GENMEM: LEARNING ULTRA-COMPACT VIDEO MEMORY BY GENERATIVE SUPERVISION

Anonymous authors
Paper under double-blind review

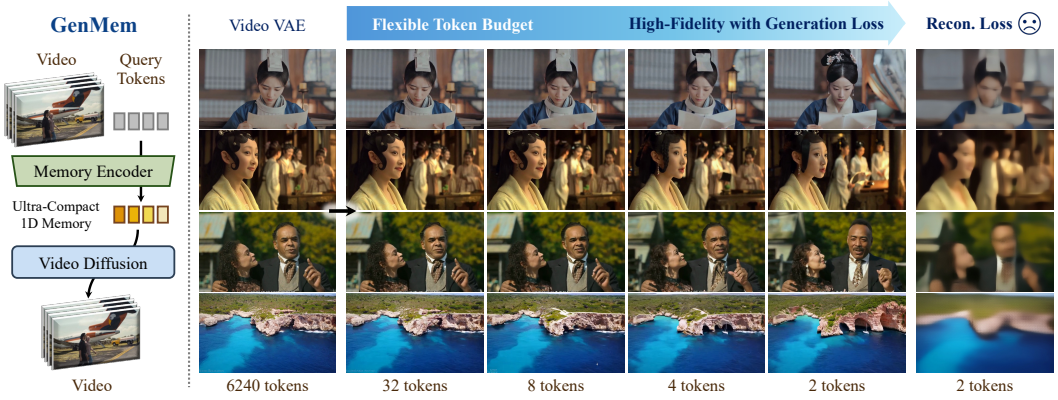


Figure 1: GenMem flexibly compresses the 6,240-token Video VAE latents into an ultra-compact 1D memory of only 32 tokens for video diffusion. Even with only 2 tokens, generative supervision yields high-fidelity outputs, while conventional reconstruction supervision leads to severe blurring.

ABSTRACT

Recent video generation models can produce long videos, but maintaining long-term consistency remains challenging because conditioning on extensive visual history requires prohibitive memory and computation. We introduce **GenMem**, a compact visual memory for long-context video generation. GenMem can compress 4 video frames of 480p resolution into only 32 tokens while retaining high-fidelity visual information. Our memory combines flexible Q-Former-based token compression with a hierarchical representation that prioritizes important semantics, appearance, and dynamics within a small token budget. To train our memory encoder, we find that supervision from the generation loss enables more aggressive compression while better preserving semantically relevant information than using vanilla reconstruction loss. This compact design allows video generators to efficiently condition on several minutes of visual history. Experiments show that GenMem enables substantially more scalable long-context generation while preserving the consistency of scenes and objects observed from distant video history.

1 INTRODUCTION

Modern video generators have achieved remarkable advances in visual quality, motion realism, and temporal coherence (Yang et al., 2025; Kong et al., 2024; Wan et al., 2025; et. al., 2025; OpenAI, 2025; Google DeepMind, 2025; ByteDance Seed, 2026; Kuaishou Technology, 2026). Extending them to long generation, however, requires preserving information from video history over much longer time horizons. Directly conditioning on the full history is computationally expensive, while existing autoregressive approaches typically rely on only a short recent context (Yang et al., 2026; Li et al., 2025; Yuan et al., 2026), causing earlier scenes, objects, and states to be gradually forgotten. This creates a fundamental challenge for long video generation: how can we compress extensive video history into a compact memory that preserves the information needed for future generation?

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.

Existing works have already explored memory compression mechanisms for video generation but remain insufficient. *Recurrent memory* methods (Yu et al., 2025c; Bian et al., 2026) continuously consolidate incoming video history into a bounded evolving state, achieving constant memory cost but repeatedly merging past information rather than preserving earlier observations explicitly, which may gradually dilute fine-grained historical details. *Compression-based methods* (Zhang et al., 2025b; 2026; Zhu et al., 2025) encode video history with substantially fewer tokens, but typically allocate memory capacity using predefined compression rates rather than adapting it to the information content of the video. *Compact visual tokenizers* (Yu et al., 2024; Atanov et al., 2026; Wang et al., 2024a) learn highly compressed or structured latent representations for efficient visual generation, including 1D token sequences and coarse-to-fine hierarchies. However, they are primarily designed as general-purpose representations of visual content rather than memories for long-term conditioning, and do not explicitly optimize what historical information should be retained for future video generation. Therefore, learning a compact and adaptive video memory that preserves generation-relevant information over long video histories remains an open challenge.

In this paper, we try to address this video history compression problem with an ultra-compact memory encoder named GenMem. As illustrated in fig. 1, GenMem compresses 4 video frames of 480p resolution into only 2–32 memory tokens while preserving generation quality. This enables highly compact long-term memory, achieving substantially higher compression than existing approaches. Even at the maximum budget, GenMem uses only 1/4 of the tokens required by prior compression methods such as TinyHistory (Zhang et al., 2026) and VideoFlexTok (Atanov et al., 2026) for the same videos.

In GenMem, we adopt three complementary designs to learn a compact memory. First, we identify query-based resampling as an effective mechanism for long-video memory compression. Inspired by Q-Former (Li et al., 2023), GenMem employs a small set of learnable queries to attend to the video latent space and aggregate spatiotemporal information into compact memory tokens. Unlike convolutional pooling or FramePack Zhang et al. (2025b), which enforce fixed local aggregation and compression patterns, query-based resampling enables each token to adaptively capture information from different regions and allocate representation capacity according to content importance. This flexibility allows GenMem to preserve more meaningful semantic information under an extremely small limited token budget.

Second, GenMem introduces a hierarchical memory structure that enables adaptive token allocation for compact representation. Inspired by hierarchical visual tokenizers (Miwa et al., 2025; Liu et al., 2025b; Bachmann et al., 2025; Atanov et al., 2026), we encourage memory tokens to follow a similar importance hierarchy, where early tokens capture global scene semantics and generation-critical information, while later tokens progressively encode finer details. During training, we randomly drop tail tokens and force retained prefix tokens to generate the whole videos. This hierarchical design encourages the encoder to pack as much information as possible into the earliest prefix tokens, yielding progressively refined representations while pushing the shortest prefixes toward maximal information density. As a result, GenMem learns a compact memory representation that can flexibly adapt its token budget at inference time.

Third, we propose to learn memory compression through the downstream generative model rather than a standalone reconstruction loss. Given a fixed pretrained video generator, GenMem first compresses a video into compact memory tokens, which are then used to condition the generator to reproduce the original video. The resulting flow-matching loss is backpropagated through the frozen generator to train the memory encoder, encouraging the memory to preserve information that is useful for generation. Unlike previous memory encoding methods (Yu et al., 2024; Miwa et al., 2025; Liu et al., 2025b), which optimize the encoder by reconstructing the original video with a decoder, our generative supervision encourages the memory to preserve information that is critical for future generation, such as scene semantics, layout, and motion dynamics. This learned continuous memory is therefore better aligned with the capabilities of video generators.

Finally, we propose a new data augmentation method to help the generator attend to long history. Simply providing long video histories with the video generator does not guarantee that the generator will effectively utilize them. When recent frames already provide sufficient visual cues, the model may ignore distant memory and behave like a short-context image-to-video generator, limiting the benefit of long-history conditioning. To address this issue, we introduce a self-supervised temporal zoom augmentation that explicitly encourages memory retrieval. By digitally zooming into the

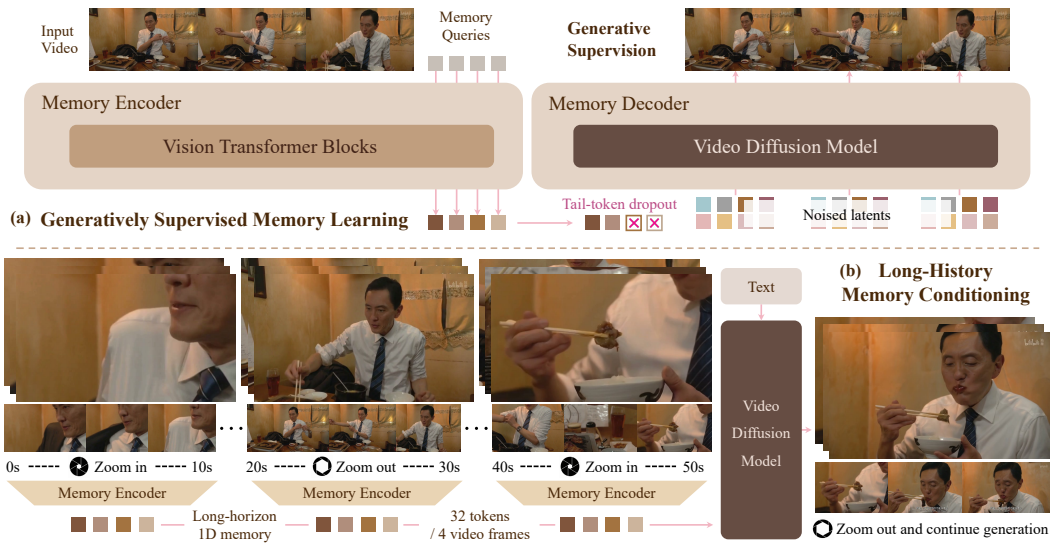


Figure 2: Overall framework of GenMem. (a) *Generatively supervised memory learning.* A query-based encoder compresses video VAE latents into compact memory tokens. Tail-token dropout enforces a flexible memory hierarchy, while the video diffusion model provides generative supervision. (b) *Long-history memory conditioning.* Compact memory enables efficient conditioning on long video histories. Zoom augmentation removes target content from recent frames, encouraging its retrieval from distant memory.

recent frames while preserving the corresponding content only in earlier full-field-of-view frames, the augmentation creates training examples where recent observations become insufficient, and the generator must recover missing context from distant memory.

We evaluate our video compression mechanism in two aspects. First, we evaluate the reconstruction ability to reconstruct all the compressed video memory. In comparison with all existing video tokenizers (Yan et al., 2025; Ye et al., 2026; Atanov et al., 2026), we achieve the best token compression ratio while preserving semantics. Second, we evaluate the ability to inject long video history into a video generator. In comparison with memory compression methods (Zhang et al., 2025b; Zhu et al., 2025), our compressed memory tokens enable the video generator to better maintain the consistency of subjects and scenes only appearing in the distant video history.

2 Related Work

Long-Horizon Video Generation builds on latent video Diffusion Transformers (DiTs) (Peebles & Xie, 2023; Kong et al., 2024; Wan et al., 2025). Autoregressive video diffusion adapts these bidirectional DiTs to causal chunk generation (Yin et al., 2025; Sand.AI et al., 2025) and supports longer interactive rollouts (Liu et al., 2026; Yang et al., 2026; Yuan et al., 2026). Complementary training methods then reduce train–test divergence through sequence-level noise schedules (Chen et al., 2024), few-step distillation (Song et al., 2023; Yin et al., 2024b;a), self-rollout distillation (Huang et al., 2025; Cui et al., 2026; Zhu et al., 2026), and long-context teacher supervision (Chen et al., 2026c). Error banks and recycling further reduce accumulated drift (Guo et al., 2026; Po et al., 2026; Li et al., 2025). Sparse attention (Xi et al., 2025; Sun et al., 2025; Xia et al., 2025; Wu et al., 2026; Zhang et al., 2025c), linear-complexity architectures (Chen et al., 2026a), and pyramidal multi-scale generation (Jin et al., 2025; Liu et al., 2025a) reduce context cost. These advances improve duration, speed, and rollout stability, yet bounded windows still discard distant history, while dense long-context processing remains costly.

Memory Designs for Video Generation preserve context that falls outside the local generation window. Local-retention methods keep a recent window with sink frames and fixed anchors (Henschel et al., 2025; Liu et al., 2026; Yi et al., 2026). Memory-selective methods retrieve past observations by semantic relevance (Ji et al., 2025; Hu et al., 2026b; Ding et al., 2026), spatial overlap (Yu et al., 2025a; Wu et al., 2025a; Zhao et al., 2026), and scene geometry (Hu et al., 2026a; Lee et al., 2026;

Hong et al., 2025). Recurrent methods propagate a bounded learned state across clips (Po et al., 2025; Chen et al., 2026d; Yu et al., 2025c; Chen et al., 2026b; Bian et al., 2026). Context-routing methods select across heterogeneous context groups (Cai et al., 2026; Jia et al., 2025). Compression-based methods retain adaptive keyframes (Zhang et al., 2025a; An et al., 2026) and pack historical frames and features (Zhang et al., 2025b; Gu et al., 2025; Wu et al., 2025b; Zhu et al., 2025; Zhang et al., 2026). Their rigid spatiotemporal patches remain intra-frame redundant. Learned-query 1D tokenizers alleviate this constraint and support more aggressive token reduction (Yu et al., 2024; Bachmann et al., 2025; Atanov et al., 2026; Li et al., 2026; Teng et al., 2026; Zheng et al., 2026). Most remain limited by discrete codes, insufficient token reduction, and GPT-style autoregressive decoding. In contrast, GenMem learns a compact set of continuous embeddings as flexible 1D memory that integrates with pretrained video diffusion backbones, enabling long-range consistency while improving generation quality.

3 METHOD

We design GenMem to compress video history into compact memory tokens for long-horizon video generation. Our method consists of three key components. First, building on a Q-Former-style encoder, we encode each video clip into a compact memory (section 3.1). Second, we train the memory encoder through a frozen pretrained video DiT, using its generation objective to encourage the memory to preserve information most useful for video generation (section 3.2). Finally, we integrate the learned memory into the video diffusion model, enabling efficient conditioning on minute-scale video histories with tractable computation (section 3.3).

3.1 Q-FORMER-BASED MEMORY ENCODING

We introduce the Q-Former-based memory encoder that compresses 4 video frames into a small set of memory tokens, as illustrated in fig. 2(a). We first encode 4 video frames using the pretrained WAN 2.1 VAE (Wan et al., 2025) and flatten its spatiotemporal latent features into visual tokens $\mathbf{Z}$. Given these visual tokens, our goal is to summarize their content using only a small number of learnable memory tokens. For a long video, we will divide it into multiple 4-frame chunks and encode these chunks separately.

Specifically, we introduce an ordered set of learnable query tokens $\mathbf{Q} = \{\mathbf{q}_1, \ldots, \mathbf{q}_M\}$ and concatenate them with the visual tokens as input to a Transformer encoder E_ϕ. The two token sets are jointly processed through stacked self-attention layers, allowing the query tokens to attend to the visual tokens and progressively aggregate information from the visual tokens. In the end, the visual tokens are discarded, and the updated query tokens are retained as the compressed memory $\mathbf{H}$:

$$\mathbf{H} = E_\phi(\mathbf{Z}, \mathbf{Q}),$$

where $\mathbf{H} = \{\mathbf{h}_1, \ldots, \mathbf{h}_M\}$ are the resulting memory tokens and M is the memory token number.

Discussion. Query-based aggregation is particularly suitable for memory compression because the learnable queries can adaptively gather information from the entire visual content, rather than being tied to fixed spatial or temporal locations in previous memory compression methods Zhang et al. (2026); Zhu et al. (2025). Similar designs have been explored in visual tokenizers (Li et al., 2023; Yu et al., 2024; Bachmann et al., 2025; Atanov et al., 2026), which typically quantize the resulting representations into discrete tokens for generative modeling. In contrast, we retain continuous query representations as memory tokens and directly condition a pretrained video generator on them, turning query-based tokenization from a generative representation into a compact memory mechanism.

3.2 GENERATIVE SUPERVISION OF MEMORY COMPRESSION

Our goal is to train the memory encoder to preserve information that is useful for video generation, rather than merely minimizing reconstruction error in the latent or pixel space. To this end, we directly supervise the memory through a pretrained video DiT. Given a video latent $\mathbf{V}$, the memory encoder produces compact memory tokens $\mathbf{H} = E_\phi(\mathbf{V})$. We then use $\mathbf{H}$ as an additional condition for the DiT and ask the generator to reconstruct the same video through the flow-matching loss,

$$\mathcal{L}_{\mathrm{FM}} = \mathbb{E}_{\boldsymbol{\epsilon},t}\left[\left\|G_\theta(\mathbf{V}_t, t; \mathbf{H}) - (\boldsymbol{\epsilon} - \mathbf{V})\right\|_2^2\right], \tag{1}$$

where $\mathbf{V}_t = (1-t)\mathbf{V} + t\boldsymbol{\epsilon}$ is the noised version, $t \in [0, 1]$ is the timestep, and $\epsilon \sim \mathcal{N}(0, 1)$ is a noise sampled from a Gaussian distribution. Because the pretrained DiT already provides a strong

generative prior for recovering texture and fine appearance, the memory does not need to explicitly preserve every low-level detail. Instead, it is encouraged to retain the compact information required by the generator to faithfully recover the video, enabling substantially stronger compression than reconstruction-oriented memory learning.

Hierarchical memory via prefix supervision. To further improve compression efficiency, we organize the memory tokens into an ordered hierarchy through prefix-based training. During each training iteration, we randomly sample a token budget k and discard all memory tokens after the first k, while keeping the same video reconstruction target and flow-matching objective as above. Since the generator must recover the video using only the retained prefix, the encoder is encouraged to pack the most important information into the earliest tokens and make each shorter prefix as self-contained as possible. As k increases, additional tokens progressively refine appearance and fine details. This training strategy therefore pushes the memory toward maximal compactness while naturally supporting flexible token budgets at inference time.

Discussion. A natural alternative is to train the memory encoder together with a separate decoder using a conventional reconstruction objective. However, such supervision encourages the compressed representation to preserve information for pixel- or latent-level reconstruction, which may not align with what a pretrained video generator actually needs. We therefore train the memory directly through the generator's flow-matching objective, allowing the generative model itself to define the information that should be retained. We compare these two training paradigms experimentally in section 4, where direct generative supervision leads to substantially more effective compression at small token budgets.

3.3 History Memory Conditioning

After learning the memory representation, we encode the preceding video history into compact memory tokens and inject them directly into the video DiT through an added *memory-conditioning branch*, as illustrated in fig. 2(b). This branch contains the projections and feed-forward pathways that process the memory tokens, which interact with the current noised video tokens through joint self-attention. To distinguish memory tokens from the current video generation tokens, we extend the 3D RoPE positional encoding by assigning the memory tokens negative temporal coordinates, while keeping the current video tokens at their original non-negative positions. This provides a neat way to incorporate long context without modifying the basic attention structure of the pretrained DiT.

Zoom-based data augmentation. To encourage the generator to actively use long-range memory, we introduce a self-supervised zoom augmentation that removes the shortcut of relying only on recent context. Specifically, we randomly select a spatial region in the video and gradually zoom into this region over the observed frames, so that surrounding scene content and other objects disappear from the recent context while remaining visible in the earlier full-view history. We then zoom back out and use the subsequent wider-view frames as the generation target. To correctly generate the reappearing content outside the zoomed-in region, the model must retrieve the corresponding scene and object information from distant memory. This simple augmentation creates long-range recall supervision from ordinary videos, without requiring specialized scene-revisitation data.

4 Experiments

We conduct our evaluation in three aspects: (1) reconstruction fidelity and token efficiency (section 4.2), (2) the effectiveness of generative supervision across memory budgets (section 4.3), and (3) long-form video continuation and distant-memory utilization (section 4.4).

4.1 Experimental Setup

Implementation details. We conduct experiments with the Wan2.1 1.3B backbone (Wan et al., 2025) at 480×832 resolution and 24 fps. Tail-token dropout samples the memory-token budget $k \in \{2, 4, 8, 16, 32\}$. The memory encoder contains 12 transformer blocks. Flow-matching training uses a flow shift of 3.0 and bell-shaped timestep weighting. More implementation details provided in appendix section D.1.

Datasets. For reconstruction tasks, we first pretrain the memory encoder on 1M LAION-5B images (Schuhmann et al., 2022), followed by training on 500K 5-second Koala-36M (Wang et al.,

Table 1: Reconstruction fidelity and token efficiency at 256×256 on OpenVid-1M (Nan et al., 2025). ElasticTok-C/-D denote continuous/discrete variants; † denotes 128 × 128 reconstruction with upsampling. Best/second-best results are bold/underlined.

Method	# Tokens	# Frames	PSNR↑	SSIM↑	LPIPS↓	FVD↓
ElasticTok-D	4096	4	30.31	<u>0.913</u>	0.133	46.9
ElasticTok-C	2048	4	**30.69**	**0.916**	<u>0.115</u>	<u>43.8</u>
LARP†	1024	16	22.36	0.672	0.296	181.1
AdapTok†	512	4	20.32	0.591	0.361	231.5
InfoTok	≈512	4	26.48	0.820	0.202	104.6
VideoFlexTok	256	4	20.70	0.673	0.258	68.1
GenMem-R	**16**	4	<u>30.33</u>	**0.916**	**0.082**	**9.6**

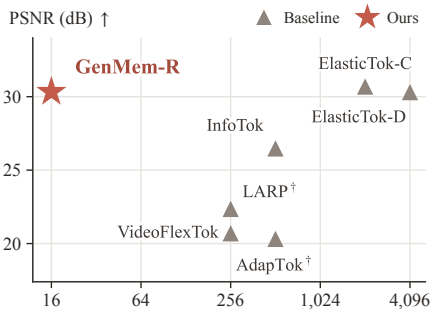


Figure 3: PSNR versus token count on OpenVid-1M. Higher and farther left is better.

2025b) clips. Reconstruction is evaluated on 500 OpenVid-1M (Nan et al., 2025) videos randomly sampled from a diverse, high-quality subset. For long-form video continuation tasks, we train on an internal dataset of 100K high-quality, single-shot cinematic videos ranging from 0.5 to 1 minute in duration and evaluate on 100 videos randomly sampled from a held-out split.

Evaluation. Following the implementation in TokenBench (et. al., 2025), we evaluate reconstruction fidelity using PSNR, SSIM, LPIPS, and FVD. To quantify semantic and motion preservation across memory budgets, we compare reconstructed videos with their inputs using ViCLIP embedding similarity (Wang et al., 2024b) and similarity between RAFT-estimated optical flows (Teed & Deng, 2020), respectively. For long-form generation, we report VBench scores (Huang et al., 2024) for video quality, subject consistency, and background consistency under standard and zoom-augmented video continuation tasks.

4.2 Token-Efficient Video Reconstruction

We first evaluate the reconstruction fidelity of GenMem under compact memory budgets. We denote variants trained with reconstruction and generative supervision as GenMem-R and GenMem-G, respectively. At 256 × 256, table 1 and fig. 3 compare GenMem-R with prior discrete 1D video tokenizers, including ElasticTok (Yan et al., 2025), LARP (Wang et al., 2025a), AdapTok (Li et al., 2026), InfoTok (Ye et al., 2026), and VideoFlexTok (Atanov et al., 2026). Using only 16 tokens budget, GenMem-R outperforms every discrete baseline across PSNR, SSIM, LPIPS, and FVD, despite using 16× fewer tokens than VideoFlexTok and up to 256× fewer than ElasticTok-D. It also uses 128× fewer tokens than ElasticTok-C while achieving comparable PSNR and SSIM and substantially better LPIPS and FVD. Figure 3 further shows that our compact memory supports high-fidelity video reconstruction with very few tokens. At 480 × 832, table 2 compares our methods with the video VAEs of Wan2.1 (Wan et al., 2025), HunyuanVideo-1.5 (Tencent Hunyuan Team, 2025), and LTX-2 (HaCohen et al., 2026). GenMem-R uses 32 tokens for 4 video frames, 6× fewer than LTX-2 (390 tokens over 8 frames) and 195× fewer than Wan2.1 (6,240 tokens over 4 frames), while retaining 30.01 dB PSNR and 8.5 FVD. Together, these results demonstrate that GenMem provides substantially more compact video representations than prior tokenizers and pretrained video VAEs while preserving strong reconstruction fidelity across resolutions.

Table 2: Reconstruction fidelity and token efficiency at 480 × 832 on OpenVid-1M (Nan et al., 2025). Best and second-best results are bolded and underlined.

Method	# Tok.	# F	PSNR↑	SSIM↑	LPIPS↓	FVD↓
Wan2.1	6240	4	**35.76**	**0.959**	**0.073**	**1.5**
HYVideo-1.5	1560	4	<u>35.45</u>	<u>0.951</u>	<u>0.090</u>	<u>2.7</u>
LTX-2	390	8	30.77	0.921	0.118	4.9
GenMem-R	**32**	4	30.01	0.894	0.138	8.5
GenMem-G	**32**	4	26.06	0.844	0.190	14.1

4.3 Memory Flexibility for Video Generation

Flexible memory should preserve generation-critical semantics and motion as its token budget decreases while remaining responsive to text during video generation. We evaluate it in three aspects: (1) reconstruction fidelity at the full budget and semantic and motion preservation at smaller budgets

Table 3: Comparison of reconstruction fidelity and semantic and motion preservation under reconstruction versus generative supervision and across memory-encoder training strategies. Training from scratch with the generation loss best preserves semantics and motion across memory budgets.

Method	Reconstruction (max k)				Semantic ($\times$100) ↑				Motion ($\times$100) ↑			
	PSNR↑	SSIM↑	LPIPS↓	FVD↓	16	8	4	2	16	8	4	2
VideoFlexTok	20.70	0.673	0.258	68.1	75.0	68.2	59.8	48.8	74.8	67.5	59.2	52.0
GenMem-R	**30.01**	**0.894**	**0.138**	**8.5**	87.9	80.6	70.3	55.2	90.3	87.7	84.1	78.9
GenMem-G												
(a) Scratch	26.06	0.844	0.190	14.1	**98.1**	**96.4**	**91.5**	**83.9**	**92.5**	**91.6**	**88.9**	**85.1**
(b) Pretrained	22.83	0.753	0.259	28.7	91.5	84.5	78.4	63.2	87.7	84.6	79.1	37.7
(c) Refiner	26.33	0.856	0.178	16.2	94.7	91.5	86.5	83.0	89.6	87.7	84.9	82.3
(d) Decoder Sup.	25.54	0.841	0.197	16.6	87.6	76.8	72.2	64.6	87.8	82.2	73.8	48.5



Figure 4: Supervision across token budgets. R_k and G_k denote reconstructions from GenMem-R and GenMem-G, respectively, using k tokens. Reconstruction supervision favors full-budget pixel fidelity but blurs at k=2, whereas generative supervision preserves global scene structure and motion as the budget decreases. The comparison spans diverse scenes, motions, and camera dynamics.

under reconstruction versus generative supervision; (2) whether reconstruction pretraining improves memory learning; and (3) whether the memory remains text-responsive for video generation.

Generative supervision enables flexible memory. As shown in table 3 and fig. 4, reconstruction supervision favors full-budget fidelity, achieving 30.01 dB PSNR and 8.5 FVD at k=32, versus 26.06 dB and 14.1 FVD with generative supervision. At the much smaller two-token budget, however, generative supervision retains 83.9 semantic and 85.1 motion similarity, compared with 55.2 and 78.9 under reconstruction supervision, whose outputs become low-fidelity and blurred. Our generatively supervised model also consistently outperforms the prior state-of-the-art flexible video tokenizer VideoFlexTok across all token budgets. These results show that the generation loss is better suited than reconstruction loss for learning a coarse-to-fine memory whose leading tokens retain generation-critical information, enabling more flexible inference-time budgets.

Reconstruction pretraining degrades flexible memory. We further investigate whether reconstruction pretraining or auxiliary reconstruction supervision can improve flexible memory learning. We compare 4 strategies in table 3: (a) *Scratch* trains the encoder directly with the generation loss; (b) *Pretrained* first trains an autoencoder and then finetunes its encoder with the generation loss; (c) *Refiner* freezes the pretrained encoder and trains a transformer refiner; and (d) *Decoder Supervision* retains the reconstruction decoder and adds the reconstruction loss during generative training. The four architectures are illustrated in the appendix (fig. 7). Among them, Scratch achieves the best semantic and motion similarity at every truncated budget and the lowest FVD. All reconstruction-



Figure 5: Qualitative results of text-responsive semantic editing with only two memory tokens. Changing the caption changes the generated subject accordingly while preserving the source scene structure, motion, and camera dynamics.

Table 4: Quantitative comparison on standard video continuation and distant-memory utilization under zoom augmentation. The latter requires recovering content visible in distant history but absent from the local context. We report VBench scores for video quality, subject consistency, and background consistency in both settings. Bold and underline denote the best and second-best scores.

Method	Backbone	Standard continuation			Zoom-augmented continuation		
		Quality↑	Subj. Cons.↑	Bkg. Cons.↑	Quality↑	Subj. Cons.↑	Bkg. Cons.↑
Local memory							
SVI	14B	**81.66**	**95.48**	**95.86**	77.32	79.42	<u>87.84</u>
Self Forcing	1.3B	79.34	90.30	92.57	76.48	76.45	85.50
Selective memory							
LongLive	1.3B	80.17	91.79	93.52	78.03	78.62	86.50
MemFlow	1.3B	80.17	91.58	93.32	78.17	79.12	86.76
Evolving memory							
Echo-Infinity	1.3B	<u>81.54</u>	90.93	93.37	<u>78.56</u>	78.99	86.72
Compressed memory							
FramePack	13B	80.64	91.58	93.92	78.50	77.77	85.88
MAG	1.3B	79.86	92.45	93.96	77.16	<u>79.55</u>	86.97
GenMem (ours)	1.3B	80.60	<u>93.77</u>	<u>94.66</u>	**79.04**	**84.44**	**90.16**

pretrained alternatives perform worse, showing that reconstruction-oriented representations transfer poorly to the generation objective. Directly training the memory encoder from scratch also avoids the costly reconstruction training, so we use this strategy in all subsequent experiments.

Memory remains text-responsive for semantic editing. We evaluate text responsiveness through semantic editing. As shown in fig. 5, changing the caption while conditioning on only the first two memory tokens changes the generated subject accordingly, while preserving the source scene structure, motion, and camera dynamics. This demonstrates that the leading memory tokens provide a semantic and dynamic scaffold without compromising text adherence.

4.4 Long Video Continuation with GenMem

Our compact memory enables long-form video continuation by efficiently conditioning on the full history over much longer time horizons, improving consistency and recovering content absent from recent frames. We evaluate it on two complementary tasks: standard video continuation against prior long-video generation methods and zoom-augmented continuation, which requires retrieving information available only in distant history.

Standard Video Continuation. We compare GenMem with prior long-video generation methods that use local memory (Li et al., 2025; Huang et al., 2025), selective memory (Yang et al., 2026; Ji et al., 2025), evolving memory (Bian et al., 2026), or compressed memory (Zhang et al., 2025b; Zhu et al., 2025), as grouped in table 4. Given the preceding long-video history, the task is to generate the subsequent 5-second video segment. The autoregressive baselines process the history through zero-timestep prediction to initialize their KV caches, subject to each method's default context window. In contrast, GenMem encodes the full preceding history in non-overlapping 5-second chunks and concatenates the resulting memory-token sequences. As shown in table 4, GenMem achieves the highest subject consistency (93.77) and background consistency (94.66) among the evaluated 1.3B models while maintaining competitive video quality (80.60). The qualitative comparison in fig. 6(a) likewise shows stronger preservation of subject identity and scene structure. FramePack and SVI

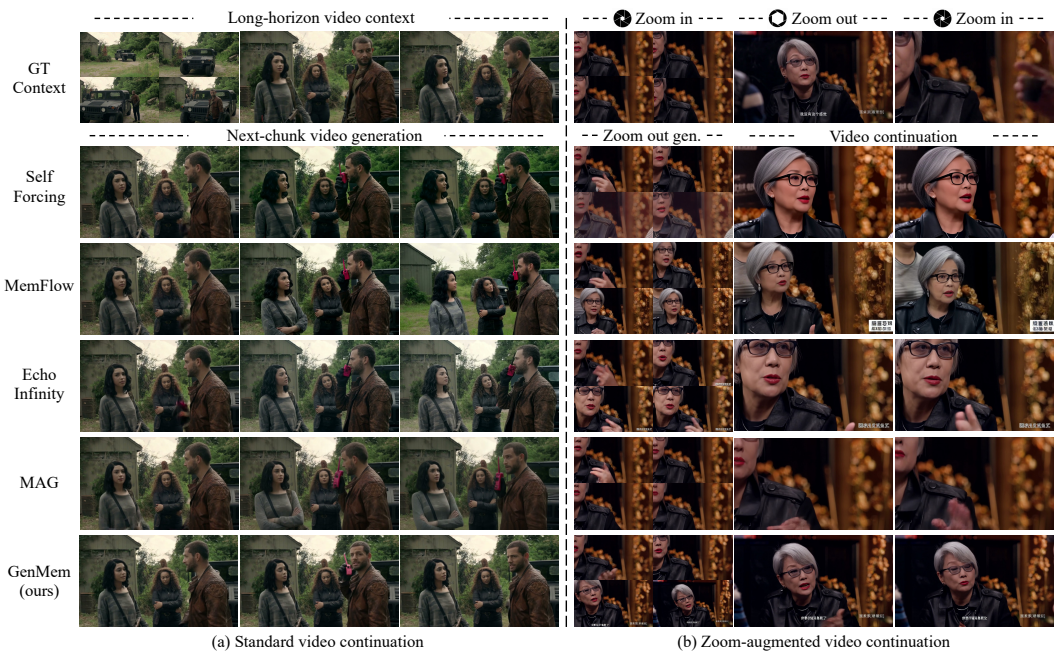


Figure 6: Qualitative comparisons on (a) standard and (b) zoom-augmented video continuation. GenMem better preserves subject identity and scene structure, particularly when target-relevant content is absent from the recent context.

use much larger backbones and achieve higher video quality, whereas GenMem provides stronger subject and scene consistency without increasing the generator size. Evaluation at longer horizons and with larger backbones is left for future work.

Zoom-Augmented Video Continuation. Standard continuation fails to reveal whether a model uses the full video history because recent frames often suffice for a smooth transition. However, maintaining long-term consistency requires recovering subjects and scenes visible only in earlier history. We therefore construct a self-supervised zoom-augmented task from ordinary videos by randomly selecting a fixed $2\times$ crop centered at the image center or one of the four corners and resizing it to the original resolution. The zoomed-in view is maintained over a long, randomly sampled temporal span subject to a minimum threshold of 5 seconds before the prediction target, ensuring that target-relevant content is absent from nearby frames but available in the earlier full-view history and thereby forcing the model to use the full historical context. On this task, GenMem outperforms every baseline, including models with much larger backbones, on all three VBench metrics: 79.04 video quality, 84.44 subject consistency, and 90.16 background consistency (table 4). As shown in fig. 6(b) and additional qualitative results in the appendix, competing methods often lose subject identity or scene layout, whereas GenMem recovers both. These results demonstrate that our compact memory representation enables effective conditioning on much longer video histories, recovering missing subject and scene information to better preserve long-range consistency.

5 Conclusion

We propose GenMem, an ultra-compact video memory representation that encodes 4 video frames into only 32 tokens. Generative supervision enables flexible memory budgets that preserve reconstruction fidelity, generation-critical semantics and motion, and text responsiveness. Our experiments demonstrate that our compact memory representation enables video generation conditioned on substantially longer histories, improving long-range subject and scene consistency.

Limitations and future work. Although GenMem achieves strong long-range consistency with a 1.3B backbone, scaling the generator may further improve fine-grained detail and text rendering. Extending the current bidirectional formulation to autoregressive training and generation could enable streaming long-video synthesis. Beyond video generation, applying the compact memory as an efficient long-horizon state for action-conditioned world models is another promising direction.

AI USE STATEMENT

Generative AI tools were used only to draft and edit portions of the paper for clarity, grammar, organization, and formatting. They were not used to develop theoretical or conceptual frameworks, formulate mathematical claims, propose or refine research hypotheses, design the methodology or experiments, implement the method, generate synthetic datasets, clean or reformat datasets, support qualitative or thematic data analysis, interpret the results, or assist with translation. The use of generative AI for mathematical proofs or critical proof ingredients is not applicable to this work. The authors reviewed every AI-assisted edit, verified the technical content and citations, and take full responsibility for the final content of the paper, including all text, claims, and artifacts produced with the assistance of generative AI.

ETHICS STATEMENT

This work introduces a generator-supervised compact visual memory that enables video models to condition efficiently on long histories while preserving generation-critical semantics and motion. We curate training videos from geographically and culturally diverse settings, spanning a broad range of subjects, scenes, activities, and visual styles, while recognizing that the model may still inherit biases from its training data and pretrained components. Generated content could be misused for impersonation or misinformation, and long-history conditioning may reproduce identifiable details present in the conditioning video. We present the method for research purposes and encourage downstream users to use appropriately licensed data, obtain necessary consent, protect personal information, disclose synthetic content, and adopt safeguards against harmful use.

REPRODUCIBILITY STATEMENT

The method, training objectives, and evaluation protocol are described in sections 3 and 4. The appendix provides the complete generative and reconstruction objectives (section B), architectural and attention details (sections D.1 and D.2), and the training curriculum (section D.3). Upon acceptance, we will release all code, training configurations, and the data-processing pipeline needed to reproduce our experiments.

REFERENCES

Zhaochong An, Menglin Jia, Haonan Qiu, Zijian Zhou, Xiaoke Huang, Zhiheng Liu, Weiming Ren, Kumara Kahatapitiya, Ding Liu, Sen He, Chenyang Zhang, Tao Xiang, Fanny Yang, Serge Belongie, and Tian Xie. OneStory: Coherent multi-shot video generation with adaptive memory. In *CVPR*, pp. 16173–16184, 2026.

Andrei Atanov, Jesse Allardice, Roman Bachmann, Oğuzhan Fatih Kar, Peter Fu, David Griffiths, Devon Hjelm, Afshin Dehghan, and Amir Zamir. VideoFlexTok: Flexible-length coarse-to-fine video tokenization. In *ICML*, 2026.

Roman Bachmann, Jesse Allardice, David Mizrahi, Enrico Fini, Oğuzhan Fatih Kar, Elmira Amirloo, Alaaeldin El-Nouby, Amir Zamir, and Afshin Dehghan. Flextok: Resampling images into 1d token sequences of flexible length. In *Forty-second International Conference on Machine Learning*, 2025.

Randall Balestriero and Yann LeCun. LeJEPA: Provable and scalable self-supervised learning without the heuristics. *arXiv preprint arXiv:2511.08544*, 2025.

Yuxuan Bian, Zeyue Xue, Songchun Zhang, Shiyi Zhang, Weiyang Jin, Yaowei Li, Junhao Zhuang, Haoran Li, Jie Huang, Haoyang Huang, Nan Duan, and Qiang Xu. Echo-infinity: Learning evolving memory for real-time infinite video generation. *arXiv*, 2026.

Black Forest Labs. FLUX.1. `https://github.com/black-forest-labs/flux`, 2024.

ByteDance Seed. Seedance 2.0. `https://seed.bytedance.com/seedance2_0`, 2026.

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. In *ICLR*, 2026.

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

Junsong Chen, Yuyang Zhao, Jincheng Yu, Ruihang Chu, Junyu Chen, Shuai Yang, Xianbang Wang, Yicheng Pan, Daquan Zhou, Huan Ling, Haozhe Liu, Hongwei Yi, Hao Zhang, Muyang Li, Yukang Chen, Han Cai, Sanja Fidler, Ping Luo, Song Han, and Enze Xie. SANA-video: Efficient video generation with block linear diffusion transformer. In *ICLR*, 2026a.

Kaijin Chen, Dingkang Liang, Xin Zhou, Yikang Ding, Xiaoqiang Liu, Pengfei Wan, and Xiang Bai. Out of sight but not out of mind: Hybrid memory for dynamic video world models. *arXiv*, 2026b.

Shuo Chen, Cong Wei, Sun Sun, Tiancheng Shen, Ping Nie, Kai Zou, Ge Zhang, Ming-Hsuan Yang, and Wenhu Chen. Context forcing: Consistent autoregressive video generation with long context. In *ICML*, 2026c.

Taiye Chen, Zihan Ding, Anjian Li, Christina Zhang, Zeqi Xiao, Yisen Wang, and Chi Jin. Recurrent autoregressive diffusion: Global memory meets local attention. In *ECCV*, 2026d.

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

Yixuan Ding, Jiahao Kong, Wei Huang, Ruijie Quan, and Yi Yang. LayerRecall: A state-conditioned memory router for long-horizon consistency in video generation. *arXiv*, 2026.

Juechu Dong, Boyuan Feng, Driss Guessous, Yanbo Liang, and Horace He. Flex attention: A programming model for generating optimized attention kernels. *ArXiv*, abs/2412.05496, 2024. URL `https://api.semanticscholar.org/CorpusID:274598006`.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

NVIDIA et. al. Cosmos world foundation model platform for physical ai. *arXiv*, 2025.

Google DeepMind. Veo 3.1. `https://deepmind.google/models/veo`, 2025.

Yuchao Gu, Weijia Mao, and Mike Zheng Shou. Long-context autoregressive video modeling with next-frame prediction. *arXiv*, 2025.

Yuwei Guo, Ceyuan Yang, Hao He, Yang Zhao, Meng Wei, Zhenheng Yang, Weilin Huang, and Dahua Lin. End-to-end training for autoregressive video diffusion via self-resampling. In *ECCV*, 2026.

Yoav HaCohen, Benny Brazowski, Nisan Chiprut, Yaki Bitterman, et al. Ltx-2: Efficient joint audio-visual foundation model. *arXiv*, 2026.

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 *CVPR*, pp. 2568–2577, 2025.

Yicong Hong, Yiqun Mei, Chongjian Ge, Yiran Xu, Yang Zhou, Sai Bi, Yannick Hold-Geoffroy, Mike Roberts, Matthew Fisher, Eli Shechtman, Kalyan Sunkavalli, Feng Liu, Zhengqi Li, and Hao Tan. RELIC: Interactive video world model with long-horizon memory. *arXiv*, 2025.

JiaKui Hu, Jialun Liu, Liying Yang, Xinliang Zhang, Kaiwen Li, Shuang Zeng, Yuanwei Li, Haibin Huang, Chi Zhang, and Yanye Lu. Geometry-as-context: Modulating explicit 3d in scene-consistent video generation to geometry context. In *CVPR*, pp. 4258–4268, 2026a.

Qixin Hu, Shuai Yang, Wei Huang, Song Han, and Yukang Chen. LongLive-RAG: A general retrieval-augmented framework for long video generation. *arXiv*, 2026b.

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. 2025.

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 *CVPR*, 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. *arXiv*, 2025.

Weinan Jia, Yuning Lu, Mengqi Huang, Hualiang Wang, Binyuan Huang, Nan Chen, Mu Liu, Jidong Jiang, and Zhendong Mao. Moga: Mixture-of-groups attention for end-to-end long video generation. *arXiv*, 2025.

Yang Jin, Zhicheng Sun, Ningyuan Li, Kun Xu, Kun Xu, Hao Jiang, Nan Zhuang, Quzhe Huang, Yang Song, Yadong Mu, and Zhouchen Lin. Pyramidal flow matching for efficient video generative modeling. In *ICLR*, 2025.

Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. Hunyuanvideo: A systematic framework for large video generative models. *arXiv*, 2024.

Kuaishou Technology. Kling 3.0. `https://kling.ai`, 2026.

JoungBin Lee, Jaewoo Jung, Jisang Han, Takuya Narihira, Kazumi Fukuda, Junyoung Seo, Sunghwan Hong, Yuki Mitsufuji, and Seungryong Kim. 3d scene prompting for scene-consistent camera-controllable video generation. In *ICLR*, 2026.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven C. H. Hoi. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International Conference on Machine Learning*, pp. 19730–19742, 2023.

Wuyang Li, Wentao Pan, Po-Chien Luan, Yang Gao, and Alexandre Alahi. Stable video infinity: Infinite-length video generation with error recycling. *arXiv*, 2025.

Yan Li, Changyao Tian, Renqiu Xia, Ning Liao, Weiwei Guo, Hongsheng Li, Jifeng Dai, Hao Li, and Xue Yang. AdapTok: Learning adaptive and temporally causal video tokenization in a 1d latent space. In *CVPR*, pp. 16163–16172, 2026.

Jinlai Liu, Jian Han, Bin Yan, Hui Wu, Fengda Zhu, Xing Wang, Yi Jiang, Bingyue Peng, and Zehuan Yuan. Infinitystar: Unified spacetime autoregressive modeling for visual generation. 2025a.

Kunhao Liu, Wenbo Hu, Jiale Xu, Ying Shan, and Shijian Lu. Rolling forcing: Autoregressive long video diffusion in real time. In *ICLR*, 2026.

Yiheng Liu, Liao Qu, Huichao Zhang, Xu Wang, Yi Jiang, Yiming Gao, Hu Ye, Xian Li, Shuai Wang, Daniel K. Du, Shu Cheng, Zehuan Yuan, and Xinglong Wu. Detailflow: 1d coarse-to-fine autoregressive image generation via next-detail prediction. *arXiv*, 2025b.

Keita Miwa, Kento Sasaki, Hidehisa Arai, Tsubasa Takahashi, and Yu Yamaguchi. One-d-piece: Image tokenizer meets quality-controllable compression, 2025. URL https://arxiv.org/abs/2501.10064.

Kepan Nan, Rui Xie, Penghao Zhou, Tiehan Fan, Zhenheng Yang, Zhijie Chen, Xiang Li, Jian Yang, and Ying Tai. Openvid-1m: A large-scale high-quality dataset for text-to-video generation. In *ICLR*, 2025.

OpenAI. Sora 2. https://openai.com/index/sora-2/, 2025.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4195–4205, 2023.

Ryan Po, Yotam Nitzan, Richard Zhang, Berlin Chen, Tri Dao, Eli Shechtman, Gordon Wetzstein, and Xun Huang. Long-context state-space video world models. In *ICCV*, pp. 8733–8744, 2025.

Ryan Po, Eric Ryan Chan, Changan Chen, and Gordon Wetzstein. BAgger: Backwards aggregation for mitigating drift in autoregressive video diffusion models. In *CVPR*, pp. 43727–43739, 2026.

Sand.AI, Hansi Teng, Hongyu Jia, Lei Sun, Lingzhi Li, Maolin Li, Mingqiu Tang, Shuai Han, Tianning Zhang, W. Q. Zhang, Weifeng Luo, Xiaoyang Kang, Yuchen Sun, Yue Cao, Yunpeng Huang, Yutong Lin, Yuxin Fang, Zewei Tao, Zheng Zhang, Zhongshu Wang, Zixun Liu, Dai Shi, Guoli Su, Hanwen Sun, Hong Pan, Jie Wang, Jiexin Sheng, Min Cui, Min Hu, Ming Yan, Shucheng Yin, Siran Zhang, Tingting Liu, Xianping Yin, Xiaoyu Yang, Xin Song, Xuan Hu, Yankai Zhang, and Yuqiao Li. MAGI-1: Autoregressive video generation at scale. *arXiv*, 2025.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. LAION-5B: An open large-scale dataset for training next generation image-text models. In *Advances in Neural Information Processing Systems*, 2022.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *ICML*, 2023.

Wenhao Sun, Rong-Cheng Tu, Yifu Ding, Zhao Jin, Jingyi Liao, Shunyu Liu, and Dacheng Tao. VORTA: Efficient video diffusion via routing sparse attention. *arXiv*, 2025.

Zachary Teed and Jia Deng. RAFT: Recurrent all-pairs field transforms for optical flow. In *ECCV*, pp. 402–419, 2020.

Tencent Hunyuan Team. Hunyuanvideo 1.5 technical report. *arXiv*, 2025.

Yao Teng, Minxuan Lin, Xian Liu, Shuai Wang, Xiao Yang, and Xihui Liu. Adaptive 1d video diffusion autoencoder. *arxiv*, 2026.

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

Hanyu Wang, Saksham Suri, Yixuan Ren, Hao Chen, and Abhinav Shrivastava. Larp: Tokenizing videos with a learned autoregressive generative prior. In *ICLR*, 2025a.

Qiuheng Wang, Yukai Shi, Jiarong Ou, Rui Chen, Ke Lin, Jiahao Wang, Boyuan Jiang, Haotian Yang, Mingwu Zheng, Xin Tao, Fei Yang, Pengfei Wan, and Di Zhang. Koala-36M: A large-scale video dataset improving consistency between fine-grained conditions and video content. In *CVPR*, pp. 8428–8437, 2025b.

Xinlong Wang, Xiaosong Zhang, Zhengxiong Luo, Quan Sun, Yufeng Cui, Jinsheng Wang, Fan Zhang, Yueze Wang, Zhen Li, Qiying Yu, et al. Emu3: Next-token prediction is all you need. *arXiv*, 2024a.

Yi Wang, Yinan He, Yizhuo Li, Kunchang Li, Jiashuo Yu, Xin Ma, Xinhao Li, Guo Chen, Xinyuan Chen, Yaohui Wang, Conghui He, Ping Luo, Ziwei Liu, Yali Wang, Limin Wang, and Yu Qiao. InternVid: A large-scale video-text dataset for multimodal understanding and generation. In *ICLR*, 2024b.

Jianzong Wu, Liang Hou, Haotian Yang, Xin Tao, Ye Tian, Pengfei Wan, Di Zhang, and Yunhai Tong. VMoBA: Mixture-of-block attention for video diffusion models. In *ICLR*, 2026.

Tong Wu, Shuai Yang, Ryan Po, Yinghao Xu, Ziwei Liu, Dahua Lin, and Gordon Wetzstein. Video world models with long-term spatial memory. In *NeurIPS*, volume 38, pp. 49371–49393, 2025a.

Xiaofei Wu, Guozhen Zhang, Zhiyong Xu, Yuan Zhou, Qinglin Lu, and Xuming He. Pack and force your memory: Long-form and consistent video generation. *arXiv*, 2025b.

Haocheng Xi, Shuo Yang, Yilong Zhao, Chenfeng Xu, Muyang Li, Xiuyu Li, Yujun Lin, Han Cai, Jintao Zhang, Dacheng Li, Jianfei Chen, Ion Stoica, Kurt Keutzer, and Song Han. Sparse videogen: Accelerating video diffusion transformers with spatial-temporal sparsity. In *ICML*, 2025.

Yifei Xia, Suhan Ling, Fangcheng Fu, Yujie Wang, Huixia Li, Xuefeng Xiao, and Bin Cui. Training-free and adaptive sparse attention for efficient long video generation. In *ICCV*, pp. 15982–15993, 2025.

Wilson Yan, Matei Zaharia, Volodymyr Mnih, Pieter Abbeel, Aleksandra Faust, and Hao Liu. Elastictok: Adaptive tokenization for image and video. In *ICLR*, 2025.

Shuai Yang, Wei Huang, Ruihang Chu, Yicheng Xiao, Yuyang Zhao, Xianbang Wang, Muyang Li, Enze Xie, Yingcong Chen, Yao Lu, et al. Longlive: Real-time interactive long video generation. In *ICLR*, 2026.

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. In *ICLR*, 2025.

Haotian Ye, Qiyuan He, Jiaqi Han, Puheng Li, Jiaojiao Fan, Zekun Hao, Fitsum Reda, Yogesh Balaji, Huayu Chen, Sheng Liu, Angela Yao, James Zou, Stefano Ermon, Haoxiang Wang, and Ming-Yu Liu. Infotok: Adaptive discrete video tokenizer via information-theoretic compression. In *ICLR*, 2026.

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. In *ICML*, 2026.

Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and William T Freeman. Improved distribution matching distillation for fast image synthesis. In *NeurIPS*, 2024a.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Frédo Durand, William T. Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *CVPR*, pp. 6613–6623, 2024b.

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 *CVPR*, 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 *SIGGRAPH Asia*, pp. 19:1–19:11, 2025a.

Qihang Yu, Mark Weber, Xueqing Deng, Xiaohui Shen, Daniel Cremers, and Liang-Chieh Chen. An image is worth 32 tokens for reconstruction and generation. *Advances in Neural Information Processing Systems*, 37:128940–128966, 2024.

Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. In *ICLR*, 2025b.

Yifei Yu, Xiaoshan Wu, Xinting Hu, Tao Hu, Yangtian Sun, Xiaoyang Lyu, Bo Wang, Lin Ma, Yuewen Ma, Zhongrui Wang, and Xiaojuan Qi. VideoSSM: Autoregressive long video generation with hybrid state-space memory. *arXiv*, 2025c.

Shenghai Yuan, Yuanyang Yin, Zongjian Li, Xinwei Huang, Xiao Yang, and Li Yuan. Helios: Real real-time long video generation model. *arXiv*, 2026.

Kaiwen Zhang, Liming Jiang, Angtian Wang, Jacob Zhiyuan Fang, Tiancheng Zhi, Qing Yan, Hao Kang, Xin Lu, and Xingang Pan. StoryMem: Multi-shot long video storytelling with memory. *arXiv*, 2025a.

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*, 2025b.

Lvmin Zhang, Shengqu Cai, Muyang Li, Chong Zeng, Beijia Lu, Anyi Rao, Song Han, Gordon Wetzstein, and Maneesh Agrawala. TinyHistory: Lightweight video history embeddings via two-stage context learning. In *ECCV*, 2026. URL `https://eccv.ecva.net/virtual/2026/poster/4003`.

Peiyuan Zhang, Yongqi Chen, Haofeng Huang, Will Lin, Zhengzhong Liu, Ion Stoica, Eric P. Xing, and Hao Zhang. Faster video diffusion with trainable sparse attention. 2025c.

Jinjing Zhao, Fangyun Wei, Zhening Liu, Hongyang Zhang, Chang Xu, and Yan Lu. Spatia: Video generation with updatable spatial memory. In *CVPR*, pp. 4245–4257, 2026.

Zihan Zheng, Zhaoyang Jia, Naifu Xue, Jiahao Li, Bin Li, Zongyu Guo, Xiaoyi Zhang, Zhenghao Chen, Houqiang Li, and Yan Lu. Generative video compression with one-dimensional latent representation. In *CVPR*, 2026.

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

Tianrui Zhu, Shiyi Zhang, Zhirui Sun, Jingqi Tian, and Yansong Tang. Memorize-and-generate: Towards long-term consistency in real-time video generation. *arXiv*, 2025.

APPENDIX

A Additional Qualitative Results **16**
A.1 Video Reconstruction 16
A.2 Text-Responsive Subject and Background Editing 16
A.3 Standard Video Continuation 17
A.4 Zoom-Augmented Video Continuation 17
B Training Objectives **17**
B.1 Generative Supervision 17
B.2 Reconstruction Supervision 18
C Memory-Learning Strategies **19**
D Implementation Details **20**
D.1 Network Architecture 20
D.2 Causal Window Attention 20
D.3 Training Curriculum 23

A Additional Qualitative Results

The additional qualitative results cover three evaluation tasks. Section A.1 compares video reconstruction across memory-token budgets under reconstruction and generative supervision. Section A.2 presents text-responsive subject and background editing, showing how the generated content changes with the prompt while retaining source structure and motion. Long-video continuation is examined in two settings: section A.3 compares methods under standard history conditioning, whereas section A.4 removes target-relevant content from recent frames to test retrieval from distant history.

A.1 Video Reconstruction

Figures 10 to 13 provide side-by-side reconstructions from VideoFlexTok (Atanov et al., 2026), GenMem-R, and GenMem-G across token budgets. The examples test object geometry during a camera move around a guitar (fig. 10), a phone-centered hand gesture (fig. 11), a graduate adjusting her cap beside a van (fig. 12), and a fine-grained hand-to-mouth action (fig. 13). Together, they span object- and human-centric content, indoor and outdoor scenes, articulated motion, and viewpoint change.

Rows correspond to token budgets k per 4 video frames, while columns show frames sampled uniformly over time. VideoFlexTok retains the coarse scene at $k{=}256$ and $k{=}32$, but its outputs can alter subject identity, color, object geometry, or hand structure; at $k{=}2$, the reconstructions often drift substantially from the input. GenMem-R, trained with reconstruction supervision, is the most faithful at its full budget but becomes blurred when reduced to two tokens. GenMem-G instead preserves the defining subject, scene layout, and action at $k{=}2$, with larger budgets primarily restoring appearance and texture. The same trend across all four videos supports the coarse-to-fine organization induced by generative supervision: leading tokens retain generation-critical semantics and motion, while later tokens add visual detail.

A.2 Text-Responsive Subject and Background Editing

Figures 14 and 15 examine whether memory and text provide complementary control at the extreme budget $k{=}2$. We encode each source video once, keep its two memory tokens fixed, and vary only the caption. The subject-editing examples in fig. 14 change costume, age, or appearance while retaining the source pose, action, scene geometry, and camera trajectory. The background-editing examples in fig. 15 replace the original landmark with the Eiffel Tower, Golden Gate Bridge, Colosseum, or Temple of Heaven while preserving the speaker's identity, framing, and motion. These results show that the compact memory anchors source-specific structure and dynamics without preventing the generator from following new text semantics.

A.3 Standard Video Continuation

Figures 16 to 18 compare standard video continuation from the same long ground-truth history. The baselines represent three approaches to historical conditioning: local context (Li et al., 2025; Huang et al., 2025), selective or evolving memory (Yang et al., 2026; Ji et al., 2025; Bian et al., 2026), and compressed memory (Zhang et al., 2025b; Zhu et al., 2025). In each figure, the first row visualizes the conditioning history and the remaining rows show the corresponding continuations.

The examples cover a close-up dialogue, a moving subject in a bright indoor space, and a two-person kitchen interaction. Although the competing methods produce plausible individual frames, their continuations exhibit changes in facial identity, pose, participant composition, or scene layout. GenMem more consistently retains the defining subjects, actions, and environments across the prediction. These examples suggest that compressing the complete history provides a more stable conditioning signal than relying only on a bounded or selectively retained context.

A.4 Zoom-Augmented Video Continuation

Figures 19 to 21 test whether a method can retrieve information from distant history rather than relying on recent frames. Zoom augmentation restricts the recent context to a partial view, so subjects and scene regions needed for the continuation remain visible only in the earlier full-view history. The first row shows this augmented conditioning history; the remaining rows compare local-context (Li et al., 2025; Huang et al., 2025), selective and evolving memory (Yang et al., 2026; Ji et al., 2025; Bian et al., 2026), and compressed-memory methods (Zhang et al., 2025b; Zhu et al., 2025) at matched continuation times.

The three examples show a man speaking on a phone, a two-character interaction in an outdoor encampment, and a seated interview. Competing methods change the subject's identity, omit the second character, or generate a layout inconsistent with the earlier full view, suggesting that they do not reliably recover the information removed from recent context. GenMem better restores the missing subjects and surrounding environments, providing qualitative evidence that its compressed memory remains informative beyond the local context window.

B Training Objectives

This section specifies the two objectives compared in the main paper. Generative supervision optimizes the memory through the video DiT using $\mathcal{L}_{\text{gen}}$ (section B.1), whereas reconstruction supervision trains a separate Vision Transformer (ViT)-based reconstruction decoder (Dosovitskiy et al., 2021) using $\mathcal{L}_{\text{recon}}$ (section B.2). Both objectives use the same prefix sampling and isotropic latent regularization, so their principal difference is the signal that determines what the memory must retain.

B.1 Generative Supervision

Generative supervision combines a task-level generation objective with representation alignment and distributional regularization. Flow matching trains the memory to retain information that the pretrained video generator can use directly; the text, global, and spatial alignment terms organize that information at complementary levels; and isotropic regularization keeps the continuous memory space well conditioned.

Flow-matching supervision. Under generative supervision, the memory encoder E_ϕ is trained through the frozen generator G_θ. The primary term is the flow-matching loss $\mathcal{L}_{\text{FM}}$ defined in the main paper, which, given a sampled memory prefix, guides the DiT to predict the clean-video velocity and thereby recover the original video. We supplement it with three cosine similarity losses, following REPA (Yu et al., 2025b), which align the memory with frozen text embeddings and video diffusion features.

The three alignment terms play distinct, complementary roles.

Text alignment. A learned projector maps the first memory token to the text embedding space. $\mathcal{L}_{\text{text}}$ minimizes its cosine distance to the text embedding of the caption, averaged across these

groups. The text encoder remains frozen throughout training. This encourages the leading token to summarize the subject, action, and scene, making even the shortest memory prefix semantically useful.

Global feature alignment. For every supported prefix $k \in \{2, 4, 8, 16, 32\}$, we average memory tokens 2 through k and map the result through a second projector. $\mathcal{L}_{\text{global}}$ measures its cosine distance to the stop-gradient, spatially pooled final-layer DiT feature, averaged across frames and prefix budgets. Applying the loss at every budget prevents useful information from appearing only near the end of the full sequence and encourages later tokens to refine what earlier tokens already preserve.

Spatial feature alignment. Before query pooling, a third projector maps each spatial encoder feature to the teacher feature space. $\mathcal{L}_{\text{spatial}}$ computes the cosine distance to the stop-gradient final-layer DiT feature at the corresponding spatial location and averages it over frames and positions. This term preserves location-specific structure that would be lost by spatial averaging alone.

Isotropic regularization. We apply sketched isotropic Gaussian regularization (SIGReg) (Balestriero & LeCun, 2025) to the complete, untruncated memory. Its purpose is to keep the continuous token space well scaled and prevent different dimensions or tokens from collapsing onto highly correlated representations. At each optimization step, SIGReg samples 1,024 random unit directions and projects the memory tokens onto each direction. It then compares the distribution of every one-dimensional projection with a standard Gaussian, using 17 uniformly spaced evaluation points over $[0, 3]$. The projected statistics are averaged across distributed workers, and the same step-seeded directions are used on every worker so that the regularizer is globally consistent. This provides a distribution-level constraint without constructing pairwise negatives.

Generative training objective. The complete objective is

$$\mathcal{L}_{\text{gen}} = \mathcal{L}_{\text{FM}} + \lambda_{\text{text}}\mathcal{L}_{\text{text}} + \lambda_{\text{global}}\mathcal{L}_{\text{global}} + \lambda_{\text{spatial}}\mathcal{L}_{\text{spatial}} + \lambda_{\text{iso}}\mathcal{R}_{\text{iso}}. \tag{2}$$

$\mathcal{L}_{\text{FM}}$ determines whether the retained information is sufficient for generation, the three alignment terms give that information a semantic-to-spatial organization compatible with the DiT, and $\mathcal{R}_{\text{iso}}$ prevents the resulting continuous representation from degenerating.

The three projection heads do not share parameters. Each consists of LayerNorm followed by a two-layer MLP with SiLU activation; its input, hidden, and output widths are 1536. Cosine distances are evaluated in FP32. The alignment losses are disabled for classifier-free negative samples because the negative caption and video history are intentionally unmatched; flow matching and regularization remain active. We weight $\mathcal{L}_{\text{FM}}$ by 1.0 and set λ_{text}, λ_{global}, λ_{spatial}, and λ_{iso} to 0.1. Flow matching uses 1,000 timesteps, a flow shift of 3.0, and bell-shaped timestep weights normalized to unit mean. Classifier-free negative conditioning is sampled with probability 0.1.

In the later stages of table 6, E_ϕ is frozen and the same $\mathcal{L}_{\text{gen}}$ is retained; only the terms connected to the optimized memory-conditioning branch and DiT LoRA adapters contribute parameter gradients.

B.2 Reconstruction Supervision

Reconstruction supervision follows a latent-first training pipeline. The memory encoder and reconstruction decoder first learn stable content recovery directly in the VAE latent space. A subsequent perceptual stage decodes the reconstructed latents into RGB video and adds pixel, perceptual, and adversarial supervision to improve visual detail, while isotropic regularization remains active throughout.

VAE latent reconstruction. Reconstruction supervision replaces the generator with a mirrored reconstruction decoder D_ψ trained jointly with the memory encoder. Given a sampled memory prefix, the decoder predicts the original VAE latents. The VAE latent reconstruction loss $\mathcal{L}_{\text{VAE}}$ measures their mean absolute difference, providing a direct and stable content-recovery target throughout image and video training. We pair it with the same isotropic regularizer used above so that accurate reconstruction does not come at the cost of a collapsed or poorly scaled memory space.

Pixel reconstruction. During perceptual post-training, the frozen Wan2.1 VAE decoder maps both the predicted and target latents back to RGB videos. The pixel loss $\mathcal{L}_{\text{pixel}}$ measures their mean absolute difference at the native decoded resolution, preserving color and low-frequency appearance. Although the VAE parameters remain frozen, gradients pass through its predicted-latent path to the

memory encoder and reconstruction decoder. We therefore enable gradient checkpointing in the VAE decoder to reduce activation memory and avoid out-of-memory errors.

Perceptual similarity. The perceptual loss $\mathcal{L}_{\mathrm{LPIPS}}$ compares frame-level visual features rather than exact pixels, making it more tolerant to small local shifts while encouraging similar structure and appearance. We use a frozen ImageNet-pretrained VGG16 feature network with the standard calibrated LPIPS weights. LPIPS compares matching random 256×256 crops from the target and reconstructed clips and is averaged frame-wise over a sampled 16-frame window. For outputs whose short side is below 256 pixels, this term is disabled.

Adversarial refinement. The adversarial loss $\mathcal{L}_{\mathrm{GAN}}$ uses a randomly initialized 3D PatchGAN to promote realistic local detail and short-range temporal coherence. Its spatiotemporal convolutions progressively reduce the clip resolution and classify local video patches rather than assigning one score to the entire clip. The discriminator sees the same 16-frame, 256×256 crops as LPIPS and uses a hinge objective that assigns positive scores to real patches and negative scores to reconstructed patches. We optimize it separately with AdamW at learning rate 10^{-4}, $(\beta_1, \beta_2) = (0.5, 0.9)$, and zero weight decay. This term is also disabled when the output short side is below 256 pixels.

Reconstruction training objective. Writing $[\cdot]_{\mathrm{post}}$ for the three terms enabled only during perceptual post-training, the complete objective is

$$\mathcal{L}_{\mathrm{recon}} = \lambda_{\mathrm{VAE}}\mathcal{L}_{\mathrm{VAE}} + \lambda_{\mathrm{iso}}\mathcal{R}_{\mathrm{iso}} + \left[\lambda_{\mathrm{pixel}}\mathcal{L}_{\mathrm{pixel}} + \lambda_{\mathrm{LPIPS}}\mathcal{L}_{\mathrm{LPIPS}} + \lambda_{\mathrm{GAN}}\mathcal{L}_{\mathrm{GAN}}\right]_{\mathrm{post}}. \quad (3)$$

For image pretraining, we use $\lambda_{\mathrm{VAE}} = 1.0$ and $\lambda_{\mathrm{iso}} = 0.1$; the video and perceptual stages increase λ_{iso} to 1.0. Perceptual post-training uses $\lambda_{\mathrm{pixel}} = 1.0$, $\lambda_{\mathrm{LPIPS}} = 1.0$, and $\lambda_{\mathrm{GAN}} = 0.1$. The memory encoder and reconstruction decoder use learning rate 5×10^{-5} during this final stage, with all three decoded-video losses enabled from the beginning. As summarized in table 6, the VAE latent reconstruction loss first establishes stable content recovery on images and then videos; perceptual post-training subsequently improves decoded-video quality without exposing an untrained encoder–decoder pair to less stable pixel, perceptual, and adversarial signals.

In the auxiliary *Decoder Supervision* strategy of section C, the decoder is retained during generative training and applies $\mathcal{L}_{\mathrm{VAE}}$ to the complete memory sequence of 32 tokens per 4 video frames, while the generative branch continues to use sampled prefixes.

C Memory-Learning Strategies

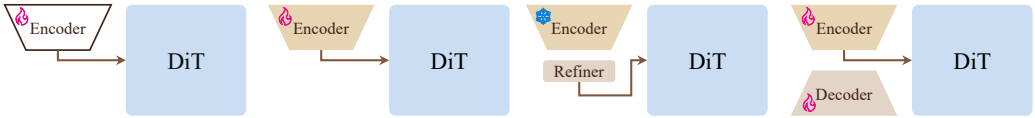
Figure 7 illustrates four strategies for learning the memory encoder, which differ in whether reconstruction is used for initialization, as an auxiliary objective, or not at all. *Scratch* trains a randomly initialized encoder directly with $\mathcal{L}_{\mathrm{gen}}$. *Pretrained* first trains the encoder inside the reconstruction autoencoder with $\mathcal{L}_{\mathrm{recon}}$ and then finetunes it with $\mathcal{L}_{\mathrm{gen}}$.

Refiner keeps the reconstruction-pretrained encoder frozen and inserts a trainable transformer between its output and the generator. For a sampled budget k, the first k tokens from every 4 video frames are retained and flattened into a sequence of length Tk before refinement. The refiner contains four pre-norm residual blocks, each comprising bidirectional self-attention and an FFN, with hidden width 1536, 12 attention heads, and FFN width 6144. 3D RoPE encodes the index of each group of 4 video frames and the token position within that group. The frozen encoder supplies the initial representation, while the refiner and the generator's memory-conditioning branch are optimized with $\mathcal{L}_{\mathrm{gen}}$.

Decoder Supervision retains the pretrained reconstruction decoder and finetunes it jointly with the memory encoder. The generative branch continues to train on a sampled prefix, whereas the auxiliary decoder always receives the complete memory sequence of 32 tokens per 4 video frames and predicts the target VAE latents. In this ablation, the auxiliary term is the VAE latent reconstruction loss only; the decoded-pixel, LPIPS, and adversarial terms are disabled. The optimized objective is $\mathcal{L}_{\mathrm{gen}} + \lambda_{\mathrm{dec}}\mathcal{L}_{\mathrm{VAE}}$, with $\lambda_{\mathrm{dec}} = 1.0$. Thus, the generation loss trains every supported prefix, while decoder supervision separately encourages the full memory sequence to preserve information needed for direct VAE latent reconstruction.

All four strategies use the same prefix sampling for the generative branch and the same frozen generator. As reported in the main paper, *Scratch* is the strongest at truncated budgets: representations

optimized for reconstruction do not transfer effectively to the flexible, generator-oriented memory we seek, and neither a learned refiner nor auxiliary decoder supervision closes this gap.



(a) Scratch (b) Pretrained (c) Refiner (d) Decoder Supervision

Figure 7: Memory-encoder training strategies: (a) *Scratch*, training from scratch with the generation loss; (b) *Pretrained*, finetuning with the generation loss after reconstruction-autoencoder pretraining; (c) *Refiner*, freezing the pretrained encoder and training a transformer refiner on its outputs with the generation loss; and (d) *Decoder Supervision*, retaining the reconstruction decoder and adding full-token latent reconstruction during generative training.

D Implementation Details

This section reports the architecture, attention implementation, and training schedule needed to reproduce the memory encoder E_ϕ under reconstruction and generative supervision. Table 5 compares GenMem-R and GenMem-G; the suffixes denote reconstruction and generative supervision, respectively, and entries centered across both columns are shared.

D.1 Network Architecture

Figure 8 shows the memory encoder and its integration with the pretrained DiT. The encoder applies a 2×2 spatial patch embedding to the VAE latents and assigns $M{=}32$ learnable query tokens to every 4 video frames, where the first query acts as a semantic token and the rest capture complementary content. Following the MMDiT transformer designs of FLUX.1 and HunyuanVideo (Black Forest Labs, 2024; Kong et al., 2024), the memory encoder in both GenMem-R and GenMem-G stacks eight single-stream blocks and four double-stream blocks. Single-stream blocks concatenate memory and video tokens and update them with shared attention and feed-forward layers, which efficiently builds a common representation in which the queries can aggregate visual evidence. Double-stream blocks give each modality its own query, key, value, output, and feed-forward projections while concatenating the projected keys and values for joint attention. The double-stream blocks thus preserve direct memory–video interaction while reserving dedicated capacity for the memory stream to select and refine what it retains.

Three-dimensional rotary position embeddings (3D RoPE) encode time, height, and width. A video patch at (t, h, w) takes its grid coordinate, whereas the r-th memory token of the t-th group of 4 video frames takes $(t, -r-1, 0)$. The negative height coordinate places the queries outside the video grid and is implemented by conjugating the corresponding complex RoPE phase. Memory tokens use the feature width of the video DiT, so the added memory-conditioning projections and feed-forward layers inside the generator can be initialized from their video-stream counterparts. Memory and noised video tokens then interact through joint self-attention while keeping separate feed-forward pathways.

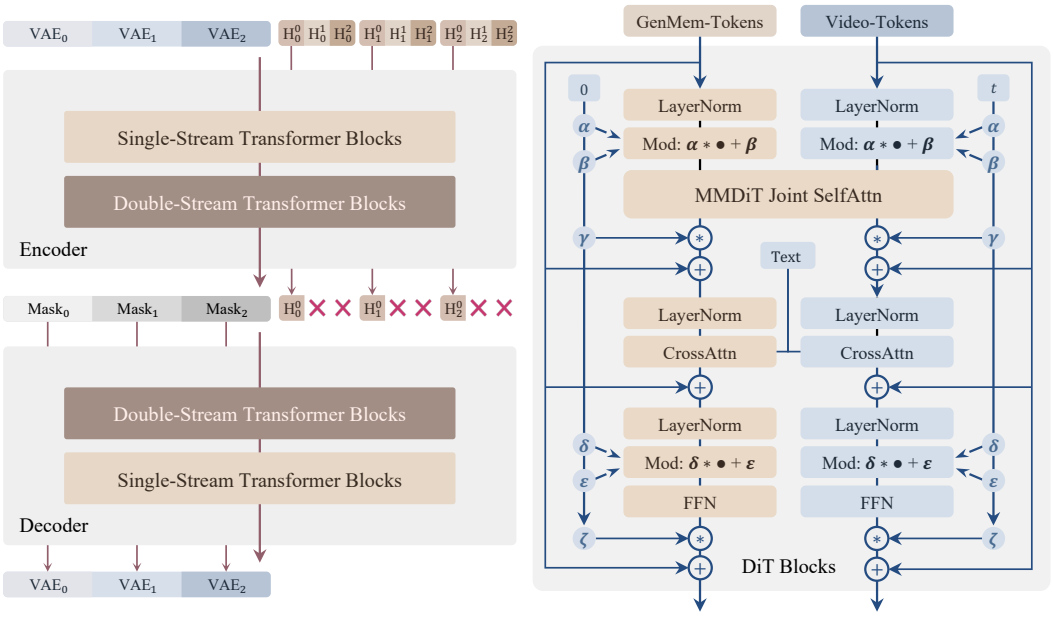
D.2 Causal Window Attention

The encoder processes memory queries and video patch tokens as a single concatenated sequence under the two constraints in fig. 9. The causal sliding window of panel (a) restricts a token for the t-th group of 4 video frames to groups $t-3$ through t, and applies to memory–memory, memory–video, and video–video pairs alike. The memory causality of panel (b) additionally restricts query m_i to keys m_j with $j \le i$, leaving attention to video tokens unaffected; panel (c) shows the combined mask. This second constraint is what makes prefix supervision well posed: because an early memory token cannot read a later one, each prefix $\mathbf{H}_{:k}$ is self-contained, and information accumulates from early to late tokens rather than circulating backward through the discarded tail.

We realize the mask with PyTorch FlexAttention (Dong et al., 2024), combining the two Boolean constraints into a block-sparse `BlockMask` with 128×128 tiles. Frame-index tables are padded to

Table 5: Configurations of GenMem-R and GenMem-G. The suffixes R and G denote reconstruction and generative supervision, respectively. Values are measured per 4 video frames where applicable.

Configuration	**GenMem**-R	**GenMem**-G
Architecture		
Backbone	—	Wan2.1 T2V 1.3B
Patch embedding	2×2 spatial	
Single-stream blocks	8	
Double-stream blocks	4	
Hidden dimension	1536	
Encoder FFN dimension	6144	
Attention heads	12	
Attention head dimension	128	
Latent channels	16	
FFN activation	GELU	
Encoder QKV bias	False	
LayerNorm ϵ	10^{-6}	
Memory tokens	32	
Prefix budgets k	$\{2, 4, 8, 16, 32\}$	
Budget sampling weights	$\{1, 1, 1, 1, 2\}$	
Position encoding	3D RoPE	
Encoder temporal mask	Causal sliding window, size 4	
Encoder memory mask	m_i attends to $m_{\leq i}$	
Encoder attention backend	FlexAttention	
Decoder attention backend	FlexAttention	FlashAttention-3
Training		
Objective	$\mathcal{L}_{\text{recon}}$ (section B.2)	$\mathcal{L}_{\text{gen}}$ (section B.1)
Optimizer	AdamW	
Adam β_1, β_2	$0.9, 0.999$	
Adam ϵ	10^{-8}	
Learning rate	10^{-4} (5×10^{-5} perceptual)	10^{-4}
Learning-rate schedule	Cosine with warmup	Constant with warmup
Warmup steps	500	
Weight decay	10^{-4}	
Per-device batch size	1	
Gradient accumulation	2	
Gradient checkpointing	True	
Distributed optimization	DeepSpeed ZeRO Stage 2	
Precision	BF16 mixed precision	
TF32 enabled	True	



(a) Overview of the GenMem ViT Autoencoder (b) GenMem Integration with Video Diffusion Model

Figure 8: Network architecture of GenMem. (a) The memory encoder aggregates every 4 video frames into an ordered sequence of memory tokens. (b) The memory tokens condition the video DiT through joint memory–video attention while retaining the pretrained text-conditioning pathway.

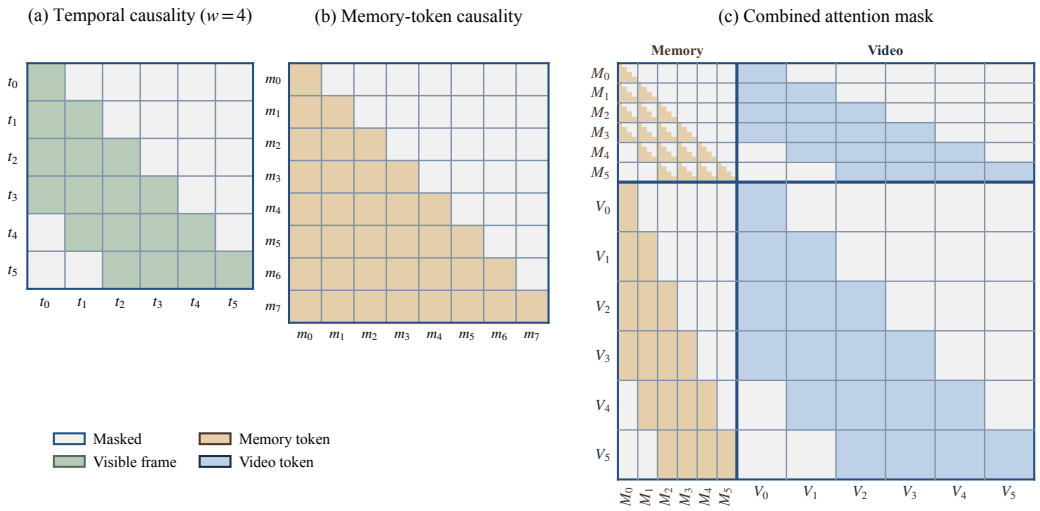


Figure 9: Attention mask used by the memory encoder. Colored cells denote visible query–key pairs, whereas near-white cells are masked. (a) A causal sliding window of size $W = 4$ restricts attention to the current and three preceding groups of 4 video frames. (b) Memory causality permits m_i to attend to m_j only when $j \leq i$. (c) Combining both constraints yields the joint mask over memory-token groups M_t and video-token groups V_t for the t-th group of 4 video frames. Group sizes are reduced for clarity.

a multiple of 128 with sentinel entries and out-of-range positions are rejected explicitly, which keeps the trailing partial tile correct under compilation. Block masks are cached by sequence geometry, window configuration, and device, with at most 64 layouts retained. The kernel is compiled with dynamic shapes and consumes the sparse layout directly, so no dense $S \times S$ mask or attention

matrix is ever materialized. Prefix sampling is handled in the same way, through a per-sample score modifier that assigns $-\infty$ to every discarded key; both supervision pipelines therefore train on self-contained prefixes at no additional memory cost.

D.3 Training Curriculum

Table 6 summarizes the two curricula in chronological order. Reconstruction supervision first trains the memory encoder and reconstruction decoder on LAION-5B images with the VAE latent reconstruction loss and isotropic regularization. The same modules are then transferred to 5-second Koala-36M clips. A final post-training stage retains the VAE latent reconstruction loss while adding decoded-pixel, LPIPS, and 3D PatchGAN terms, allowing perceptual refinement to begin from an already stable latent autoencoder.

Generative supervision likewise begins on images and transfers to 5-second videos, but jointly optimizes the memory encoder and the DiT's memory-conditioning branch with $\mathcal{L}_{\text{gen}}$. We then freeze the memory encoder and train only the memory-conditioning branch so that the generator learns to read a fixed memory representation. In the final stage, rank-8 LoRA adapters are enabled on the DiT attention and feed-forward projections and are optimized together with the memory-conditioning branch; the pretrained DiT weights remain frozen. During video training, first-frame and last-history-frame conditioning are sampled as mutually exclusive modes with probability 0.1 each, while the remaining samples condition on memory tokens alone.

Table 6: Training curriculum for reconstruction and generative supervision. Rows are ordered chronologically within each pipeline, with each stage initialized from the preceding stage. "Mem.-cond. branch" denotes the memory-conditioning branch in the video DiT.

Optimized modules	**Dataset**	**Losses**
Reconstruction supervision		
Memory encoder, reconstruction decoder	LAION-5B images	$\mathcal{L}_{\text{VAE}} + \mathcal{R}_{\text{iso}}$
Memory encoder, reconstruction decoder	Koala-36M 5 s clips	$\mathcal{L}_{\text{VAE}} + \mathcal{R}_{\text{iso}}$
Memory encoder, reconstruction decoder, 3D PatchGAN	Koala-36M 5 s clips	$\mathcal{L}_{\text{recon}}$
Generative supervision		
Memory encoder, mem.-cond. branch	LAION-5B images	$\mathcal{L}_{\text{gen}}$
Memory encoder, mem.-cond. branch	Koala-36M 5 s clips	$\mathcal{L}_{\text{gen}}$
Mem.-cond. branch	Koala-36M 5 s clips	$\mathcal{L}_{\text{gen}}$
Mem.-cond. branch, DiT LoRA (rank 8)	Koala-36M 5 s clips	$\mathcal{L}_{\text{gen}}$

The staged freeze separates representation learning from generator adaptation: the first two stages determine what the memory stores, whereas the last two determine how the DiT reads that fixed representation. Restricting adaptation to the memory-conditioning branch and low-rank updates preserves the pretrained video prior. Raising the LoRA rank or finetuning the DiT in full did not improve fidelity and degraded semantic and motion preservation, indicating that generator capacity is not the bottleneck.

All stages use AdamW with BF16 mixed precision. The encoder runs on the FlexAttention path of section D.2, while the DiT keeps its own optimized attention implementation. The 3D PatchGAN used in perceptual post-training is optimized separately with a hinge loss.



Figure 10: Reconstruction of an acoustic guitar during a camera move through an indoor scene. VideoFlexTok changes the guitar's color and geometry and, at two tokens, largely loses the instrument and room identity. GenMem-G preserves the guitar, plant, furniture layout, and camera motion with a memory budget of only two tokens.

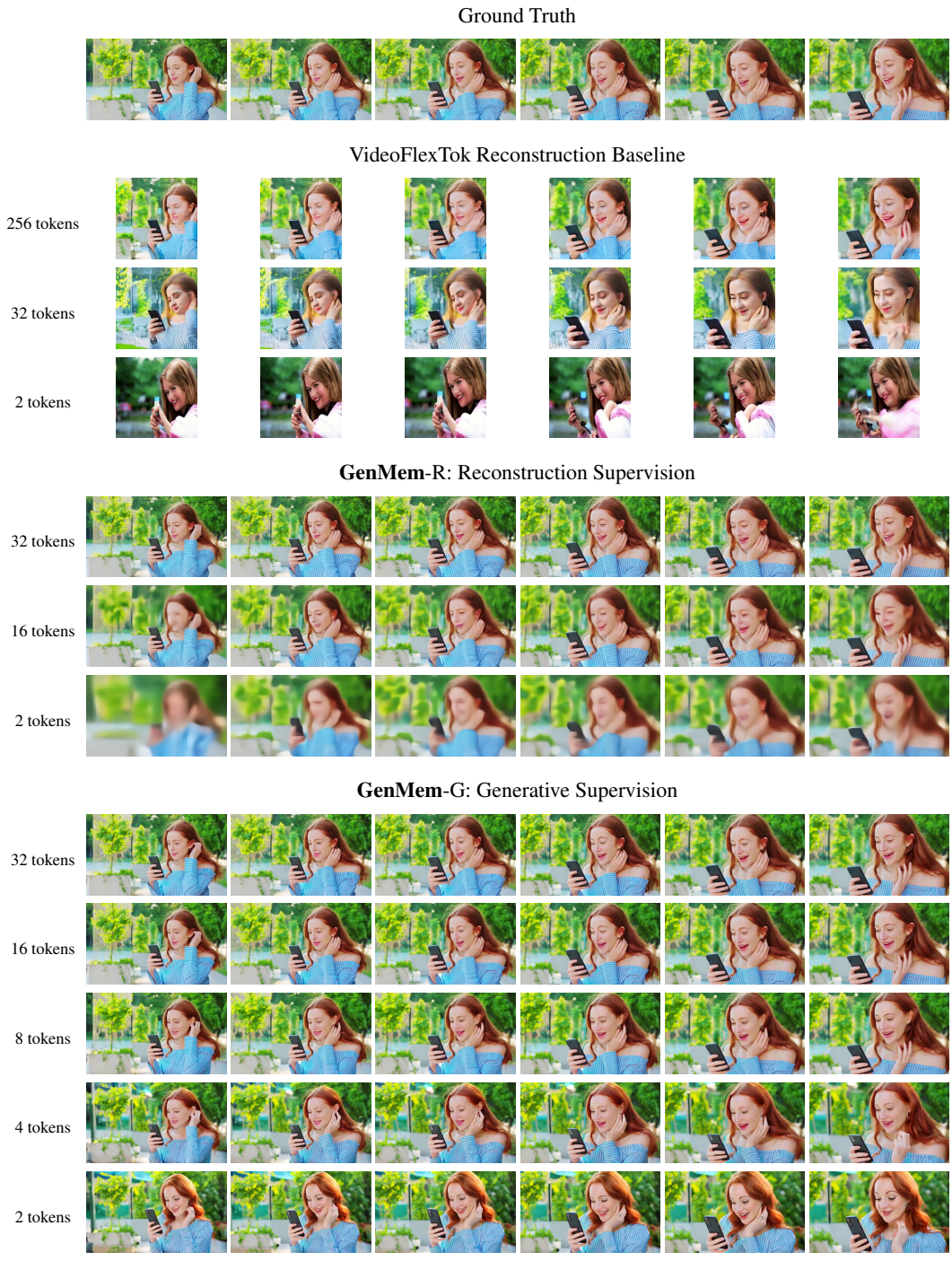


Figure 11: Reconstruction of a woman reacting to her phone. VideoFlexTok changes her identity, hair and clothing colors, and hand geometry, with severe semantic drift at two tokens. GenMem-G preserves the woman, phone, blue striped clothing, outdoor setting, and hand gesture with a memory budget of only two tokens.

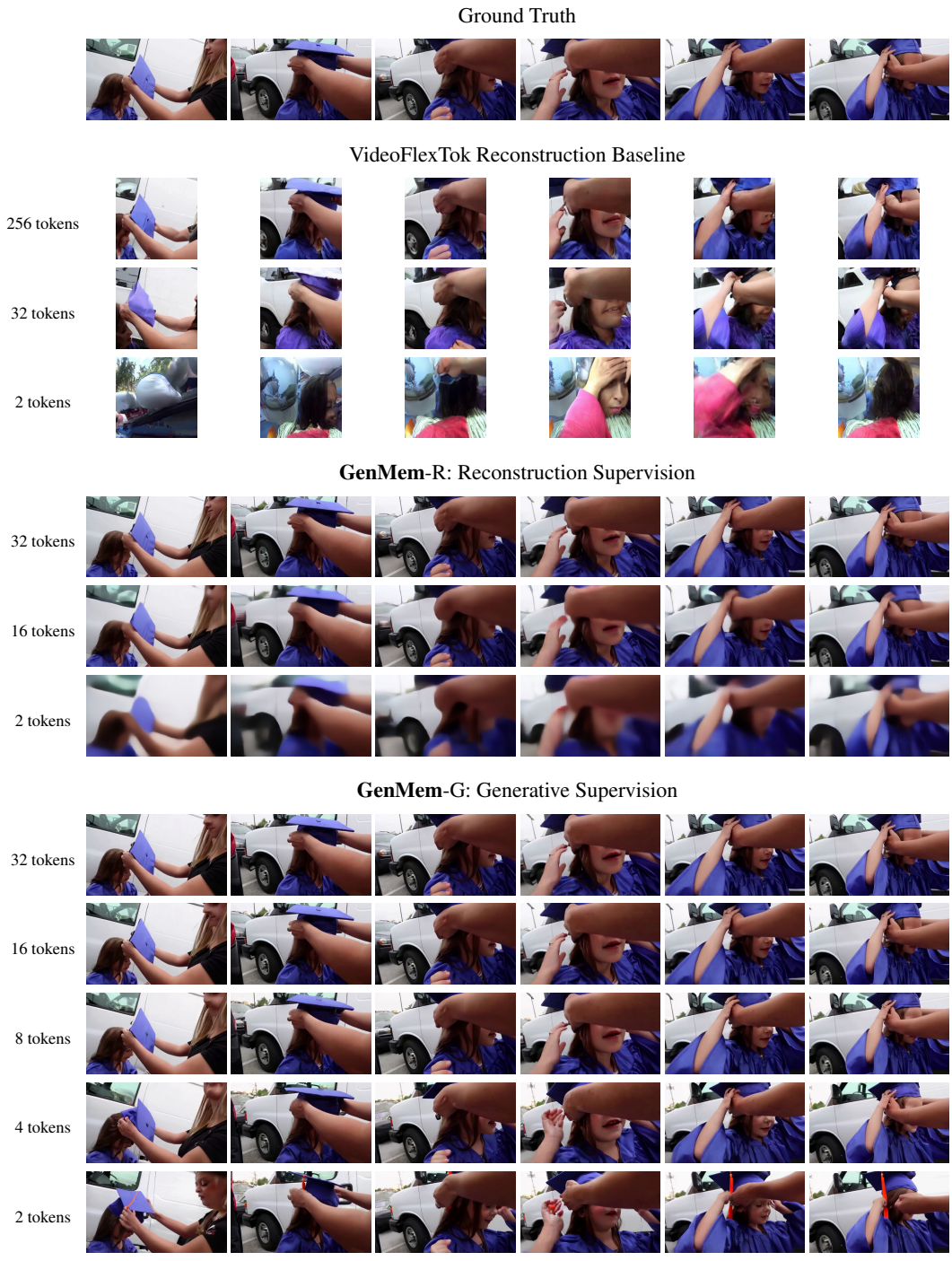


Figure 12: Reconstruction of a graduate adjusting her cap beside a white van. VideoFlexTok changes the graduate's identity and clothing color and, at two tokens, loses the graduation scene and hand–cap interaction. GenMem-G preserves the blue gown, cap, van, and coordinated hand motion with a memory budget of only two tokens.

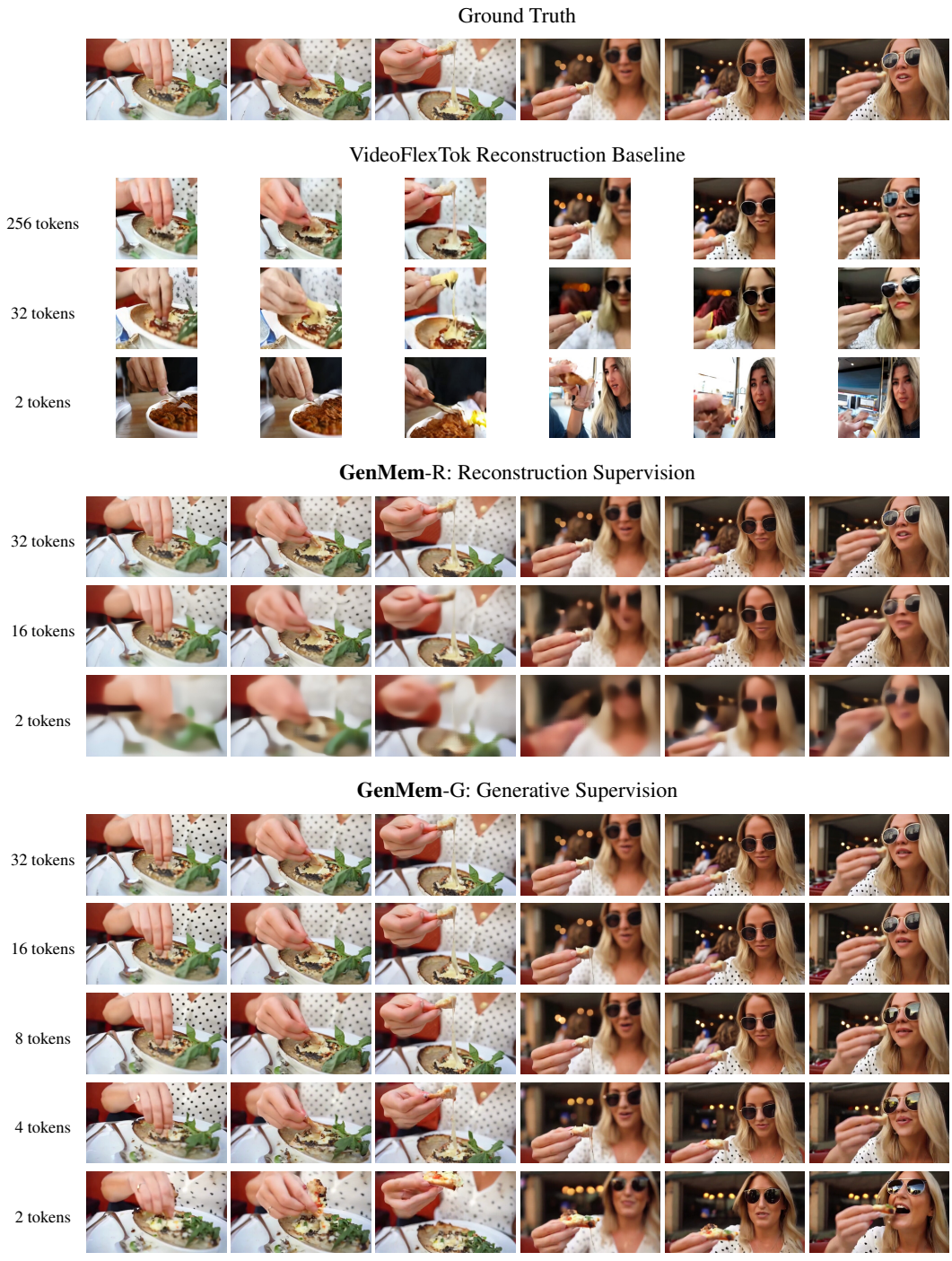


Figure 13: Reconstruction of a woman in sunglasses raising food to her mouth. VideoFlexTok shifts her identity and appearance and, at two tokens, replaces the woman and restaurant scene while breaking the eating action. GenMem-G preserves the woman, sunglasses, food, cafe setting, and hand-to-mouth motion with a memory budget of only two tokens.



Figure 14: More qualitative results of text-responsive subject editing. The captions change the superhero costume, transform a child into an older adult, and modify subject identity in studio, interview, and office scenes. The generated videos follow these edits while preserving the source pose, object interaction, scene layout, and camera motion.



Figure 15: More qualitative results of text-responsive background editing. Given an input video of a speaker in front of the Jefferson Memorial, the captions replace the background with the Eiffel Tower, Golden Gate Bridge, Colosseum, or Temple of Heaven. The generated videos exhibit strong text adherence while preserving the speaker's identity, framing, gestures, and temporal motion.

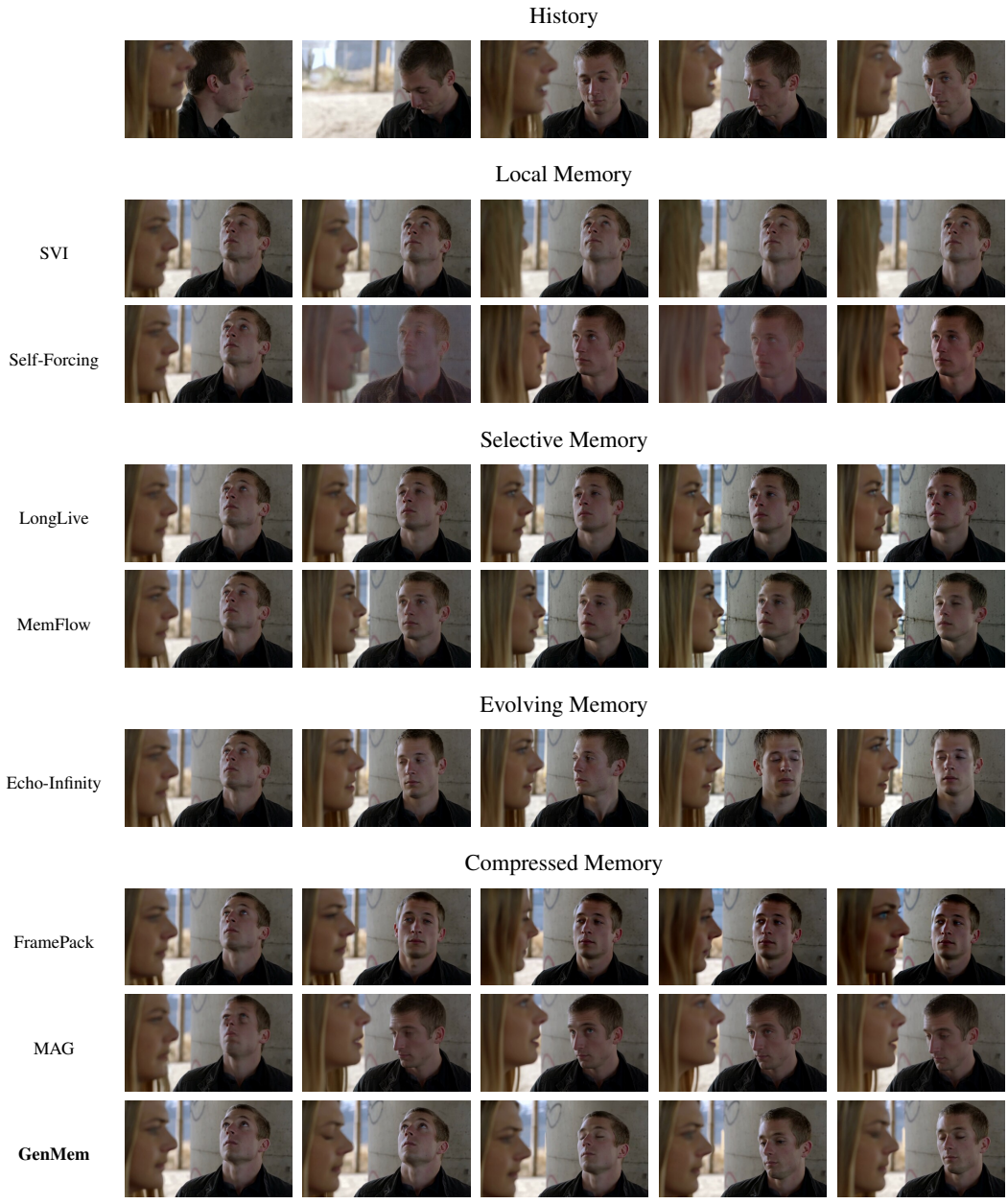


Figure 16: Standard video continuation of a close-up dialogue. Several competing methods drift in facial identity or speaker composition, whereas GenMem preserves the two speakers and the close-up scene structure.

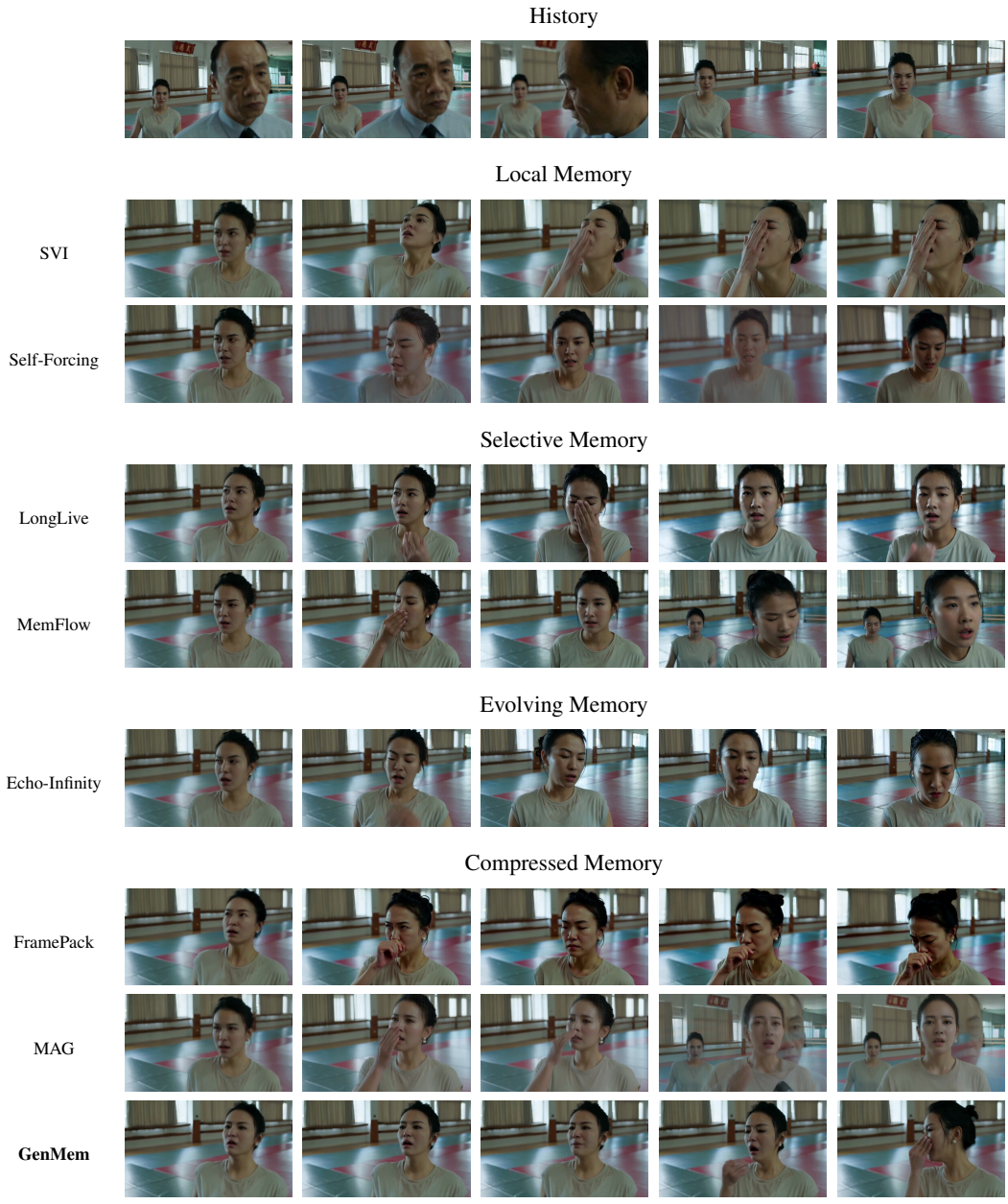


Figure 17: Standard video continuation of a moving subject in a bright indoor space. GenMem maintains the subject's identity, pose, and hand-to-face action more consistently throughout the prediction.

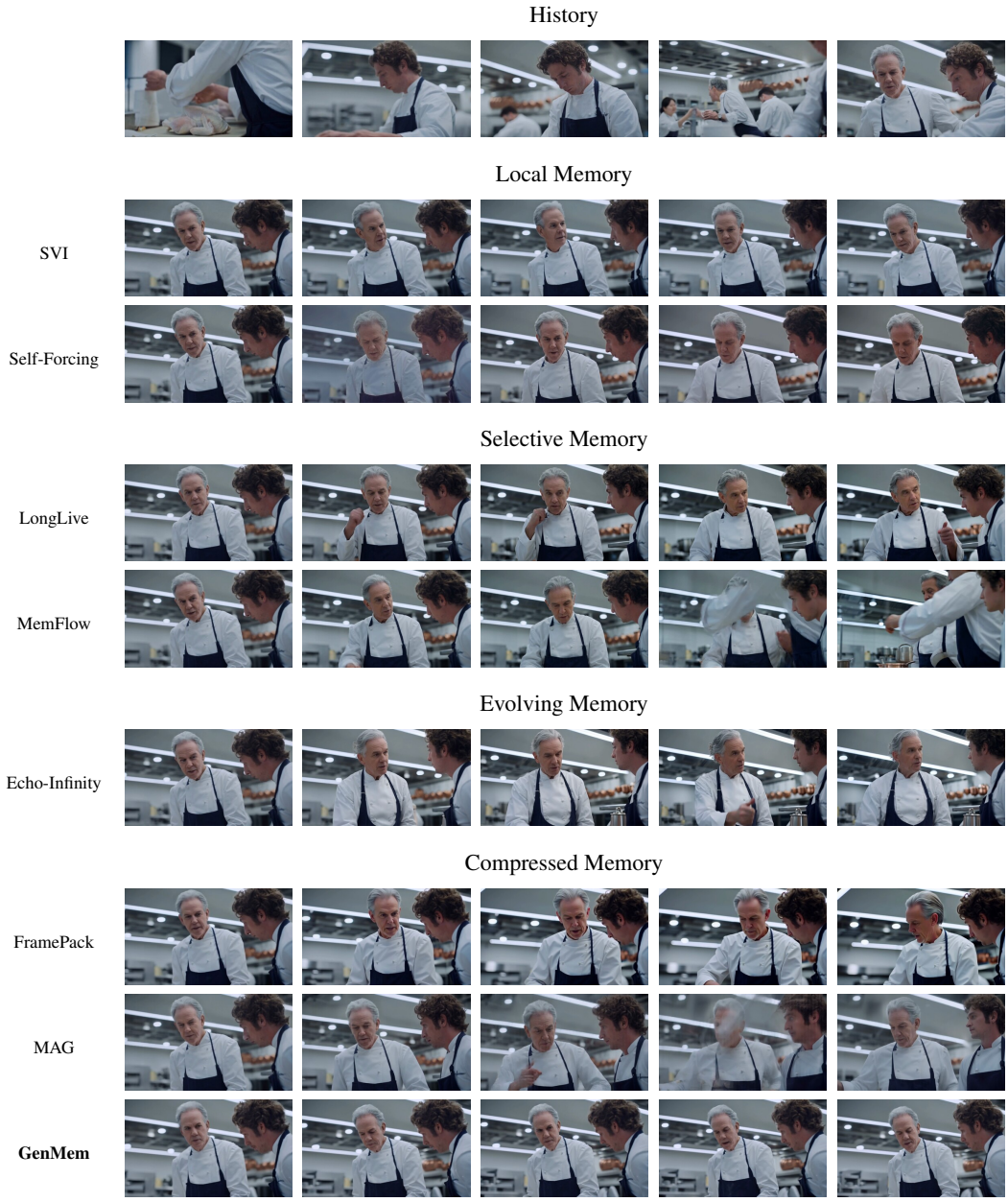


Figure 18: Standard video continuation of a two-person kitchen interaction. GenMem preserves both participants, their spatial relationship, and the surrounding kitchen structure.

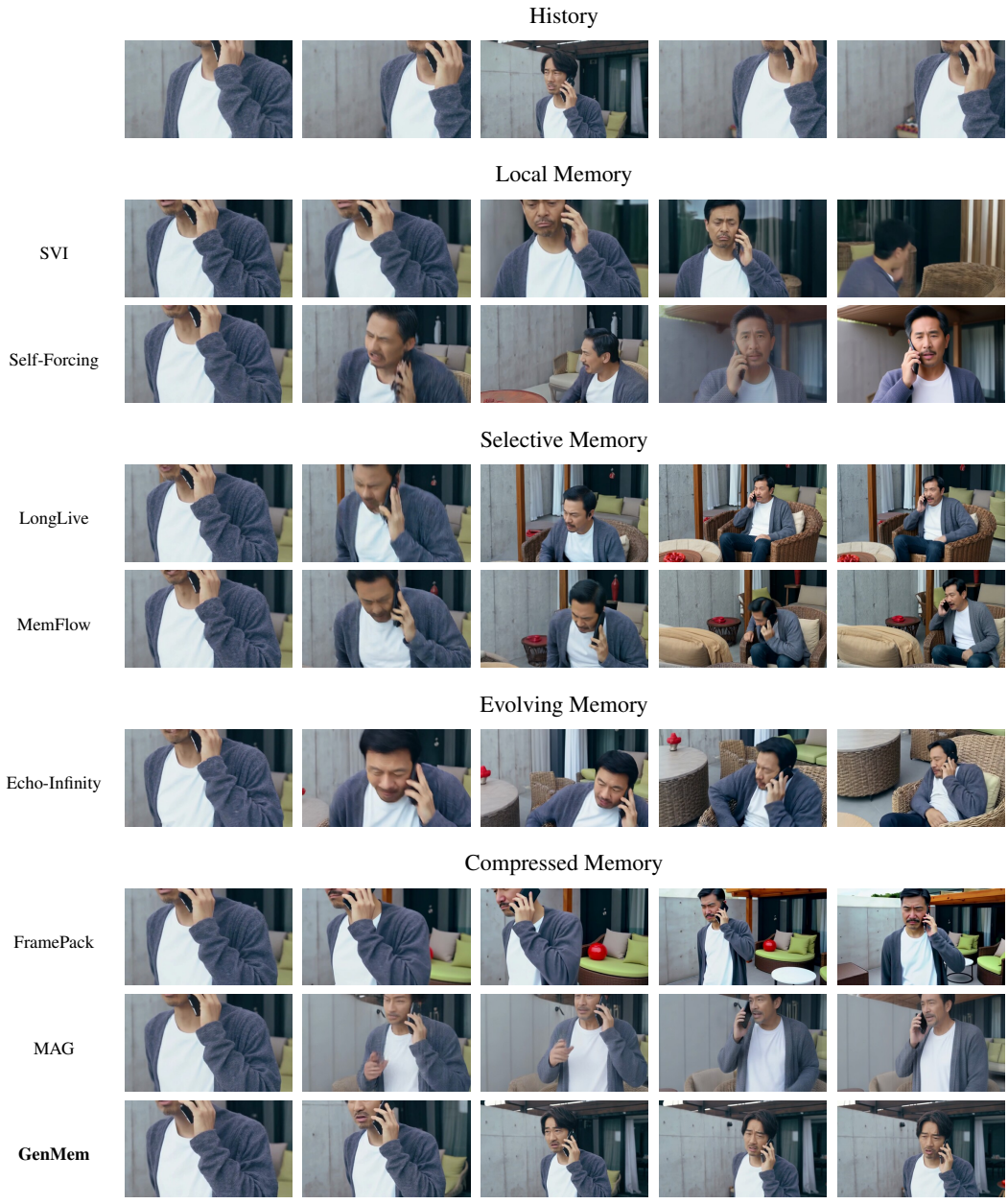


Figure 19: Zoom-augmented video continuation of a man speaking on a phone. From the earlier full-view history, GenMem recovers the man's identity, standing pose, clothing, and outdoor patio beyond the recent crop.

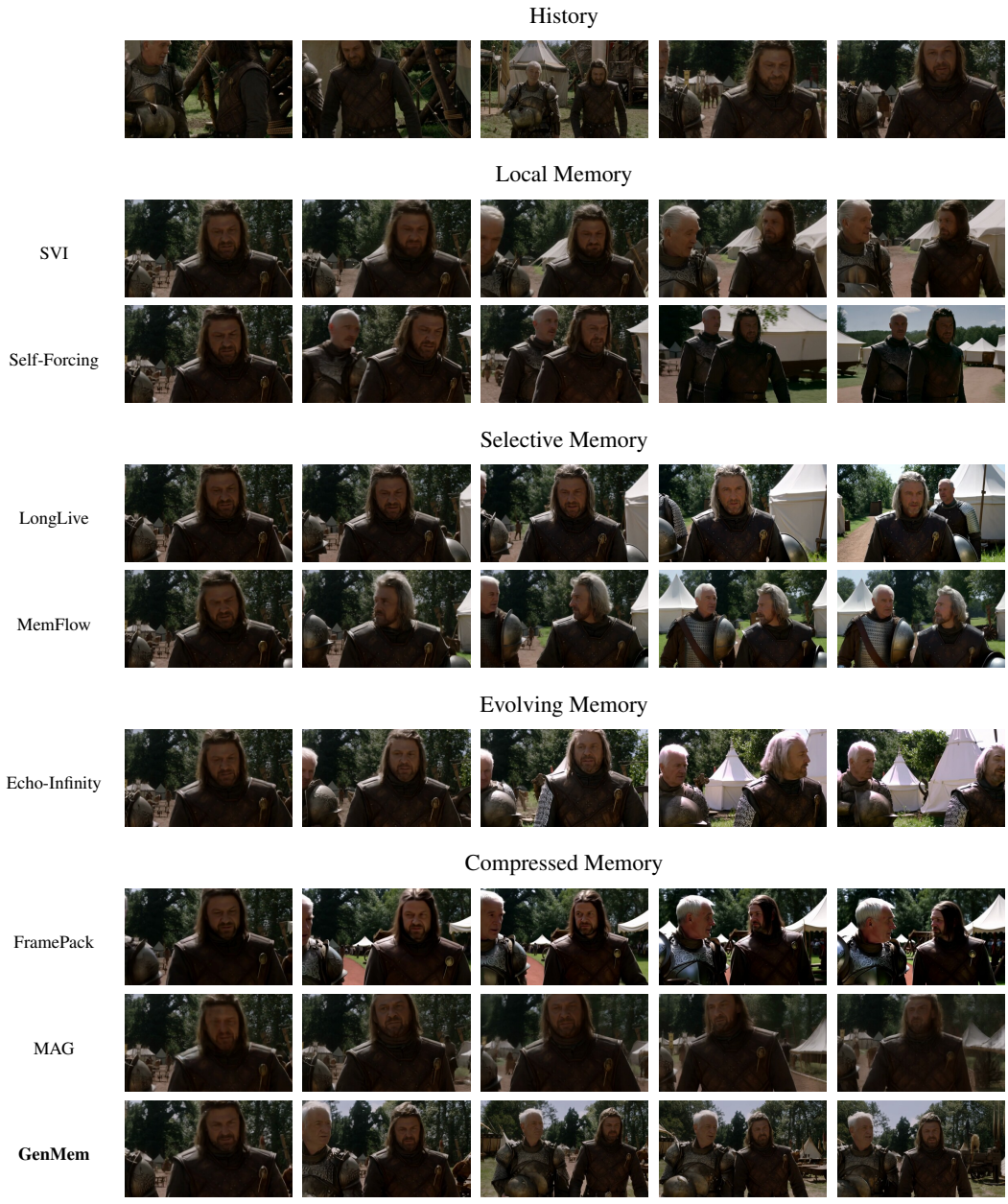


Figure 20: Zoom-augmented video continuation of an outdoor character interaction. GenMem restores the second character, their relative positions, and the wider encampment visible only in the earlier full-view history.

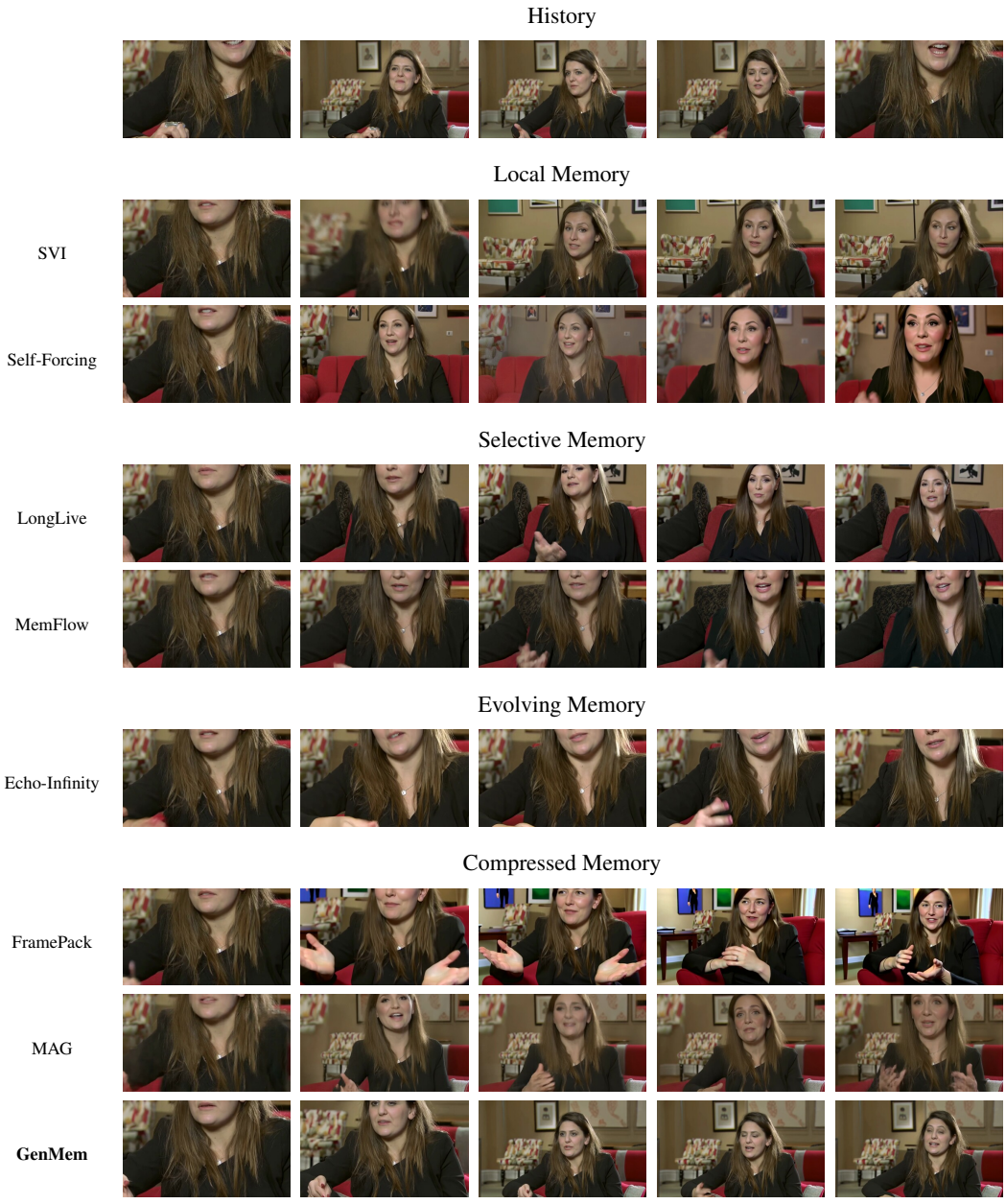


Figure 21: Zoom-augmented video continuation of a seated interview. GenMem recovers the speaker's identity, full seated composition, chair, and background from the earlier full-view history.