interactive multi-prompt scenario. On the interactive benchmark (Yang et al., 2026a; Ji et al., 2025; Yesiltepe et al., 2025; Liu et al., 2026; Yang et al., 2026b), the gains from this shift concentrate in the dimensions that temporal memory degrades first: Dynamic Degree and late-rollout prompt adherence. SlotMemory attains the best Dynamic Degree among the evaluated methods and the best text alignment in the final segment (62.97 at 50–60 s), while its aggregate Quality Score stays within a point of the strongest baseline (81.61 against 82.32). At the 30-second horizon it also leads in Total, Quality, and Dynamic scores, improving Dynamic Degree by 31.2% over MemFlow, the strongest memory-based baseline. Across the evaluated horizons, these results support treating structured semantic representation–at least as important as raw temporal capacity–as a critical primitive for persistent and high-fidelity long-form video synthesis.

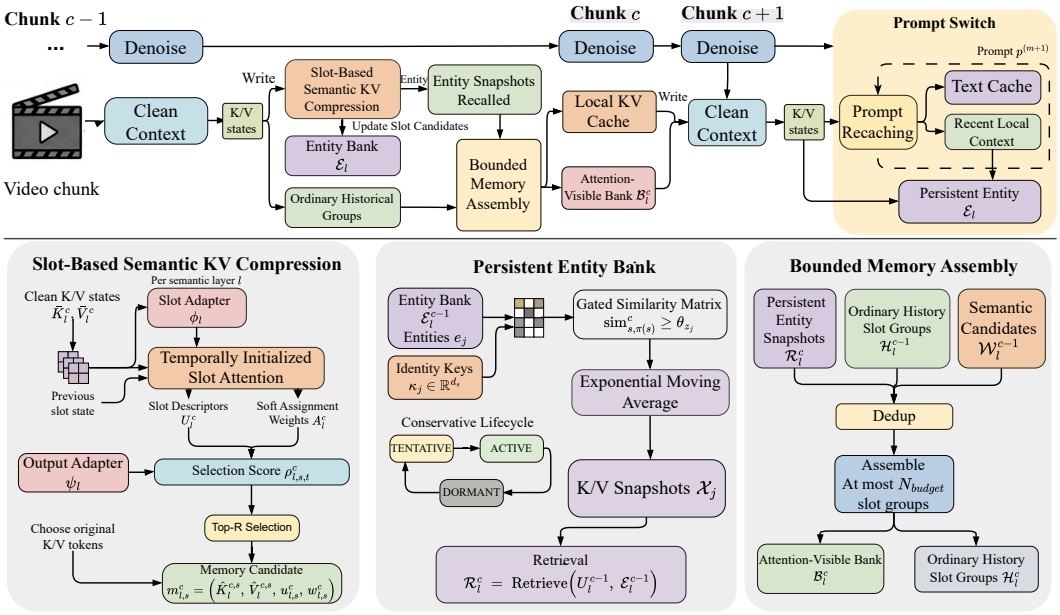


Figure 2: **Overview of SlotMemory for streaming long-video generation.** Top: each chunk reads the local KV cache and long-term bank, then writes clean states for subsequent chunks. Bottom: semantic slots address representative original K/V tokens, a persistent entity bank links identities across chunks and prompt transitions, and bounded assembly selects memory under a fixed budget.

2 Related Work

Long and Streaming Video Generation. Autoregressive (AR) streaming frameworks generate chunks sequentially under fixed computational budgets while managing key–value (KV) caches (Kim et al., 2024; Henschel et al., 2025; Teng et al., 2025). Such frameworks supplement local windows with explicit historical memory banks and prompt-aware retrieval (Yang et al., 2026a; Yu et al., 2025), but these banks store raw frames, chunk segments, or unclustered tokens, so identity is not reusable once an entity leaves the temporal window or the prompt changes. Retrieval-based designs retain raw fidelity at frame/chunk granularity (Hu et al., 2026; Zhao et al., 2026a), while consolidation-based designs compress chunk/segment history into bounded tokens (Lu et al., 2026; Kim et al., 2026), entangling entity identity with transient dynamics. Entity-centric designs address reusable entities but fall outside the attention stack: EM-Vid keeps a training-free, entity-indexed bank of sparse latent patches (Vandersanden et al., 2026), and Memento grounds subject memory in reconstruction (Wei et al., 2026). None of the surveyed designs combines entity-level addressing with raw KV fidelity; SlotMemory does so by routing original, uncompressed KV tokens through semantic slots in a structured, fixed-budget memory bank inside the diffusion transformer. A fuller account of this line is given in Section A.1.

Object-Centric Representation Learning. Object-centric representation learning decomposes scenes into discrete slots rather than monolithic features (Locatello et al., 2020; Kipf et al., 2021; Wu et al., 2022). This line establishes slots as robust persistent primitives, yet treats them as generative latents that decode into pixels or feed downstream predictors, not as transformer-compatible long-term KV memory. SlotMemory bridges the two lines by using slots purely as routing addresses over original KV tokens, imposing object-centric structure on the streaming attention stack without a lossy slot bottleneck. Further object-centric variants are covered in Section A.2.

3 Method

We formulate long-form, multi-prompt video synthesis as bounded-memory streaming generation and propose SlotMemory, a layer-wise hybrid memory system for causal video diffusion transformers. Rather than storing raw frames or unclustered tokens, it retains recent raw KV states for local detail, compresses clean intermediate-layer history into semantically addressable slots, and links those slots to persistent entities that survive temporary absence and prompt changes, so that slots

Table 1: **Quantitative comparison of interactive long video generation.** All models are evaluated using their official checkpoints. The best result in each column is shown in **bold**, and the second best is underlined.

Method	Quality Score ↑	Dynamic Degree ↑	Text Align ↑					
			0–10 s	10–20 s	20–30 s	30–40 s	40–50 s	50–60 s
Infinity-RoPE (Yesiltepe et al., 2025)	79.99	36.33	69.58	66.20	65.38	64.51	63.81	59.98
LongLive (Yang et al., 2026a)	79.09	28.30	71.88	**67.97**	**67.41**	**66.36**	**65.91**	62.29
MemFlow (Ji et al., 2025)	78.65	26.05	71.95	66.74	65.79	64.95	64.03	60.29
IAMFlow (Liu et al., 2026)	78.82	24.30	**72.41**	66.91	65.69	64.62	64.26	59.87
Anchor Forcing (Yang et al., 2026b)	**82.32**	72.87	71.83	67.83	65.82	65.28	64.70	62.41
SlotMemory	81.61	**81.38**	70.83	67.02	65.38	66.16	65.09	**62.97**

determine *which* past states to recall while the recalled content remains the original K/V consumed by attention. Figure 2 shows the resulting read–write–update cycle.

3.1 Bounded-Memory Streaming: Formulation and Overview

Given M text prompts $\mathcal{P} = \{p^{(1)}, \ldots, p^{(M)}\}$ with switch points $\mathcal{S} = \{s_1, \ldots, s_{M-1}\}$, a causal streaming diffusion transformer synthesizes C consecutive chunks of T_c latent frames, and at switch point s_m the active prompt changes from $p^{(m)}$ to $p^{(m+1)}$ while the accumulated history must remain usable. At layer l and chunk c, the denoising query Q_l^c attends to a short-term local KV cache $\mathcal{C}_l^c$ augmented by a fixed-capacity long-term KV bank $\mathcal{B}_l^c$:

$$\begin{aligned} K_l^{\mathrm{ctx},c} &= [K_l^{\mathrm{sink}}; K_l^{\mathrm{bank},c}; K_l^{\mathrm{local},c}], \\ V_l^{\mathrm{ctx},c} &= [V_l^{\mathrm{sink}}; V_l^{\mathrm{bank},c}; V_l^{\mathrm{local},c}], \\ Y_l^c &= \mathrm{Attn}(Q_l^c, K_l^{\mathrm{ctx},c}, V_l^{\mathrm{ctx},c}). \end{aligned} \tag{1}$$

The sink and local segments come from $\mathcal{C}_l^c$ and the bank segments from $\mathcal{B}_l^c$, while cross-attention conditions denoising on a text-conditioning cache refreshed only at switch points.

Two states govern the bank without entering attention directly: the slot state U_l^c, which records the slot descriptors of chunk c and initializes the next decomposition, and the persistent entity bank $\mathcal{E}_l^c$, which tracks visual identities and their K/V snapshots across chunks. The write path is causally delayed: chunk c reads only states written by earlier chunks, so the states it produces become visible from chunk $c+1$ onward. After generation, clean-context re-encoding yields $(\bar{K}_l^c, \bar{V}_l^c)$, which update the local cache and, at the next write point, are compressed into semantic candidates and committed to the entity bank. The layer-wise cache layout and state transitions are detailed in Section B.2; bounded memory assembly and prompt adaptation are specified in Section B.5. The resulting latency and memory overhead are measured in Section D.2.

3.2 Slot-Based Semantic KV Compression

The first component converts completed-chunk history into entries addressable by meaning rather than position. A naive compression stores low-dimensional token summaries, but diffusion attention directly consumes K/V features, and projected summaries cannot be recovered without loss. SlotMemory therefore separates memory *address* and *content*: slots provide compact semantic addresses, while stored content remains original transformer K/V.

For each chunk, clean key tokens are flattened over heads and projected into a slot-aligned latent space by adapter ϕ_l, so that each slot assigns complete tokens rather than individual heads:

$$Z_l^c = \phi_l(\bar{K}_l^c), \qquad (U_l^c, A_l^c) = \mathrm{SlotAttn}(Z_l^c; \bar{U}_l^{c-1}). \tag{2}$$

Here U_l^c denotes slot descriptors and A_l^c their soft token assignments. Slot Attention initializes from previous slot state U_l^{c-1} when available, otherwise a learned initializer U_l^0, enabling consecutive chunks to share a slot vocabulary.

A slot defines a semantic memory unit but is read in the original K/V space, so each slot stores R assigned tokens verbatim instead of descriptors. Selected states are gathered as

$$(\hat{K}_l^{c,s}, \hat{V}_l^{c,s}) = \big(\bar{K}_l^c[I_{l,s}^c], \bar{V}_l^c[I_{l,s}^c]\big), \tag{3}$$

Table 2: **Quantitative comparisons under single-prompt 5-second setting.** Among autoregressive models, the best result in each metric column is shown in **bold**, and the second-best is underlined.

Model	#Params	Resolution	Dynamic ↑	Evaluation scores ↑		
				Total	Quality	Semantic
Diffusion models						
LTX-Video (HaCohen et al., 2024)	1.9B	768×512	75.00	82.33	85.20	70.83
Wan2.1 (Wan et al., 2025)	1.3B	832×480	73.61	84.27	85.24	80.43
Autoregressive models						
SkyReels-V2 (Chen et al., 2025)	1.3B	960×540	48.61	81.72	82.93	76.90
CausVid (Yin et al., 2025)	1.3B	832×480	63.89	82.96	83.94	79.02
Pyramid Flow (Jin et al., 2025)	2B	640×384	68.06	81.56	83.89	72.24
Self-Forcing (Huang et al., 2026)	1.3B	832×480	63.89	83.83	84.60	80.79
Infinity-RoPE (Yesiltepe et al., 2025)	1.3B	832×480	58.33	83.55	84.34	80.43
DeepForcing (Yi et al., 2025)	1.3B	832×480	63.89	83.85	84.61	80.84
MemFlow (Ji et al., 2025)	1.3B	832×480	50.00	81.76	83.74	73.86
LongLive (Yang et al., 2026a)	1.3B	832×480	37.50	82.81	83.26	**81.01**
Anchor Forcing (Yang et al., 2026b)	1.3B	832×480	**73.61**	**83.99**	**84.84**	80.57
SlotMemory	1.3B	832×480	70.27	83.68	84.38	80.85

where $I^c_{l,s}$ contains the R selected token indices; memory entries thus remain attention-compatible K/V groups rather than compressed embeddings. The token-selection score, write-quality estimate, and candidate construction are given in Section B.3.

3.3 Persistent Entity Bank

Slot initialization ties a chunk's semantic decomposition to its predecessor, but slots are not identities: an entity may map to different slots after gaps, and slots may be reassigned after prompt changes. Entity memory complements slot memory by maintaining, independently per semantic layer, a reusable identity bank $\mathcal{E}^c_l = \{e_j\}_{j=1}^{J}$ across the video. Each entity e_j stores an identity key κ_j for slot comparison, a quality estimate, lifecycle state, and bounded transformer-compatible K/V snapshots. Association determines slot-to-identity correspondence for the latest completed chunk. Given slot descriptors $U^{c-1}_{l,s}$ from chunk $c-1$ and keys in $\mathcal{E}^{c-1}_l$, admissible pairs are scored by cosine similarity and assigned under a gate:

$$\mathrm{sim}^c_{s,j} = \cos\big(U^{c-1}_{l,s}, \kappa^{c-1}_j\big). \tag{4}$$

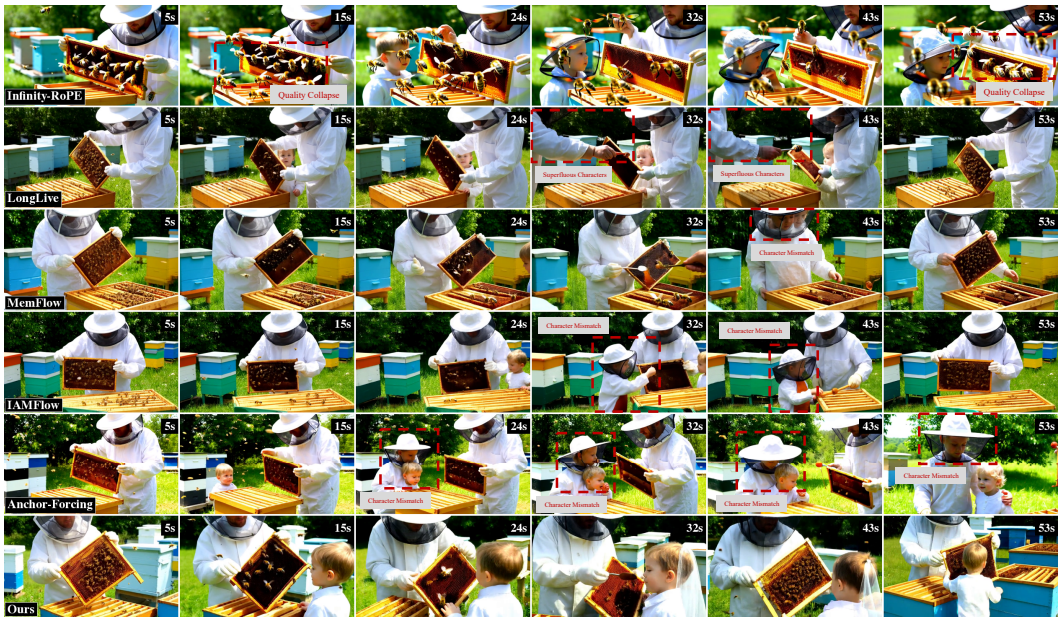
Hungarian matching solves the gated assignment: matched slots reinforce existing identities, while unmatched slots with sufficient write quality initialize tentative entities. The matching gate and lifecycle transitions are specified in Section B.4.

An identity requires visual content to condition generation. Each entity stores bounded K/V history from representative slot-compressed tokens, so its retrieved snapshots remain original attention-compatible K/V rather than embeddings. Retrieval occurs before current slot commitment, so chunk c accesses history through chunk $c-1$:

$$\mathcal{R}^c_l = \mathrm{Retrieve}(U^{c-1}_l, \mathcal{E}^{c-1}_l), \qquad \mathcal{B}^c_l = \mathrm{Assemble}(\mathcal{R}^c_l, \mathcal{H}^{c-1}_l, \mathcal{W}^{c-1}_l). \tag{5}$$

Returned groups are inserted into attention by the assembly step in Section B.5; current slots are then committed for later chunks. Retrieval, snapshot retention, and persistent access to temporarily absent entities are detailed in Section B.4: continuity for an entity that re-enters after a long absence comes from restoring its stored snapshots into the bank rather than from re-identifying the returning slot. Retrieved snapshots retain the rotary phases under which they were written; Section D.3 examines this design choice.

Training. Training must expose memory to self-generated states, since the model writes inference-time entries during rollout. Each iteration continues this rollout: the generator synthesizes chunks sequentially, writes each completed chunk's clean state into memory, and conditions future chunks on stored states. The generator is adapted by distribution matching distillation (DMD), where a trainable fake-score network and frozen teacher score the same noisy sample, and their difference guides generator updates.



0-10s:**A beekeeper in a full white protective suit** is calmly inspecting a honeycomb frame taken from a bustling beehive in a sunny apiary. Bees are buzzing all around. Other hives are visible in the background.
10-20s:**The beekeeper inspecting the honeycomb in the sunny apiary is assisted by a young child**, also in a smaller, child-sized beekeeper suit, who is watching with wide-eyed fascination. The bees are active.
20-30s:**The beekeeper and the child in beekeeper** suits are in the sunny apiary. The beekeeper points out the queen bee on the honeycomb frame to the child. The child leans in for a closer look. The other hives are in the background.
30-40s:**The beekeeper in the sunny apiary** gently scrapes a small amount of fresh honey from the frame with a tool and offers it to the child. **The child** tastes the honey, their eyes lighting up behind their veil.
40-50s:**The beekeeper and the child** in the sunny apiary carefully place the honeycomb frame back into the bustling beehive. They work together, their movements slow and deliberate to avoid disturbing the bees.
50-60s:**The beekeeper and the child** walk away from the hives in the sunny apiary. The beekeeper puts an arm around the child's shoulder, a shared experience of wonder at the world of bees.

Figure 3: **Qualitative comparison of interactive long-form text-to-video generation.**

The objective combines this term with memory regularization:

$$\mathcal{L}_{\text{gen}} = \mathcal{L}_{\text{DMD}} + \lambda_{\text{slot}} \mathcal{L}_{\text{slot}}. \tag{6}$$

These terms address slot drift, inaccurate content representation, and token selection unrelated to actual chunk usage. Full implementation details are collected in Section C, which specifies the streaming rollout (Section C.1), the two-stage schedule and frozen modules (Section C.3), the loss definitions (Section C.2), and the optimization recipe (Section C.4).

4 Experiments

4.1 Experiment Setups

Implementation Details. We instantiate SlotMemory using the pretrained Wan2.1-T2V-1.3B backbone (Wan et al., 2025), which generates 5-second clips at 16 FPS and a resolution of 832 $\times$ 480. Following the optimization strategy of LongLive (Yang et al., 2026a) and MemFlow (Ji et al., 2025), the training process is divided into two phases. First, the base model is adapted into a few-step causal-attention generator via a Self-Forcing Distribution Matching Distillation pipeline (Huang et al., 2026) on the VidProM dataset (Wang & Yang, 2024), utilizing short-window attention and first-frame sink tokens to stabilize autoregressive rollouts. Second, we freeze the backbone weights and initialize the slot adapter and memory modules for streaming long-tuning. This phase utilizes 60-second multi-prompt sequences where each iteration continues the model's own rollout. Prompt switches are sampled uniformly between 5 and 55 seconds to train the module to perform both entity preservation and context suppression. The memory bank and sparse activation modules are optimized for 3,000 steps using AdamW (learning rates of 1.0×10^{-5} for the actor and 2.0×10^{-6} for the critic). Training is completed in approximately 12 hours on 8 NVIDIA H200 GPUs.

Benchmarks and Datasets. To evaluate temporal robustness across varying horizons, we establish benchmarks for three distinct scenarios: short-video generation (5s), long single-prompt generation

Table 3: **Quantitative comparisons under a 30-second single-prompt setting.** We evaluate extended-horizon generation capabilities against recent autoregressive streaming models (↑ higher is better). The best result in each metric column is shown in **bold**, and the second-best is underlined.

Model	Total Score ↑	Quality Score ↑	Semantic Score ↑	Dynamic Score ↑	Imaging Quality ↑	Temporal Style ↑
Self-Forcing (Huang et al., 2026)	82.21	83.38	77.54	51.85	68.17	23.31
Infinity-RoPE (Yesiltepe et al., 2025)	83.16	84.17	79.13	57.04	67.94	23.79
DeepForcing (Yi et al., 2025)	82.26	83.20	78.50	52.86	67.65	23.51
MemFlow (Ji et al., 2025)	82.95	83.86	79.31	60.46	67.68	23.89
LongLive (Yang et al., 2026a)	82.77	83.31	**80.64**	40.19	68.96	23.97
Anchor Forcing (Yang et al., 2026b)	83.25	84.40	78.67	79.17	**69.37**	**24.17**
SlotMemory	**83.79**	**84.86**	79.49	**79.30**	69.11	24.09

(30s,120s,240s), and interactive multi-prompt narratives (60s). For 5s and long single-prompt evaluations, we adopt standard protocols from Self-Forcing (Huang et al., 2026) and LongLive (Yang et al., 2026a). The 60s interactive benchmark follows MemFlow (Ji et al., 2025), comprising 100 narrative scripts. Each script contains six consecutive 10s prompts, requiring the model to maintain semantic consistency across five discrete transitions.

Evaluation Metrics. We perform a multidimensional quantitative assessment using the VBench (Huang et al., 2024) and VBench-Long (Huang et al., 2025) frameworks. Short-video performance is measured via Total, Quality, Semantic, and Dynamic scores. For the 30-second horizon, we extend this to include Imaging Quality and Temporal Style to capture long-range perceptual stability. In the 60-second interactive setting, we report the aggregate Quality score to reflect overall visual fidelity. To specifically analyze the impact of prompt transitions, we compute segment-wise text align use Qwen3-VL-Embedding-8B (Li et al., 2026a) for each 10-second interval, providing a granular view of how well semantic identities are preserved as the narrative evolves.

4.2 Main Experiments

Interactive Multi-Prompt Generation. Prompt switches are the sharpest test of what a memory keeps, because the instruction changes while the entities it describes do not. As shown in Table 1, SlotMemory attains the best Dynamic Degree by a wide margin (81.38 against 72.87 for the next best method) while ranking second on aggregate Quality (81.61 against 82.32 for Anchor Forcing); LongLive and MemFlow stay below 80 on Quality and below 30 on Dynamic. The segmented text alignment shows that this margin is not an artifact of one segment: SlotMemory is not the best method early on, but it is the most stable one across the sequence, losing 7.86 points from the first segment to the last, against 9.4–12.5 points for the baselines, and it ends on the highest score of any method. A memory that carries entities across switches should decay more slowly than one that rebuilds the scene from the recent window at each switch, and the separation appears exactly where such a delay accumulates: at the end of the sequence. The qualitative grid in Figure 3 shows the same pattern on a six-prompt apiary script, where the baselines lose the content or the cast while SlotMemory keeps the beekeeper, the child, and the frame consistent throughout. Additional scripts, covering a cast change and two stress cases with annotated failure modes, are shown in Section F.

Long-Video Generation. At 30 seconds the rollout is long enough for accumulated drift to dominate, and this is the setting where slot memory separates from the temporal memory it replaces. Table 3 places SlotMemory first on Total (83.79), Quality (84.86), and Dynamic (79.30) and second on the remaining three dimensions, with no other method leading on more than two. Dynamic Degree is the diagnostic column: MemFlow and LongLive fall to 60.46 and 40.19, while SlotMemory holds 79.30. A temporal memory preserves consistency by re-using recent content, which ties how much a scene can move to how much of it is replayed; a memory that stores entity appearance instead of scene content leaves the scene free to move. The nearest competitor, Anchor Forcing, matches SlotMemory to within 0.6 points on Total and Quality, so the methods are separated not by perceptual quality but by what each memory keeps: ours stays close to LongLive on Semantic, while Anchor Forcing drops to 78.67. The same separation persists at 120 s and 240 s, where appearance metrics saturate while the margin in Dynamic Degree widens; see Section D.1 and Table D.4.

Table 4: **Core mechanism ablation.** Tail alignment averages 30–60s; text drift is the 0–10s to 50–60s drop. FG-ARI and mBO use ReferDINO (Liang et al., 2025) pseudo-masks from prompt-extracted object text, averaged over the top three layers. Bold and underline mark the best and second-best results.

Method	Text avg. ↑	Tail align. ↑	Text Drift↓	FG-ARI ↑	mBO ↑	VBench quality ↑
Baseline w/o Slot Compression	65.63	63.09	11.66	–	–	78.65
+ Slot (no entity bank)	66.15	63.71	11.17	8.01	9.42	78.89
+ Entity Bank (w/o $\mathcal{L}_{\text{dist}}$)	**66.63**	**65.12**	**7.84**	<u>10.42</u>	**15.64**	<u>81.50</u>
Top-k KV Memory	65.66	63.58	10.23	–	–	80.59
Ours (full)	<u>66.24</u>	<u>64.74</u>	<u>7.86</u>	**11.77**	<u>15.53</u>	**81.61**



Figure 4: **Slot-wise representative tokens.** Colored patches mark retained tokens for different slots across two sequences; slots track distinct objects and scene regions over time.

Short-Video Generation. A 5-second clip is too short for the bank to be exercised, so the question here is not whether memory helps but whether attaching it costs anything. It does not (Table 2): SlotMemory lands within 0.6 and 0.9 points of the Wan2.1-T2V-1.3B backbone on Total and Quality, and among autoregressive models it is second on both Dynamic and Semantic. The Dynamic column again carries the mechanism: MemFlow and LongLive lose motion at this horizon, whereas SlotMemory is higher than every autoregressive baseline except Anchor Forcing. This is what the design predicts, since a bank that only becomes useful once the early frames leave the recent window should be close to inert when no such history exists yet.

Summary of Scaling Performance. The three horizons give one consistent picture: at 5 seconds SlotMemory behaves like the backbone it extends, at 30 seconds it wins the dimensions that depend on memory, and at 60 seconds it is the most stable method across five prompt switches. The margin over the baselines therefore grows with horizon and concentrates in motion, appearance, and late-segment prompt adherence rather than in short-clip fidelity, which is what the persistent memory protects and what the baselines lose first. Extended quantitative results, including the long-horizon and efficiency measurements, are collected in Section D.

User study. We conducted a human evaluation of long-form video quality across identity preservation, action completion, scene consistency, temporal stability, and object interaction. Twenty users compared 300 video groups and preferred our method in 81 cases (27%), versus Anchor-Forcing in 62 (20.7%), IAMFlow in 37 (12.3%), Infinity-RoPE in 43 (14.3%), LongLive in 52 (17.3%), and MemFlow in 25 (8.3%); SlotMemory was the most frequently preferred method in this comparison. The protocol, evaluation criteria, and quality-control rules are given in Section E. Latency, peak memory, and throughput under the interactive protocol are reported in Section D.2 (Table D.5).

Table 5: **Loss ablation under the full SlotMemory mechanism (slot compression + persistent entity bank).** Tail text alignment is averaged over 30–60s. Best and second-best results are highlighted in **bold** and underlined, respectively.

Slot regularization	Subject consistency ↑	Background consistency ↑	Dynamic degree ↑	VBench quality ↑	Tail text align. ↑
$\mathcal{L}_{\text{con}} + \mathcal{L}_{\text{rec}}$ (w/o $\mathcal{L}_{\text{dist}}$)	93.07	92.76	**92.96**	<u>81.50</u>	**65.12**
$\mathcal{L}_{\text{con}} + \mathcal{L}_{\text{dist}}$ (w/o $\mathcal{L}_{\text{rec}}$)	**95.36**	**93.90**	59.08	79.99	63.24
Complete ($\mathcal{L}_{\text{con}} + \mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{dist}}$)	<u>95.00</u>	<u>93.86</u>	<u>81.38</u>	**81.61**	<u>64.74</u>

4.3 Ablation Studies

We ablate in the 60-second interactive setting of Table 1, varying one axis at a time—what the memory is made of, what supervises the slots, and how much memory is kept—while holding the backbone, local window, chunk protocol, prompt-switch protocol, and retained-token budget fixed.

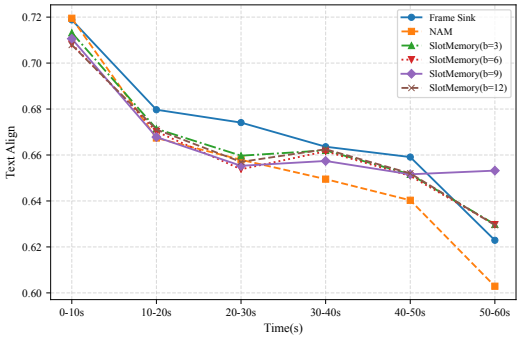


Figure 5: **Segment-wise text alignment across memory designs and bank sizes.**

Impact of Slot Compression and Entity Persistence. Table 4 separates semantic addressing from entity persistence. Slot compression groups retained tokens by entity but barely changes aggregate scores without cross-chunk identity matching. Adding the entity bank reduces text drift from 11.17 to 7.84 and improves tail alignment, mBO, and FG-ARI, while average alignment changes little. We analyze the identity key's positional representation in Section D.3.

Structure versus Budget. The Top-k row separates the retained tokens from the addresses they are stored under. Untyped tokens of the same budget recover most of the compression loss in overall quality, so the budget is not the bottleneck; what they cannot express is which token belongs to which entity, and the variant consequently behaves like slot-only memory at the end of the sequence. Slots are what make the tokens addressable: they are the unit the entity bank associates, the unit a prompt switch retrieves by, and the unit bounded assembly selects from. Figure 4 shows that grouping directly, and the FG-ARI and mBO columns of Table 4 quantify its coherence over time.

What the Two Slot Losses Supervise. The two regularizers in Table 5 are not interchangeable. Removing reconstruction leaves the highest subject and background consistency but takes the motion with it—slots that are no longer grounded in the tokens the backbone renders collapse Dynamic Degree from 93 to 59—whereas removing distillation keeps motion and the best tail alignment but gives up consistency and slot–object alignment. Neither term dominates, and only the complete objective reaches the best overall VBench quality without collapsing on either axis.

Capacity and Memory Design. Bank capacities differ mainly in the final segment: nine blocks improve alignment and motion, while the six-block default yields the highest overall quality. In Figure 5, frame-sink caching and NAM match slot memory early but lose roughly twice as much alignment by the end, when entity recall matters most.

5 Conclusion

We introduce SlotMemory, an object-centric KV memory that uses semantic slots to retrieve original transformer KV tokens and maintain entity continuity across streaming chunks and prompt switches. On 60-second interactive narratives, it achieves the highest Dynamic Degree and final-segment text alignment among evaluated methods while ranking second in aggregate Quality. At 30 seconds, it leads in Total, Quality, and Dynamic scores, with 31.2% higher Dynamic Degree than MemFlow. Ablations further show that persistent entity tracking, beyond slot compression alone, reduces late-stage text drift and improves object-level coherence. These results indicate that addressable entity memory is what preserves motion and identity at long horizons.

AI USE STATEMENT

In this work, we used generative AI tools to improve the readability, grammar, clarity, and overall presentation of the manuscript, and to identify potential typographical, grammatical, and formatting errors. Additionally, generative AI tools were used for language polishing and proofreading of portions of the manuscript. All AI-assisted text was carefully reviewed and revised by the authors to ensure that it accurately reflects the underlying research and does not introduce unsupported claims, factual errors, or changes to the intended technical meaning.

ETHICS STATEMENT

This work does not introduce or release a new dataset containing personal or sensitive information. The proposed method is a general-purpose memory mechanism for long-form video generation and is evaluated on synthetic video-generation tasks. We are not aware of any additional ethical, privacy, security, or legal concerns specific to the methodology or experiments presented in this work.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the proposed architecture, memory organization, slot-based KV compression, persistent entity bank, retrieval and update procedures, training objectives, and streaming state transitions in the main paper and appendix. In particular, the appendix specifies the notation, tensor shapes, layer placement, memory budgets, causal update order, entity association and retrieval procedures, training schedule, and optimization settings. The experimental section further documents the Wan2.1-T2V-1.3B backbone, training data, benchmark settings, evaluation protocols, and hardware configuration. To facilitate reproducibility and future research, we plan to release the model and source code of our method publicly after publication, together with the relevant configuration files and implementation details. These materials, combined with the detailed specifications provided in the paper and appendix, are intended to support reproduction of the proposed method and experimental results.

REFERENCES

Omer Bar-Tal, Hila Chefer, Omer Tov, Charles Herrmann, Roni Paiss, Shiran Zada, Ariel Ephrat, Junhwa Hur, Guanghui Liu, Amit Raj, et al. Lumiere: A space-time diffusion model for video generation. In *SIGGRAPH Asia 2024 Conference Papers*, pp. 1–11, 2024.

Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets, 2023.

Christopher P Burgess, Loic Matthey, Nicholas Watters, Rishabh Kabra, Irina Higgins, Matt Botvinick, and Alexander Lerchner. Monet: Unsupervised scene decomposition and representation, 2019.

Shengqu Cai, Ceyuan Yang, Lvmin Zhang, Yuwei Guo, Junfei Xiao, Ziyan Yang, Yinghao Xu, Zhenheng Yang, Alan Yuille, Leonidas Guibas, et al. Mixture of contexts for long video generation, 2025.

Guibin Chen, Dixuan Lin, Jiangping Yang, Chunze Lin, Junchen Zhu, Mingyuan Fan, Hao Zhang, Sheng Chen, Zheng Chen, Chengcheng Ma, et al. Skyreels-v2: Infinite-length film generative model, 2025.

Honglin Chen, Wanhee Lee, Hong-Xing Yu, Rahul Mysore Venkatesh, Joshua B Tenenbaum, Daniel Bear, Jiajun Wu, and Daniel LK Yamins. Unsupervised 3d scene representation learning via movable object inference. *Transactions on Machine Learning Research*, 2024.

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 *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 17702–17711, 2025.

Aniket Didolkar, Andrii Zadaianchuk, Rabiul Awal, Maximilian Seitzer, Efstratios Gavves, and Aishwarya Agrawal. Ctrl-o: Language-controllable object-centric visual representation learning. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 29523–29533, 2025a. doi: 10.1109/CVPR52734.2025.02749.

Aniket Rajiv Didolkar, Andrii Zadaianchuk, Anirudh Goyal, Michael Curtis Mozer, Yoshua Bengio, Georg Martius, and Maximilian Seitzer. On the transfer of object-centric representation learning. In *The Thirteenth International Conference on Learning Representations*, 2025b. URL https://openreview.net/forum?id=bSq0XGS3kW.

Gamaleldin Elsayed, Aravindh Mahendran, Sjoerd Van Steenkiste, Klaus Greff, Michael C Mozer, and Thomas Kipf. Savi++: Towards end-to-end object-centric learning from real-world videos. *Advances in Neural Information Processing Systems*, 35:28940–28954, 2022.

Martin Engelcke, Adam R Kosiorek, Oiwi Parker Jones, and Ingmar Posner. GENESIS: Generative Scene Inference and Sampling with Object-Centric Latent Representations. In *International Conference on Learning Representations (ICLR)*, 2020.

Xueji Fang, Liyuan Ma, Zhiyang Chen, Mingyuan Zhou, and Guo-jun Qi. Inflvg: Reinforce inference-time consistent long video generation with grpo, 2025.

Klaus Greff, Raphaël Lopez Kaufman, Rishabh Kabra, Nick Watters, Christopher Burgess, Daniel Zoran, Loic Matthey, Matthew Botvinick, and Alexander Lerchner. Multi-object representation learning with iterative variational inference. In *International conference on machine learning*, pp. 2424–2433. PMLR, 2019.

Yuchao Gu, Weijia Mao, and Mike Zheng Shou. Long-context autoregressive video modeling with next-frame prediction, 2025. URL https://arxiv.org/abs/2503.19325.

Yuwei Guo, Ceyuan Yang, Anyi Rao, Zhengyang Liang, Yaohui Wang, Yu Qiao, Maneesh Agrawala, Dahua Lin, and Bo Dai. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning, 2023.

Yoav HaCohen, Nisan Chiprut, Benny Brazowski, Daniel Shalem, Dudu Moshe, Eitan Richardson, Eran Levin, Guy Shiran, Nir Zabari, Ori Gordon, et al. Ltx-video: Realtime video latent diffusion, 2024.

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. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 2568–2577, 2025.

Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models, 2022a.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Advances in neural information processing systems*, 35:8633–8646, 2022b.

Qixin Hu, Shuai Yang, Wei Huang, Song Han, and Yukang Chen. Longlive-rag: A general retrieval-augmented framework for long video generation, 2026. URL https://arxiv.org/abs/2606.02553.

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

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, 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*, 2024.

Ziqi Huang, Fan Zhang, Xiaojie Xu, Yinan He, Jiashuo Yu, Ziyue Dong, Qianli Ma, Nattapol Chanpaisit, Chenyang Si, Yuming Jiang, Yaohui Wang, Xinyuan Chen, Ying-Cong Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. VBench++: Comprehensive and versatile benchmark suite for video generative models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025. doi: 10.1109/TPAMI.2025.3633890.

Allan Jabri, Sjoerd van Steenkiste, Emiel Hoogeboom, Mehdi SM Sajjadi, and Thomas Kipf. Dorsal: Diffusion for object-centric representations of scenes. In *International Conference on Learning Representations*, volume 2024, pp. 43030–43052, 2024.

Sihui Ji, Xi Chen, Shuai Yang, Xin Tao, Pengfei Wan, and Hengshuang Zhao. Memflow: Flowing adaptive memory for consistent and efficient long video narratives, 2025.

Yang Jin, Zhicheng Sun, Ningyuan Li, Kun Xu, Hao Jiang, Nan Zhuang, Quzhe Huang, Yang Song, Yadong Mu, and Zhouchen Lin. Pyramidal flow matching for efficient video generative modeling. In *International Conference on Learning Representations*, volume 2025, pp. 23378–23402, 2025.

Jihwan Kim, Junoh Kang, Jinyoung Choi, and Bohyung Han. Fifo-diffusion: Generating infinite videos from text without training. *Advances in Neural Information Processing Systems*, 37: 89834–89868, 2024.

Youngrae Kim, Qixin Hu, C. C. Jay Kuo, and Peter A. Beerel. Memrope: Training-free infinite video generation via evolving memory tokens, 2026. URL https://arxiv.org/abs/2603.12513.

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, 2021.

Avinash Kori, Francesco Locatello, Fabio De Sousa Ribeiro, Francesca Toni, and Ben Glocker. Grounded object-centric learning. In *International Conference on Learning Representations*, volume 2024, pp. 42559–42603, 2024a.

Avinash Kori, Francesco Locatello, Ainkaran Santhirasekaram, Francesca Toni, Ben Glocker, and Fabio De Sousa Ribeiro. Identifiable object-centric representation learning via probabilistic slot attention. *Advances in Neural Information Processing Systems*, 37:93300–93335, 2024b.

Mingxin Li, Yanzhao Zhang, Dingkun Long, Keqin Chen, Sibo Song, Shuai Bai, Zhibo Yang, Pengjun Xie, An Yang, Dayiheng Liu, Jingren Zhou, and Junyang Lin. Qwen3-vl-embedding and qwen3-vl-reranker: A unified framework for state-of-the-art multimodal retrieval and ranking, 2026a. URL https://arxiv.org/abs/2601.04720.

Zhiyuan Li, Rongzhen Zhao, Wenyan Yang, Wenshuai Zhao, Pekka Marttinen, and Joni Pajarinen. Rethinking temporal consistency in video object-centric learning: From prediction to correspondence. In *Forty-third International Conference on Machine Learning*, 2026b. URL https://openreview.net/forum?id=aH0jZGucwj.

Tianming Liang, Kun-Yu Lin, Chaolei Tan, Jianguo Zhang, Wei-Shi Zheng, and Jian-Fang Hu. Referdino: Referring video object segmentation with visual grounding foundations. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 20009–20019, 2025. doi: 10.1109/ICCV51701.2025.01861.

Jinzhuo Liu, Jiangning Zhang, Wencan Jiang, Yabiao Wang, Dingkang Liang, Zhucun Xue, Ran Yi, and Yong Liu. Advancing narrative long video generation via training-free identity-aware memory, 2026. URL https://arxiv.org/abs/2605.18733.

Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. *Advances in neural information processing systems*, 33:11525–11538, 2020.

Yu Lu, Junjie Yang, Piotr Koniusz, YuXin Song, and Yi Yang. Fademem: Distance-aware memory consolidation for autoregressive video diffusion, 2026. URL https://arxiv.org/abs/2606.10671.

Riccardo Majellaro, Jonathan Collu, Aske Plaat, and Thomas M. Moerland. Explicitly disentangled representations in object-centric learning. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=r8UFp9olQ0.

Anna Manasyan, Maximilian Seitzer, Filip Radovic, Georg Martius, and Andrii Zadaianchuk. Temporally consistent object-centric learning by contrasting slots. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 5401–5411, 2025.

Bac Nguyen, Yuhta Takida, Naoki Murata, Chieh-Hsin Lai, Toshimitsu Uesaka, Stefano Ermon, and Yuki Mitsufuji. Improved object-centric diffusion learning with registers and contrastive alignment. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=WQHj907qSI.

Maximilian Seitzer, Max Horn, Andrii Zadaianchuk, Dominik Zietlow, Tianjun Xiao, Carl-Johann Simon-Gabriel, Tong He, Zheng Zhang, Bernhard Schölkopf, Thomas Brox, et al. Bridging the gap to real-world object-centric learning, 2022.

Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-a-video: Text-to-video generation without text-video data, 2022.

Shanwen Tan, Hao Li, Jingtao Zhang, Xiaosong Jia, Xue Yang, Shaofeng Zhang, and Yanyong Zhang. Swift: Prompt-adaptive memory for efficient interactive long video generation, 2026. URL https://arxiv.org/abs/2605.09442.

Hansi Teng, Hongyu Jia, Lei Sun, Lingzhi Li, Maolin Li, Mingqiu Tang, Shuai Han, Tianning Zhang, WQ Zhang, Weifeng Luo, et al. Magi-1: Autoregressive video generation at scale, 2025.

Jente Vandersanden, Matheus Gadelha, Chun-Hao P. Huang, Hyeonho Jeong, and Yulia Gryaditskaya. Em-vid: Training-free entity-centric memory for efficient and consistent multi-shot video generation, 2026. URL https://arxiv.org/abs/2605.23610.

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

Jiuniu Wang, Hangjie Yuan, Dayou Chen, Yingya Zhang, Xiang Wang, and Shiwei Zhang. Modelscope text-to-video technical report, 2023.

Wenhao Wang and Yi Yang. Vidprom: A million-scale real prompt-gallery dataset for text-to-video diffusion models. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. URL https://openreview.net/forum?id=pYNl76onJL.

Xuan Wei, Longbin Ji, Guan Wang, Xiangrui Liu, Zhenyu Zhang, Shuohuan Wang, Yu Sun, and Qingqi Hong. Memento: Reconstruct to remember for consistent long video generation, 2026. URL https://arxiv.org/abs/2606.14667.

Jay Zhangjie Wu, Yixiao Ge, Xintao Wang, Stan Weixian Lei, Yuchao Gu, Yufei Shi, Wynne Hsu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 7623–7633, 2023.

Mingqiang Wu, Weilun Feng, Zhefeng Zhang, Haotong Qin, Yuqi Li, Guoxin Fan, Xiaokun Liu, Zhulin An, Libo Huang, Yongjun Xu, and Chuanguang Yang. Echo-forcing: A scene memory framework for interactive long video generation, 2026. URL https://arxiv.org/abs/2605.16003.

Ziyi Wu, Nikita Dvornik, Klaus Greff, Thomas Kipf, and Animesh Garg. Slotformer: Unsupervised visual dynamics simulation with object-centric models, 2022.

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*, 2026a. URL https://openreview.net/forum?id=nCAODkpsPJ.

Yang Yang, Tianyi Zhang, Wei Huang, Jinwei Chen, Boxi Wu, Xiaofei He, Deng Cai, Bo Li, and Peng-Tao Jiang. Anchor forcing: Anchor memory and tri-region rope for interactive streaming video diffusion, 2026b. URL https://arxiv.org/abs/2603.13405.

Hidir Yesiltepe, Tuna Han Salih Meral, Adil Kaan Akan, Kaan Oktay, and Pinar Yanardag. Infinity-rope: Action-controllable infinite video generation emerges from autoregressive self-rollout, 2025.

Jung Yi, Wooseok Jang, Paul Hyunbin Cho, Jisu Nam, Heeji Yoon, and Seungryong Kim. Deep forcing: Training-free long video generation with deep sink and participative compression, 2025.

Tianwei Yin, Qiang Zhang, Richard Zhang, William T Freeman, Fredo Durand, Eli Shechtman, and Xun Huang. From slow bidirectional to fast autoregressive video diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22963–22974, 2025.

Jiwen Yu, Jianhong Bai, Yiran Qin, Quande Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Xihui Liu. Context as memory: Scene-consistent interactive long video generation with memory retrieval. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*, pp. 1–11, 2025.

Andrii Zadaianchuk, Maximilian Seitzer, and Georg Martius. Object-centric learning for real-world videos by predicting temporal feature similarities. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=t1jLRFvBqm.

Lvmin Zhang, Shengqu Cai, Muyang Li, Gordon Wetzstein, and Maneesh Agrawala. Frame context packing and drift prevention in next-frame-prediction video diffusion models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025a.

Weipu Zhang, Adam Jelley, Trevor McInroe, and Amos Storkey. Objects matter: Object-centric world models improve reinforcement learning in visually complex environments. In *Reinforcement Learning and Video Games Workshop @ RLC 2025*, 2025b. URL https://openreview.net/forum?id=C6aSY4DKDa.

Lin Zhao, Yushu Wu, Yifan Gong, Yanzhi Wang, and Pu Zhao. Omnimem: Scalable and adaptive memory retrieval for long video generation, 2026a. URL https://arxiv.org/abs/2605.30519.

Rongzhen Zhao, Zhiyuan Li, Juho Kannala, and Joni Pajarinen. Internalizing temporal consistency in video object-centric learning without explicit regularization, 2026b. URL https://arxiv.org/abs/2605.31508.

Yanpeng Zhao, Siyu Gao, Yunbo Wang, and Xiaokang Yang. Dynavol: Unsupervised learning for dynamic scenes through object-centric voxelization. In *International Conference on Learning Representations*, volume 2024, pp. 4009–4028, 2024.

A Extended Related Work

This section supplies the broader literature and mechanism-level details behind the compressed summary in Section 2. It expands the citation base of the two lines that the main text only sketches, and states where each line stops short of the memory primitive used by SlotMemory.

A.1 Streaming Video Generation and Memory Mechanisms

From clip synthesis to streaming rollouts. Early text-to-video diffusion models generate isolated clips (Ho et al., 2022b; Singer et al., 2022; Ho et al., 2022a; Wu et al., 2023; Guo et al., 2023; Blattmann et al., 2023; Wang et al., 2023; Bar-Tal et al., 2024; HaCohen et al., 2024; Wan et al., 2025; Jin et al., 2025; Chen et al., 2025). Because each rollout remains bounded, long-video generation has converged on autoregressive streaming frameworks that synthesize chunks under a fixed budget and carry state across chunks through KV caches. These systems differ mainly in how they stabilize the rollout: FIFO-Diffusion generates infinite video without training by a queue-based pipeline (Kim et al., 2024); StreamingT2V adds appearance and motion conditioning for extendable narratives (Henschel et al., 2025); CausVid distills a bidirectional model into a few-step causal generator (Yin et al., 2025); Self-Forcing trains on self-generated rollouts to close the train–test gap (Huang et al., 2026); Infinity-RoPE controls positional phases for unbounded, action-controllable rollouts (Yesiltepe et al., 2025); Deep Forcing combines deep sink tokens with participative compression in a training-free manner (Yi et al., 2025); Gu et al. (2025) cast the problem as next-frame prediction; FramePack packs frame context and suppresses drift (Zhang et al., 2025a); TTT-Video updates a test-time memory for one-minute generation (Dalal et al., 2025); and Magi-1 (Teng et al., 2025) and Inflvg (Fang et al., 2025) scale or reinforce autoregressive rollouts.

Memory organization. Where these frameworks keep history, they do so through explicit banks. LongLive maintains an interactive KV cache with sink protection (Yang et al., 2026a); Context-as-memory retrieves scene-consistent context for interactive generation (Yu et al., 2025); Mixture-of-Contexts composes multiple contexts for long video (Cai et al., 2025); MemFlow adapts the memory flow to the narrative (Ji et al., 2025); and Anchor Forcing combines anchor memory with a tri-region RoPE (Yang et al., 2026b). The memory mechanisms summarized in Section 2 differ in what they address and what they store. LongLive-RAG retrieves self-generated latents with a learned retrieval embedding and a windowed temporal-delta loss (Hu et al., 2026); OmniMem selects a sparse subset of the full historical KV cache and shares the selection across heads (Zhao et al., 2026a); FadeMem consolidates KV blocks into a distance-aware hierarchy that keeps near history dense and far history sparse (Lu et al., 2026); MemRoPE compresses past keys into long- and short-term exponential-moving-average tokens and re-indexes them online with RoPE (Kim et al., 2026); SWIFT injects prompt-adaptive segment anchors into a semantic cache (Tan et al., 2026); Echo-Forcing maintains scene-level anchors and recall frames with difference-aware decay (Wu et al., 2026); EM-Vid indexes entity-relevant latent patches and attends only to them, without training (Vandersanden et al., 2026); and Memento trains subject reconstruction together with a dual-query memory (Wei et al., 2026). What distinguishes SlotMemory from this spectrum is not the budget but the addressing content: these designs address frames, chunks, scenes, or externally stored summaries, and the entity-centric designs listed above index entities outside the attention stack; SlotMemory instead assigns entity-level addresses inside the diffusion transformer and keeps the stored values as original KV tokens.

A.2 Object-Centric Representation Learning

Foundational object-centric models decompose a scene into slots (Burgess et al., 2019; Greff et al., 2019; Engelcke et al., 2020; Locatello et al., 2020). The subsequent literature extends them along four directions that are only summarized in Section 2. First, temporal and real-world video: conditional video slot attention (Kipf et al., 2021), SAVi++ for end-to-end learning from real videos (Elsayed et al., 2022), SlotFormer for visual dynamics simulation (Wu et al., 2022), and DINOSAUR-style models that scale object-centric learning to natural images and their successors (Seitzer et al., 2022; Didolkar et al., 2025b), with temporally consistent slot contrast (Manasyan et al., 2025) and grounded or noise-regularized variants (Kori et al., 2024a). Second, geometric structure: slot-conditioned diffusion decoders (Jabri et al., 2024), object-centric voxelization for dynamic scenes (Zhao et al., 2024), and movable-object inference for 3D scene representation (Chen et al.,

2024). Third, identifiability and disentanglement: probabilistic slot attention (Kori et al., 2024b) and explicitly disentangled slot representations (Majellaro et al., 2025), together with recent work that separates identity from pose or static from dynamic factors (Li et al., 2026b; Zhao et al., 2026b) and object-centric world models (Zhang et al., 2025b). Fourth, generative and language conditioning: language-controllable slot inference (Didolkar et al., 2025a) and diffusion-coupled slots with register tokens and contrastive alignment (Nguyen et al., 2026). Across all four directions, the slot is the representation that is reconstructed, predicted from, or decoded; memory is therefore bounded by the fidelity of that representation. SlotMemory inverts this arrangement by keeping the content in the original KV space and letting slots act only as routing keys, so the same semantic decomposition can be queried by attention without a low-dimensional reconstruction bottleneck.

B Method Specification

This appendix specifies the exact instantiations, ordering, and edge-case behavior that Section 3 abstracts over. Equations already given there are referenced rather than repeated, and all values are the defaults used in the reported experiments.

B.1 Notation and Architecture Placement

Units. A *token* is one patchified latent position. With patch size $(1, 2, 2)$ on a 60×104 latent grid, one latent frame contains 1,560 tokens, so a generated chunk of T_c=3 latent frames contains N=4,680 tokens. A *slot* is one of S=7 semantic groups produced per written chunk, described by a d_s=64-dimensional address and R=64 selected original K/V tokens. An *entity* is a persistent identity with a d_s-dimensional key, a scalar quality estimate, a lifecycle state, and up to M=4 stored K/V snapshots. The Wan2.1-T2V-1.3B backbone (Wan et al., 2025) uses L=30 layers with H=12 attention heads and D=128 channels per head ($H{\cdot}D$=1,536); the batch size is B=1 in all reported runs.

States. *Local cache* $\mathcal{C}_l^c$ holds a fixed sink prefix (3 frames) (Yang et al., 2026a) and a rolling window (12 frames). *Bank* $\mathcal{B}_l^c$ is the attention-visible memory: on semantic layers it holds at most N_{budget}=6 memory blocks of 7 groups$\times$64 tokens, i.e. 42 groups (2,688 K/V tokens); on raw layers it holds the most recent $3\times$1,560=4,680 tokens. *Ordinary history* $\mathcal{H}_l^c$ is a notational abstraction for the non-retrieved part of the bank; the implementation stores it inside the same K/V tensors rather than in a separate object. A *candidate* is the set of S groups with their write scores produced by compressing one chunk, and an *entity snapshot* is one stored K/V group of 64 tokens belonging to an entity.

Layer placement. Semantic memory is active on layers $\ell \in [L/3,\ 2L/3] = [10, 20]$, i.e. 11 of the 30 layers (the bound is inclusive in the implementation). These layers run slot compression and entity tracking independently; all remaining layers keep only raw KV windows and maintain neither slots nor entities. Every layer owns its own cache, bank, slot state, and (where applicable) entity table. Table B.1 summarizes the resulting tensor shapes.

B.2 Streaming State and Causal Update Order

Memory timing is controlled by two events. A *read event* occurs only on the *first* denoising step of a chunk; it consumes any pending clean K/V and rebuilds the bank before self-attention reads it. A *write event* occurs only for chunks whose starting frame index is a multiple of the write period; with the default period of three chunks (9 latent frames), one chunk in three stages its clean K/V as pending memory. A chunk therefore never attends to memory it produced itself: states written by chunk c become visible at the first denoising step of chunk $c + 1$, and chunks outside the write schedule contribute to the local cache but not to the bank.

The implemented per-chunk order is:

1. Set the active prompt. At a switch point, reset the text-conditioning cache, recompute it with the new prompt, and re-encode the most recent local context under the new prompt (up to 12 frames at inference and up to 21 during training); the bank, slot state, and entity bank are left untouched.
2. At the first denoising step, consume the pending clean K/V from the previous write event. On semantic layers, compress it, retrieve entity snapshots, assemble the bounded bank, and

Table B.1: **Tensor shapes in SlotMemory.** Values are the defaults of the reported runs.

Tensor	Shape	Value
Clean K/V of one chunk	$B \times N \times H \times D$	N=4,680, $H \cdot D$=1,536
Slot latent Z_l^c	$B \times N \times d_s$	d_s=64
Slot descriptors U_l^c	$B \times S \times d_s$	S=7
Assignments A_l^c	$B \times S \times N$	—
Selected slot K/V	$B \times S \times R \times H \times D$	R=64
Bank K/V $\mathcal{B}_l^c$	$B \times N_{\text{bank}} \times H \times D$	$N_{\text{bank}} \leq$ 2,688
Entity keys	$B \times E \times d_s$	E=32
Entity snapshots	$B \times E \times M \times R \times H \times D$	M=4
Write scores	$B \times S$	—

commit the new slots to the entity bank. On raw layers, append the pending K/V to the raw window and truncate it to 4,680 tokens.

3. Denoise the chunk while self-attention reads [sink; bank; local]. Steps after the first reuse the same bank and do not touch memory.
4. Re-encode the generated chunk with clean context to obtain $(\bar{K}_l^c, \bar{V}_l^c)$, and write these clean states into the local cache.
5. If chunk c belongs to the write schedule, stage $(\bar{K}_l^c, \bar{V}_l^c)$ as pending memory; otherwise the bank is unchanged.

B.3 Slot Construction and K/V Token Selection

Compression operates on the clean K/V of a written chunk. Heads are flattened so that a complete token, rather than an individual head, is assigned to a slot. The input adapter ϕ_l is a two-layer MLP, $1{,}536 \to 4{,}096 \to 64$ (LayerNorm and ReLU after the first layer), and Slot Attention (Locatello et al., 2020) uses S=7 slots, 2 iterations, a GRU update, and ε=10^{-6}. Slots are initialized from the previous chunk's clean slot descriptors (detached) when available (Manasyan et al., 2025), otherwise from a learned initializer. The output adapter ψ_l is $64 \to 1{,}024 \to 1{,}024 \to 1{,}024 \to 1{,}536$ with ReLU activations.

For slot s and token t, the implemented selection score is $\rho_{s,t} = 1.5 \cos(\psi_l(u_s), \bar{k}_t) + 0.3 \log(A_{s,t} + 10^{-6})$, i.e. the reconstruction term is weighted 1.5 and the assignment-log prior 0.3. Each slot keeps its top R=64 tokens by ρ, giving 448 selected tokens per chunk; the selected entries are gathered from the original K/V tensors. Selection is hard: the top-R indices are non-differentiable, while gradients still reach the selected K/V values; the reconstruction loss uses a straight-through hard assignment whose forward pass takes the top-1 slot but whose backward pass flows through the soft weights. No explicit tie-breaking rule is imposed, so ties follow the implementation's top-k ordering.

Each slot also receives a write quality that combines coverage and reconstruction fidelity: the coverage term is the slot's share of total assignment mass, and the reconstruction term is the assignment-weighted cosine similarity between the decoded prototype and the clean keys, mapped to $[0, 1]$. The two terms are combined as $w = \text{clamp}(\text{coverage} + 0.5 \cdot \text{reconstruction}, 0, 1)$ and used both for later retention and for entity admission.

B.4 Entity Association and Lifecycle

Entity tracking runs independently per semantic layer. Each layer maintains at most $E_{\max}$=32 entities, and each entity stores at most M=4 K/V snapshots. Association compares the previous chunk's slot descriptors with entity keys by cosine similarity; a small index prior (0.03) is added only for ACTIVE entities seen in the immediately preceding step. Candidate pairs are filtered by a state-dependent gate and solved by Hungarian matching with a dummy column that absorbs unmatched slots. By default, DORMANT entities are excluded from association, so a returning entity is not re-identified; its stored snapshots remain retrievable instead.

Table B.2: **Entity-bank defaults.** Thresholds are cosine similarities unless noted.

Parameter	Value
Entity capacity $E_{\max}$ / snapshots per entity M	32 / 4
Association gate (ACTIVE, TENTATIVE)	0.70
Association gate for DORMANT (re-ID disabled by default)	0.78
TENTATIVE $\rightarrow$ ACTIVE	2 matched chunks
TENTATIVE removal	>2 consecutive unmatched chunks
ACTIVE $\rightarrow$ DORMANT	2 consecutive unmatched chunks
Identity-key EMA base rate / minimum similarity	0.10 / 0.70
Snapshot write: write quality / match similarity	≥ 0.40 / ≥ 0.50
Admission of an unmatched slot to TENTATIVE	write quality ≥ 0.40

Lifecycle decisions are per layer. A new TENTATIVE entity is born when an unmatched slot has sufficient write quality, unless it is a near-duplicate of an already matched entity (similarity > 0.97 with low quality). A TENTATIVE entity becomes ACTIVE after two matched chunks and is cleared after more than two consecutive unmatched chunks. An ACTIVE entity becomes DORMANT after two consecutive unmatched chunks. Only DORMANT entities and low-quality TENTATIVE entities are evictable; among them, the lowest retention score is removed first, where retention combines quality, match count, and dormancy age. ACTIVE entities are never evicted, and unmatched slots are discarded rather than replacing a confirmed identity when the table is full.

Matched entities update their identity key by an EMA whose rate is proportional to write quality and match similarity, and only when that similarity exceeds 0.70; quality itself follows a $0.9/0.1$ EMA. A snapshot is written when both write quality and match similarity pass the thresholds in Table B.2; when all M snapshot slots are occupied, the snapshot with the lowest quality-plus-age score is replaced. Snapshots store original K/V groups, so retrieval returns transformer-compatible content rather than embeddings.

B.5 Retrieval, Memory Assembly, and Prompt Switching

Retrieval is performed only in no-gradient forwards and, in the implementation, before the current chunk's slots are committed; chunk c therefore queries the entity state that was available at chunk $c-1$. Each candidate entity is scored by its best slot similarity plus small bonuses for quality, match count, and a current-chunk match; snapshot scores add snapshot quality, a recency bonus that decays linearly over 50 chunks, and a match bonus. Selection proceeds in three quota rounds—matched entities, persistent non-matched entities (ACTIVE or DORMANT), and recent high-quality snapshots— followed by a global backfill up to the history budget. At most one snapshot per entity is returned. Because DORMANT entities are not matched but are eligible in the persistent round, their snapshots can still be restored after the entity has left the scene.

The bank is assembled under the budget N_{budget} of Section B.1. When retained history plus the new candidates already fits, they are concatenated directly. Otherwise, the new chunk contributes 7 groups, leaving $42 - 7 = 35$ groups for history. Retrieved entity snapshots take priority; the remaining budget is filled by history groups ranked by their stored write quality, falling back to the text–visual score (0.7 text, 0.3 visual) when no write score exists. Retrieved and history groups are deduplicated by the cosine similarity of their mean-pooled keys: a history group with similarity above 0.98 to a retrieved snapshot is dropped and replaced by the next-ranked group. The final bank is the concatenation of retrieved snapshots, retained history, and the new groups, in that order, and its entries remain original K/V tokens.

At a prompt switch, the text-conditioning cache is recomputed and the recent local window is re-encoded under the new prompt; the bank, slot state, and entity bank are preserved. Thus the switch changes what the scene should become, while entity memory preserves what established entities should continue to look like.

Positional handling. Clean keys are stored after rotary position embedding has been applied at their source positions, and bank keys are concatenated with the sink and local segments without

Algorithm 1 Causal Streaming Generation with Hybrid Persistent Memory

Require: prompts $\{p^{(m)}\}_{m=1}^{M}$, switch points $\mathcal{S}$, causal generator G
Ensure: video latents $V^1, \ldots, V^C$

initialize all local/text caches, the entity bank, and the memory bank to empty
for $c = 1$ **to** C **do**
  set the active prompt p_c for chunk c
  if $c = 1$ **then**
    $\mathcal{C}_l^{\text{text},1} \leftarrow \text{EncodeText}(p^{(1)})$ for all l
  else if chunk c begins at a switch point s_m **then**
    $\mathcal{C}_l^{\text{text},c} \leftarrow \text{EncodeText}(p^{(m+1)})$ for all l
    recompute the recent local context under $p^{(m+1)}$ ▷ the entity bank and persistent snapshots are retained
  else
    $\mathcal{C}_l^{\text{text},c} \leftarrow \mathcal{C}_l^{\text{text},c-1}$ for all l
  end if
  if $c > 1$ **then**
    for all semantic layers l **do**
      $(U_l^{c-1}, \mathcal{W}_l^{c-1}) \leftarrow \text{Compress}\big(\bar{K}_l^{c-1}, \bar{V}_l^{c-1}, \bar{U}_l^{c-2}\big)$ ▷ $\bar{U}_l^0$ is the learned initializer
      $\bar{U}_l^{c-1} \leftarrow U_l^{c-1}$
      $\mathcal{R}_l^c \leftarrow \text{Retrieve}\big(U_l^{c-1}, \mathcal{E}_l^{c-1}\big)$
      $(\mathcal{B}_l^c, \mathcal{H}_l^c) \leftarrow \text{Assemble}\big(\mathcal{R}_l^c, \mathcal{H}_l^{c-1}, \mathcal{W}_l^{c-1}\big)$
      $\mathcal{E}_l^c \leftarrow \text{UpdateEntities}\big(\mathcal{E}_l^{c-1}, U_l^{c-1}, \mathcal{W}_l^{c-1}\big)$
    end for
    for all shallow and deep layers l **do**
      $\mathcal{B}_l^c \leftarrow \text{UpdateRawWindow}\big(\mathcal{B}_l^{c-1}, \bar{K}_l^{c-1}, \bar{V}_l^{c-1}\big)$
    end for
  else
    $\mathcal{B}_l^1 \leftarrow \emptyset, \mathcal{H}_l^1 \leftarrow \emptyset$ for all layers l ▷ no historical memory before the first chunk
  end if
  $V^c \leftarrow \text{Denoise}\big(p_c, \{\mathcal{C}_l^c\}_l, \{\mathcal{B}_l^c\}_l, \{\mathcal{C}_l^{\text{text},c}\}_l\big)$
  obtain $(\bar{K}_l^c, \bar{V}_l^c)$ for all l via $\text{CleanReencode}(V^c)$
  update the local caches and stage $\bar{K}_l^c, \bar{V}_l^c$ as pending memory ▷ memory from chunk c is read only from chunk $c+1$ onward
end for

re-indexing. Retrieved snapshots therefore keep the RoPE phases of the chunks in which they were written; the implementation does not remap them to the current chunk's positions.

Worked example. Consider an entity that is matched at chunks 1–3 and then leaves the scene. After two consecutive unmatched chunks it becomes DORMANT, which removes it from ordinary association but keeps its snapshots. During the absence, its snapshots can still be returned by the persistent quota if their visual relevance is high. When the entity reappears, the slot is again unmatched because DORMANT entities are not re-identified by default; it either initializes a new TENTATIVE entity (if its write quality passes the admission threshold) or is discarded. Visual continuity therefore comes from restoring the dormant snapshots into the attention bank, not from re-associating the returning slot with the old identity—a behavior that also defines a limitation of the current implementation.

B.6 Algorithm

Algorithm 1 summarizes the per-chunk procedure whose states and update order are specified in Section B.2.

C Training and Implementation

C.1 Streaming Rollout and Supervised Region

Memory is trained on the model's own rollout rather than on ground-truth video. Each iteration advances a virtual sequence of up to 240 latent frames: the generator synthesizes a chunk, writes the completed chunk's clean state into memory, and conditions subsequent chunks on that memory, so slot routing and entity retrieval are exercised under the same self-generated distribution as at inference. A supervised chunk spans 21 latent frames, of which at least 18 are newly generated; the remaining leading frames are copied from the previous chunk as overlap and serve as context only. Only newly generated frames receive gradients: both the distillation objective and the fake-score objective are averaged over a binary frame mask that excludes the overlap, and overlap frames are detached. Before scoring, the first frame of the window is decoded and re-encoded through the VAE to reduce the train–inference discrepancy of the conditioning frame. Prompt switches are sampled inside the rollout from the frame set $\{21, 39, \ldots, 201\}$, so memory is trained on early, middle, and late transitions within a sequence.

C.2 Training Objectives

Three networks participate. The generator G_θ and the fake-score network C_ϕ are both 1.3B variants, whereas the real-score teacher R is a frozen 14B model. C_ϕ is updated every optimization step and G_θ every fifth step, so the two alternate on the same self-generated sequence.

Generator distribution matching. The generator uses a flow-matching parameterization: for a clean latent x, noise ε, and noise level σ_t, $x_t = (1-\sigma_t)x + \sigma_t\varepsilon$, $v^\star = \varepsilon - x$, and a prediction $\hat{v}$ is converted to $\hat{x}_0 = x_t - \sigma_t\hat{v}$. Both score networks predict x_0 at the same x_t, with classifier-free guidance applied to the real score (w_{real}=3.0) and disabled for the fake score (w_{fake}=0). The distribution-matching direction is

$$g = \frac{\hat{x}_0^{\text{fake}} - \hat{x}_0^{\text{real}}}{\text{mean}|x - \hat{x}_0^{\text{real}}|}, \tag{7}$$

followed by non-finite-value replacement, and the generator loss is

$$\mathcal{L}_{\text{DMD}} = \tfrac{1}{2}\,\mathbb{E}\big[(x - \text{sg}(x-g))^2\big], \tag{8}$$

where the target is stop-gradient. The expectation is a masked mean over newly generated frames. Noise levels are integers sampled uniformly from $[0, 1000)$, passed through the shift transformation with factor 5.0, and clamped to $[20, 980]$.

Fake-score denoising. The fake score is trained on detached rollouts by regressing the flow target of the generated sample, $\mathcal{L}_{\text{critic}} = \mathbb{E}\big[(\hat{v}_\phi - (\varepsilon - x))^2\big]$ with $\hat{v}_\phi = (x_t - \hat{x}_0^{\text{fake}})/\sigma_t$, again averaged over newly generated frames only. The critic objective contains no memory terms.

Slot regularization. Let U and A be the current chunk's slot descriptors and soft assignments, and let $\bar{U}$ denote the previous chunk's clean slots, which are detached. Three terms supervise the compressor; each is accumulated once per differentiable forward and averaged over the forwards within an optimization step. (i) Temporal consistency treats the same-index slots of consecutive chunks as positives and all $B{\cdot}S$ slots in the batch as negatives (Manasyan et al., 2025), with ℓ_2-normalized slots and temperature τ_c=0.1:

$$\mathcal{L}_{\text{con}} = -\frac{1}{BS}\sum_{i=1}^{BS} \log \frac{\exp\big(\text{sim}(\bar{u}_i, u_i)/\tau_c\big)}{\sum_{j=1}^{BS}\exp\big(\text{sim}(\bar{u}_i, u_j)/\tau_c\big)}. \tag{9}$$

(ii) Feature fidelity decodes each slot with ψ_l and reconstructs tokens under a straight-through hard assignment (hard forward, soft backward); the per-token loss is $1 - \cos(\hat{x}_{b,t}, \text{sg}(\bar{K}_{b,t}))$, weighted by the top-1 assignment confidence clamped at 0.1 and by an area weight that compensates small-token slots (exponent 0.25, normalized to mean one, capped at five). (iii) Selection distillation matches the token-selection distribution to the tokens the chunk actually uses: the teacher attends the chunk's own clean keys and values with its clean queries and weights attention mass by value magnitude, the student is a softmax over the selection score ($1.5\cos + 0.3\log A$) with temperature 0.2, and the loss is their cross-entropy over tokens, averaged over batch and slots. The teacher is computed without gradients.

Table C.3: **Loss terms and their roles.** "Generator updates" means the steps on which G_θ is optimized.

Loss	Target behavior	Inputs	Active	Weight
$\mathcal{L}_{\text{DMD}}$	match the generator distribution to the teacher	generated latents, noise level, prompt	generator updates	1.0
$\mathcal{L}_{\text{critic}}$	fit the fake score to generated samples	detached rollouts, flow target	every step	1.0
$\mathcal{L}_{\text{con}}$	keep slot identity across chunks	consecutive-chunk slot descriptors	all compressor updates	λ_{slot}
$\mathcal{L}_{\text{rec}}$	ground slots in transformer features	clean keys, decoded prototypes, assignments	unfrozen-grouper updates	$0.01\lambda_{\text{slot}}$
$\mathcal{L}_{\text{dist}}$	align selection with actual token usage	clean queries/keys/values, selection scores	unfrozen-grouper updates	$0.01\lambda_{\text{slot}}$

The generator objective is therefore

$$\mathcal{L}_{\text{gen}} = \mathcal{L}_{\text{DMD}} + \lambda_{\text{slot}}\big(\mathcal{L}_{\text{con}} + 0.01\,\mathcal{L}_{\text{rec}} + 0.01\,\mathcal{L}_{\text{dist}}\big), \qquad \lambda_{\text{slot}} = 0.2. \tag{10}$$

Table C.3 summarizes all losses, their inputs, their active stage, and their weights.

C.3 Frozen Modules and Optimization Schedule

The teacher, the text encoder, and the VAE are frozen. The 1.3B generator and fake-score backbones are also frozen and adapted only through LoRA (rank 256, α=256, dropout 0, bf16) applied to the attention blocks; the memory compressor is trained in full. Training proceeds in two stages over 3000 updates. Stage-I (the first 1000 updates) freezes the slot grouper, the initializer, and the dynamics predictor, so the input/output adapters and diffusion-side parameters adapt to a fixed slot vocabulary; only the contrastive term is accumulated, while reconstruction and selection distillation are disabled. Stage-II (the remaining 2000 updates) unfreezes these modules and resumes from the Stage-I checkpoint, enabling end-to-end slot optimization with all three memory losses. The stage boundary is implemented by a configuration flag plus an early-stopping check at update 1000. Throughout, "freezing the slot grouper" refers exactly to these three modules; the adapters are never frozen. The entity bank has no trainable parameters: it is a state machine updated without gradients, and it is therefore not optimized by any loss. EMA is disabled in the LoRA configuration, and the two backbones are optimized alternately as described above.

C.4 Training Recipe

The reported model starts from a released causal base checkpoint and initializes the slot modules from the slot-attention checkpoint of Manasyan et al. (2025), pretrained on real videos. Training takes 3000 updates on 8 GPUs with FSDP, split into the 1000-update frozen-grouper stage and the 2000-update end-to-end stage described above. The generator uses AdamW with learning rate 1×10^{-5} and the critic with 2×10^{-6} (β_1=0, β_2=0.999, weight decay 0.01), batch size 1, and gradient accumulation 1. Checkpoints are written every 50 updates (keeping the most recent three) and validation videos every 500 updates.

The streaming rollout uses chunks of 21 latent frames with a maximum sequence length of 240 frames, at least 18 newly generated frames per chunk, and 3 latent frames per generation block. Each block is denoised with the step schedule $\{1000, 750, 500, 250\}$ and a clean re-encoding pass at zero context noise. Prompt switches are drawn uniformly from frame indices $\{21, 39, \ldots, 201\}$, covering early, middle, and late transitions. The teacher is a frozen Wan2.1-T2V-14B model (Wan et al., 2025), the fake score is a 1.3B model, and the generator is a causal 1.3B model; guidance is 3.0 on the real score and disabled on the fake score, and the timestep shift is 5.0 with denoising loss type `flow`.

D Additional Quantitative Results

This section extends the main quantitative evaluation along the two axes that Section 4.2 does not cover: horizon scaling beyond 60 s and end-to-end efficiency.

D.1 Long-Horizon Scaling

The 30 s and 60 s results in Section 4.2 show that memory matters more as the rollout lengthens, but they cannot distinguish whether the advantage of entity-addressed memory keeps accumulating

Table D.4: **Quantitative comparison on long video generation.** We report VBench-Long (Huang et al., 2025) metrics. Within each base model group, best results are bold and second best are underlined. Baseline results are taken from MemRoPE (Kim et al., 2026) and LongLive-RAG (Hu et al., 2026). "–" indicates unavailable results and is counted as the last rank when computing Avg. Rank.

Method	Subject Consistency ↑	Background Consistency ↑	Motion Smoothness ↑	Dynamic Degree ↑	Aesthetic Quality ↑	Imaging Quality ↑	Avg. Rank ↓
120 seconds							
Results on LongLive (Yang et al., 2026a)							
LongLive (Yang et al., 2026a)	96.93	95.64	98.58	47.12	57.90	66.95	4.08
Deep Forcing (Yi et al., 2025)	97.17	95.72	98.57	46.03	56.98	66.03	4.58
Infinity-RoPE (Yesiltepe et al., 2025)	96.81	95.65	98.48	53.59	56.73	66.19	5.00
MemRoPE (Kim et al., 2026)	97.54	**96.29**	**98.73**	–	59.25	**69.43**	**2.17**
LongLive-RAG (Hu et al., 2026)	97.22	95.88	98.62	50.25	57.57	66.95	2.92
SlotMemory	**97.59**	95.72	98.56	**78.26**	**62.27**	68.92	2.25
240 seconds							
Results on LongLive (Yang et al., 2026a)							
LongLive (Yang et al., 2026a)	97.02	95.80	98.52	33.21	56.70	66.90	3.83
Deep Forcing (Yi et al., 2025)	96.65	96.03	98.50	39.17	57.75	66.48	3.50
Infinity-RoPE (Yesiltepe et al., 2025)	97.11	96.34	**98.99**	–	57.47	63.38	3.33
MemRoPE (Kim et al., 2026)	97.37	**96.39**	98.59	38.25	58.90	**68.93**	**1.83**
SlotMemory	**97.67**	95.77	97.71	**77.96**	**61.97**	68.57	2.50

with horizon or saturates while temporal memories continue to decay. Table D.4 separates these possibilities at 120 s and 240 s, and the answer is the second: our scores are essentially horizon-invariant whereas every baseline that reports the diagnostic dimension degrades, so what grows with horizon is the separation between methods, not our absolute quality. This is the distinction the long-horizon setting is designed to isolate.

The six metrics divide into two families that respond oppositely to what a memory stores. Subject Consistency, Dynamic Degree, and Aesthetic Quality measure whether appearance and motion survive; Background Consistency and Motion Smoothness reward re-using or stabilizing scene content, and Motion Smoothness is highest for video that changes little. SlotMemory leads the first family at both horizons (Subject 97.59/97.67, Dynamic 78.26/77.96, Aesthetic 62.27/61.97) but ranks fifth in the second at 240 s, with Background Consistency 95.77 and Motion Smoothness 97.71. The second-family deficit is small in absolute terms—0.62 behind the best background score and 0.79–1.28 behind the smoothest method—whereas the motion gap is 38.8–44.7 points. This asymmetry is the signature of the design: a bank that stores entity-bearing K/V tokens instead of replaying recent scene content leaves the scene free to move, so it gives up some smoothness and background stability in exchange for motion. The same trade-off is already visible at 60 s, where the replay-based baselines collapse in Dynamic Degree (LongLive 28.30, MemFlow 26.05) while SlotMemory reaches 81.38; at 240 s it becomes the dominant axis of separation.

The scaling pattern shows that this behaviour is not a transient of the 30–60 s regime. Across horizons our Dynamic Degree is 79.30 at 30 s, 81.38 at 60 s, 78.26 at 120 s, and 77.96 at 240 s—a spread of 3.4 points over an eightfold increase in length (the 60 s protocol is interactive while the others are single-prompt)—and our Subject Consistency and Aesthetic Quality change by at most 0.35 points between the two long horizons. The baselines move in the opposite direction: between 120 s and 240 s LongLive loses 13.9 Dynamic points and Deep Forcing 6.9, and the only Dynamic value reported by MemRoPE at 240 s is 38.25. Two consequences follow. First, appearance metrics saturate (every method is at or above 96.6 on Subject Consistency), so they stop discriminating and the informative signal moves to the motion dimensions. Second, a fixed bank of 42 groups per semantic layer already supports 240 s without the decay observed in the baselines, which is the long-horizon form of the paper's capacity argument: what binds is what the memory stores, not how long the video is, because entity addresses are independent of temporal distance.

The average-rank column must be read against this split, since it weights both families equally. At 240 s SlotMemory is first on Subject Consistency, Dynamic Degree, and Aesthetic Quality but fifth on Background Consistency and Motion Smoothness, which yields a rank of 2.50; MemRoPE attains 1.83 by leading Background and Imaging Quality while not reporting Dynamic Degree at 120 s and reaching only 38.25 at 240 s. A memory that keeps a stable, slowly changing scene can therefore improve an aggregate rank without preserving the motion that the dynamic metrics are

Table D.5: Efficiency comparison for interactive long video generation on a single NVIDIA H200 GPU. We report the average latency, average peak GPU memory usage, and diffusion-only throughput.

Method	Latency (s)↓	Peak VRAM (GB)↓	Diffusion FPS↑
LongLive (Yang et al., 2026a)	73.040 ± 0.398	33.24	18.3
MemFlow (Ji et al., 2025)	81.423 ± 0.000	31.47	16.2
Anchor Forcing (Yang et al., 2026b)	88.269 ± 0.354	34.15	14.6
SlotMemory	83.005 ± 0.281	36.82	14.9

designed to capture. We accordingly treat the per-dimension pattern, rather than the aggregate rank, as the evidence about memory quality.

D.2 Efficiency and Memory Overhead

Table D.5 measures the interactive protocol on a single H200 GPU. SlotMemory completes the sequence in 83.0 s at 14.9 diffusion FPS, which is 13.6% slower than LongLive (73.0 s, 18.3 FPS) and 1.9% slower than MemFlow (81.4 s, 16.2 FPS), while remaining 6.0% faster than Anchor Forcing (88.3 s, 14.6 FPS). Its peak memory of 36.8 GB is 3.6, 5.4, and 2.7 GB above LongLive, MemFlow, and Anchor Forcing, respectively, the largest relative increase (17.0%) being over MemFlow. The structured memory therefore adds a bounded overhead: the method's latency stays within the range of the evaluated memory baselines and it remains faster than one of them, at the cost of the largest peak memory among the four measured systems.

D.3 RoPE and Positional Encoding Analysis

A remaining question about the entity bank is whether its identity key is confounded by absolute position. Keys are rotated by RoPE before they are cached, and the identity pathway reads the same cached tensors, so a slot descriptor summarizes content whose phase encodes *when* it occurred as well as *what* it is. Because association is decided by the cosine similarity between slot descriptors, two occurrences of the same entity that differ only in temporal phase can be separated even when their content is identical. The concern is therefore one of matching rather than attention: an entangled *who–when* representation could weaken long-range association and complicate the matching gate.

We examined this question with three configurations that share the attention payload (RoPE-rotated K/V) and differ only in how the identity key is obtained: grouping slots from rotated keys and using the resulting descriptor, which is the configuration adopted in this work; grouping from rotated keys but pooling pre-rotation keys for identity; and deriving both grouping and identity from pre-rotation keys. Table D.6 reports the comparison on the entity-bank variant without selection distillation.

The interventions change the consistency dimensions in the direction predicted by the confound hypothesis, but only locally. Deriving identity from pre-rotation slots yields the highest Subject Consistency (93.31), and pooling pre-rotation keys yields the lowest alignment drift (6.98), confirming that the absolute phase affects long-range matching. Both interventions, however, cost more than they recover: the pre-rotation-slot variant gives up 3.93 points of Dynamic Degree and 0.51 points of VBench Quality relative to the adopted configuration, and the pooled-key variant gives up 0.19 points of Quality for its 0.86-point reduction in drift. Across the three configurations the adopted design remains the strongest on average alignment (66.63), Dynamic Degree (92.96), and overall quality (81.50), while the alternatives lead only on individual dimensions.

We therefore retain the coupled form, in which the identity key is computed from the same rotated representation in which the recalled snapshot is consumed by attention. This keeps association and retrieval in one space: a match then means that the content compared is the content that will be attended to. Decoupling identity from the payload moves the matcher into a space the generator does not attend in, which is consistent with the alternatives' local gains not translating into aggregate quality; deriving grouping from pre-rotation keys additionally changes slot formation, and this variant loses the most motion. The phase is not pure nuisance either, since the relative phase be-

Table D.6: **Effect of the positional representation used for entity identity.** All configurations share the same attention payload (RoPE-rotated K/V) and differ only in how the identity key is obtained. Text alignment is averaged over the six 10 s segments and drift is the drop from 0–10 s to 50–60 s. Results use the entity-bank variant without selection distillation under the 60 s interactive protocol; the configuration adopted in this work is indicated and the best value per metric is shown in **bold**.

Identity key	**Slot grouping**	Text alignment ↑	Alignment drift ↓	Subject consistency ↑	Dynamic degree ↑	VBench quality ↑
Rotated-key slot (adopted)	Rotated keys	**66.63**	7.84	93.07	**92.96**	**81.50**
Pooled pre-rotation key	Rotated keys	66.59	**6.98**	92.89	91.42	81.31
Pre-rotation slot	Pre-rotation keys	65.85	7.33	**93.31**	89.03	80.99

tween two keys encodes their temporal proximity, a useful cue for associating the same entity across nearby chunks and for separating similar-looking entities.

E Human Evaluation Protocol

We conducted a blinded six-way human evaluation to assess the overall perceptual quality of long-form interactive videos generated by SlotMemory and five baseline methods. The study was designed to complement the automatic metrics by jointly evaluating perceptual properties that are difficult to capture with a single numerical measure, including long-term identity preservation, action completion, scene consistency, temporal stability, and object interaction.

E.1 Design and Stimuli

The evaluation used the 100 interactive scripts of the main benchmark. For each script we generated videos with six methods—SlotMemory, Anchor-Forcing (Yang et al., 2026b), IAMFlow (Liu et al., 2026), Infinity-RoPE (Yesiltepe et al., 2025), LongLive (Yang et al., 2026a), and MemFlow (Ji et al., 2025)—conditioned on exactly the same prompt sequence, and repeated generation with three seeds, giving $100 \times 3 = 300$ six-way comparison groups. Each group therefore contained one output per method under matched textual conditions, so differences within a group reflect the generation methods rather than the conditions.

Stimuli were normalized to a common spatial resolution, frame rate, encoding format, bitrate, and duration, and audio tracks were removed. Method names, filenames, watermarks, and other identifying metadata were stripped, and no method-specific post-processing was applied.

Each trial presented the six videos of one group anonymously in a 2×3 grid labelled A–F, with the method-to-position mapping randomized independently per trial and approximately balanced across the study to reduce position bias. All videos used identical player dimensions and playback settings; participants could replay them but had to watch each video in full before responding, and the interface exposed no method-specific information.

E.2 Procedure and Evaluation Criteria

We recruited 20 adult participants. Participation was voluntary, participants were informed of the procedure, the task, and their right to withdraw, and no sensitive personal information was collected.

Each participant evaluated 15 groups, giving $20 \times 15 = 300$ valid overall-preference judgments; each script–seed group was judged once, and both the assignment of groups to participants and the presentation order were randomized. After watching all six videos of a trial, participants selected the single best overall video, jointly considering five criteria: identity preservation (consistent appearance, clothing, colour, and role of recurring characters and objects); action completion (prompted actions visibly completed within the corresponding segments); scene consistency (background, layout, and persistent elements coherent across transitions unless the script requests a change); temporal stability (no flicker, abrupt appearance changes, deformation, or inconsistent transitions); and object interaction (plausible contact, spatial relations, and action consequences). Participants were explic-

Table E.7: Results of the blinded six-way human evaluation. Each of the 300 script–seed groups contributed one valid overall-preference label. Participants selected one video after jointly considering identity preservation, action completion, scene consistency, temporal stability, and object interaction.

Method	Selected Count	Preference Rate
SlotMemory	**81**	**27.0%**
Anchor-Forcing (Yang et al., 2026b)	62	20.7%
LongLive (Yang et al., 2026a)	52	17.3%
Infinity-RoPE (Yesiltepe et al., 2025)	43	14.3%
IAMFlow (Liu et al., 2026)	37	12.3%
MemFlow (Ji et al., 2025)	25	8.3%
Total	300	100.0%

itly told that stability is not the absence of motion and that a nearly static video should not be preferred for containing little variation; stability was to be judged together with whether the prompted actions, motion, and scene changes occurred. Each trial yielded a single overall-preference label rather than separate ratings per criterion.

E.3 Validity and Quality Control

Trials that could not be compared reliably, for example because of a playback failure, could be marked invalid; such responses were never treated as preferences, and the group was reassigned and re-evaluated until one valid judgment per script–seed group was obtained. Participant-level or trial-level data were excluded only under pre-specified criteria: an incomplete session, a recorded viewing duration indicating that the videos were not watched in full, a failed attention check, or severe inconsistency on repeated quality-control trials. Repeated trials served only to detect substantial inconsistency and were not part of the 300 preference labels. No response was excluded for disagreeing with the expected ranking or the authors' hypotheses. Because each group received one valid label, the study provides broad coverage across conditions rather than an estimate of inter-rater agreement.

E.4 Aggregation and Results

Each method's preference rate is $P_m = N_m/N$, where N_m is the number of valid trials in which the method was selected and $N = 300$; every valid trial selected exactly one method, so $\sum_m N_m = 300$. These rates are aggregate preference frequencies, not absolute quality scores.

Table E.7 reports the results. SlotMemory was selected in 81 of the 300 groups (27.0%), ahead of Anchor-Forcing (62; 20.7%), LongLive (52; 17.3%), Infinity-RoPE (43; 14.3%), IAMFlow (37; 12.3%), and MemFlow (25; 8.3%). SlotMemory was therefore the most frequently preferred method, by 6.3 percentage points over the second-ranked Anchor-Forcing and 9.7 points over LongLive.

F More Qualitative Evidence

This section provides the extended qualitative evidence behind the claims of Section 4.2: identity persistence across prompt switches, adherence to the changing instruction, motion, and visual quality. The comparisons follow the protocol and baselines of the quantitative interactive evaluation (Yang et al., 2026a; Ji et al., 2025; Yesiltepe et al., 2025; Liu et al., 2026; Yang et al., 2026b)—six prompts of 10 s and keyframes sampled at matched positions within each segment—so that all methods are examined at the same narrative moments. As shown in Figure F.1, Figure F.2, and Figure F.3, three scripts are included to cover complementary demands on memory rather than to maximize the margin: a dialogue-driven scene whose cast changes at a switch, a two-character scene with a brief transient event, and a low-motion scene with several objects.

What the failures reveal. Figure F.4 makes the numerical pattern of the core ablation concrete. Keeping history as untyped tokens allows the same visual evidence to be stored without an object-level owner, so the generator can reproduce an entity twice or, once the original subject has left the local window, drop it altogether; this is the regime in which multiple objects and a missing character appear. Grouping tokens into slots fixes the object count while the entities are still recent, but grouping alone does not decide that a slot in one chunk denotes the same object as a slot in another. When that correspondence is absent, the retrieved content can be semantically plausible while belonging to the wrong state, which surfaces as character mismatch and attribute drift. The entity bank is what turns a per-chunk decomposition into a persistent correspondence, so the visual difference between the ablated variants is not a difference in how much memory is kept but in what the kept states represent.

The distribution of the annotations is equally informative: they are sparse and segment-local rather than present in every frame, which is why an average over the whole sequence dilutes them while metrics restricted to the tail of the rollout separate the variants. This is the qualitative counterpart of the pattern in Table 4, where the average alignment barely moves but tail alignment and text drift change sharply: what the variants lose is not general visual quality but the ability to recall an entity correctly once recency can no longer substitute for memory.

Taken together, the comparisons show where the difference documented in Section 4.2 becomes visible: the compared methods retain the set or the layout of a scene while losing the identity of the characters that populate it, or they remain visually stable by changing very little, whereas the proposed method keeps the cast and the depicted action across the switches.



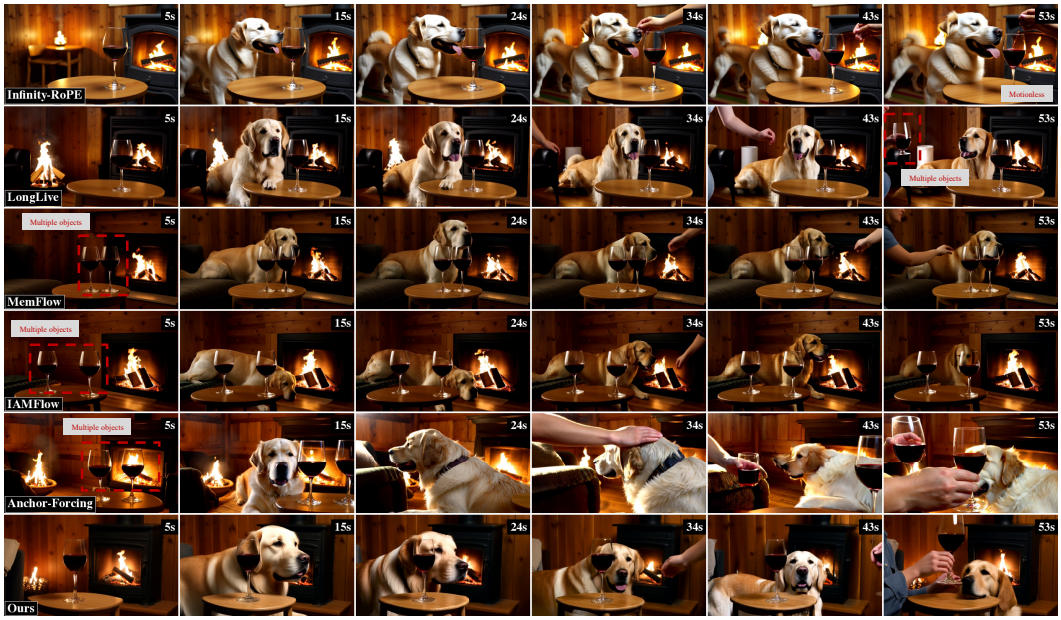
0-10s:**A young executive** in a sharp business suit is giving a presentation in a modern, glass-walled boardroom. She is pointing to a chart on a large screen. Her colleagues are watching attentively.
10-20s:**The young executive** giving the presentation in the boardroom is interrupted when **the company's charismatic CEO** unexpectedly enters the room with a broad smile. All the colleagues turn to look.
20-30s:**The young executive and the charismatic CEO** stand at the front of the boardroom. The CEO praises the executive's work and then makes a surprise announcement about a major new project. The colleagues look shocked and excited.
30-40s:**The young executive and the charismatic CEO** in the boardroom field questions from the excited colleagues. The CEO gestures for the young executive to answer, showing his confidence in her.
40-50s:**The young executive and the charismatic CEO** in the boardroom shake hands in front of all the colleagues. The CEO announces that the young executive will be leading the new project. She looks both honored and determined.
50-60s:**The young executive and the charismatic CEO** stand side-by-side in the boardroom as their colleagues erupt in applause. The young executive's career has just taken a dramatic and positive turn.

Figure F.1: **Extended comparison on a dialogue-driven script with a cast change.** The narrative moves from a presentation by a young executive to a joint announcement with the CEO, who enters at the second prompt, and ends with the two characters sharing the stage. Each row shows one method over the six segments. The case tests whether the original subject is retained while a new character is introduced and the interaction remains coherent; the cast change is the first point at which a temporal memory can lose the initial subject.



0-10s:**An astronomer** is looking through the eyepiece of a massive telescope inside a large, dark observatory dome. The slit in the dome is open to a brilliant, star-filled night sky.
10-20s:**The astronomer** looking through the massive telescope in the dark observatory dome is joined by **a junior research** assistant who brings a star chart and a cup of coffee. The starry sky is breathtaking.
20-30s:**The astronomer and the research assistant** are in the dark observatory dome. The astronomer steps away from the telescope and discusses the star chart with the assistant, pointing to a specific constellation. The stars are incredibly clear.
30-40s:**The astronomer and the research assistant** in the dark observatory dome both look up as a bright meteor streaks across the open slit of the dome. They watch in shared awe. The telescope is a silent giant.
40-50s:**The astronomer in the dark** observatory dome allows **the research assistant** to look through the eyepiece of the massive telescope. The assistant's face fills with wonder as they see a distant galaxy for the first time.
50-60s:**The astronomer and the research assistant** stand together in the dark observatory dome, sharing a quiet moment of discovery and mentorship under the vast, star-filled night sky.

Figure F.2: **Extended comparison on a two-character script with a transient event.** An astronomer is joined by a research assistant in an observatory, where a bright meteor briefly crosses the sky. The case combines a persistent two-character cast, a low-light background, and a short event that must not dominate the memory. Labels mark the failure modes observed in the compared outputs, including attribute shift, quality collapse, and character mismatch.



0-10s:**A glass of red wine** sits on a small table next to a crackling fireplace in a cozy, wood-paneled study. The room is dark except for the warm glow of the fire.
10-20s:**A glass of red wine** sits next to the crackling fireplace in the cozy study. A large, fluffy golden retriever dog pads silently into the room and lies down on the rug in front of the fire.
20-30s:**The glass of red wine and the golden retriever** are in the cozy study. The dog rests its head on its paws, its eyes closing as it enjoys the warmth of the crackling fireplace.
30-40s:**The glass of red wine and the golden retriever** are in the cozy study. A person's hand enters the frame and gently strokes the dog's soft fur. The dog's tail gives a lazy thump-thump on the rug.
40-50s:**The glass of red wine and the golden retriever** are in the cozy study. **The hand picks up the glass of wine from the table.** The dog remains sleeping peacefully by the crackling fireplace.
50-60s:**The person sips the red wine** while continuing to pet the sleeping golden retriever, a perfect picture of quiet contentment in the cozy, fire-lit study.

Figure F.3: **Extended comparison on a low-motion multi-object script.** A glass of wine and a sleeping dog share a dimly lit study, and a hand enters the frame to stroke the dog and lift the glass. The scene deliberately contains little motion but several objects whose states change over time. Labels mark motionless output and spurious additional objects in the compared outputs, which is the failure mode that consistency-oriented metrics alone do not penalize.



0-10s:**A mail carrier** is delivering mail on a quiet, leafy suburban street, placing letters into a curbside mailbox. The sun is shining and the neighborhoo is peaceful.
10-20s:**The mail carrier** delivering mail on the quiet suburban street is suddenly barked at by **an enthusiastic golden retriever puppy** that comes bounding out of a nearby yard, wagging its tail furiously.
20-30s:**The mail carrier and the golden retriever puppy** are on the quiet suburban street. The mail carrier, unafraid, kneels down and greets the puppy with a friendly scratch behind the ears. The puppy licks his face.
30-40s:**The mail carrier** on the quiet suburban street takes a dog biscuit from his mailbag and offers it to **the golden retriever puppy**, which takes it gently. The neighborhood is tranquil.
40-50s:**The mail carrier and the golden retriever puppy** on the quiet suburban street. The puppy, having finished its treat, playfully nudges the mail carrier's hand for more pets. The mail carrier obliges with a smile.
50-60s:**The mail carrier** gives the golden retriever puppy one last pat before continuing on his route down the quiet, leafy suburban street. **The puppy** sits on the lawn, watching him go.

Figure F.4: **Visualizing the core mechanism ablation on a two-entity script.** Rows correspond to the variants of Table 4 and columns to the six prompt segments; labels mark the failure modes observed in the individual outputs, including multiple objects, character mismatch, attribute drift, and a missing character.