TEMPORAL RESIDUAL FOR CONSISTENT AUTOREGRESSIVE LONG VIDEO DIFFUSION

Anonymous authors
Paper under double-blind review



Figure 1: LongLive (top) and Temporal Residual (TR)+LongLive (bottom) at 0, 20, 40, and 60 seconds for the displayed prompt. TR preserves the child's appearance while achieving a higher dynamic degree. Full generated videos are available in the supplements.

ABSTRACT

Autoregressive video diffusion enables efficient long-video generation, but maintaining visual consistency while allowing scenes to evolve remains challenging. Existing approaches improve robustness to generated histories or retain selected visual anchors, yet bounded context windows lose access to earlier appearance cues, while fixed visual anchors cannot summarize an evolving history and restrict video motion. To resolve the three-way tension among long-range consistency, meaningful dynamics, and the efficiency of bounded-context autoregressive video generation, an efficient global memory that evolves with video progress is needed alongside local context and fixed visual anchors. We introduce *Temporal Residual* (TR), a lightweight evolving plug-in global state that extends pretrained generators through brief post-training. TR combines three key designs: *Evolving Global State*, *Dual-Path Memory Injection*, and *Chunk-Synchronous Evolution Schedule*. Together, they make evolving visual history available throughout generation while allowing the scene to continue changing. Applied to Self Forcing and LongLive with brief post-training, TR improves long-range consistency and visual quality while preserving meaningful dynamics. Human preferences and ablations further support the value of this global state.

1 INTRODUCTION

Long video generation requires more than extending a sequence of plausible short clips. A subject must remain recognizable while moving, a scene must retain its identity while the viewpoint changes, while the amount of motion must remain appropriate to the user's prompt. Diffusion transformers have substantially improved short-video synthesis (Peebles & Xie, 2023; Wan, Team et al., 2025; Gao et al., 2025; Seedance, Team et al., 2025; 2026), but full-sequence denoising couples computation and memory to a preallocated temporal canvas. Autoregressive diffusion instead appends frames or chunks while reusing cached features (Chen et al., 2024; Yin et al., 2025; Huang et al., 2025a). Its bounded context makes long generation practical, yet also limits how far the model can look back as a video unfolds.

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 methods improve long-video generation in two complementary ways. Self-rollout training, long-horizon tuning, and moving-window refinement make generators more robust to their own predictions and better coordinate nearby frames (Huang et al., 2025a; Yang et al., 2025; Liu et al., 2025; Zhu et al., 2026). Opening-frame sinks and adaptive or position-adjusted anchors instead retain visual references after local features are evicted (Yang et al., 2025; Liu et al., 2025; Lu et al., 2026; Cui et al., 2026). The former cannot recover forgotten appearance cues; the latter preserves selected references without fully representing how the scene has evolved. Expanding the full context would sacrifice the bounded-context efficiency of autoregressive generation. These limitations motivate an efficient global memory that persists and evolves with the video.

This need suggests separating the roles of recent and distant context. In addition to the local key-value (KV) window that retains detailed evidence for current motion and rendering and the fixed anchor frames to supply texture details, a much smaller state can preserve information that remains relevant after it leaves the KV cache window. Crucially, this persistent state should evolve with committed video content rather than freeze the scene at its first appearance. Pretrained video diffusion transformers (DiTs) already provide strong appearance and motion priors, so such a state need only to guide the existing generator, not replace its temporal modeling. This offers a way to retain useful history without expanding the local attention window as the video grows.

We introduce *Temporal Residual* (TR), a lightweight plug-in module that serves as an evolving global memory for autoregressive long video diffusion models. Its three key designs are (i) *Evolving Global State*, a compact recurrent summary maintained alongside the bounded cache; (ii) *Dual-Path Memory Injection*, which exposes the state through feature modulation and attention while reusing the host DiT; and (iii) a *Chunk-Synchronous Evolution Schedule*, which updates memory once per committed chunk instead of on every denoising pass. The state therefore persists across cache eviction and changes with the scene, while the backbone continues to generate from its familiar local context. The same interface is attached to generators with different cache policies without expanding their window with video length.

We implemented TR on two popular autoregressive hosts, Self Forcing (Huang et al., 2025a) and LongLive (Yang et al., 2025), and TR brings significant improvements to them. On 60-second MovieGenBench (Polyak et al., 2024), adding it to LongLive preserves already strong consistency while allowing substantially more scene movement, boosting the dynamic degree from 35.18 to 52.85; adding it to Self Forcing improves consistency and visual quality while retaining meaningful motion. These gains show that an evolving history channel can help whether a host begins with a fixed anchor or only a moving window. Paired human judgments favor both TR variants, and ablations show why a compact, continuously updated state is more useful than stronger historical conditioning alone.

Our contributions are threefold: (i) We introduce a **compact recurrent history channel** that extends the temporal reach of bounded-context autoregressive video DiTs without growing their local KV windows; (ii) We develop a **host-compatible** read-generate-write interface through **Evolving Global State**,**Dual-Path Memory Injection**, and a **Chunk-Synchronous Evolution Schedule**; (iii) We demonstrate **consistent gains in long-video consistency and visual quality** across Self Forcing and LongLive while preserving meaningful dynamics, supported by quantitative and qualitative results and human preferences.

2 Related Work

Autoregressive video diffusion makes long generation practical by reusing a bounded context (Chen et al., 2024; Yin et al., 2025). Self Forcing, Causal Forcing, and LongLive improve robustness over generated histories and extended rollouts, while Rolling Forcing jointly refines frames within a moving window (Huang et al., 2025a; Zhu et al., 2026; Yang et al., 2025; Liu et al., 2025). These methods improve how the generator uses its available context, but cannot restore appearance cues once they leave the window. TR complements them with a persistent summary that does not require the local window to grow with video length.

Opening-frame sinks retain an early visual reference (Yang et al., 2025; Liu et al., 2025); adaptive anchors and positional schemes seek to reduce the rigidity of that reference (Lu et al., 2026; Cui et al., 2026; Dalva & Yanardag, 2026). Other approaches retrieve past features or compress history

into recurrent memory (Zhao et al., 2026; Hu et al., 2026; Yu et al., 2025; Li et al., 2026; Liu et al., 2026). Closest to TR, VideoSSM (Yu et al., 2025) updates an SSM state from evicted tokens and ARL^2 (Li et al., 2026) replaces cross-frame attention with a recurrent linear state written after the clean pass; TR instead keeps the host attention window and cache intact and injects a per-block state through both AdaLN and attention, allowing the same interface to complement both sink-free and sink-based generators.

3 METHOD

Design overview. *Temporal Residual* (TR) extends the temporal reach of a pretrained video DiT without expanding its local KV window. A full history of frame features would grow with video length, while a fixed reference would fail to track what the video has become. Our design separates a small persistent summary from the detailed recent context and gives each a distinct role. We realize this through (i) *Evolving Global State*, which carries a recurrent summary beyond the bounded window; (ii) *Dual-Path Memory Injection*, which lets that summary guide feature modulation and attention through the existing DiT; and (iii) a *Chunk-Synchronous Evolution Schedule*, which changes the summary only when a generated chunk is committed. The three choices make history persistent, accessible, and aligned with actual video progress.

Together these choices form a read-generate-write loop beside the existing generator (Figure 2). The host cache supplies detailed recent evidence, while the recurrent state provides a condition that can outlive cache eviction. Every denoising pass reads the same prior state; only the committed chunk contributes a write. The number of state tokens and the cache budget remain fixed as generation proceeds, and the backbone retains responsibility for synthesis. We first define the state, then its two read paths, its write schedule, and the training interface.

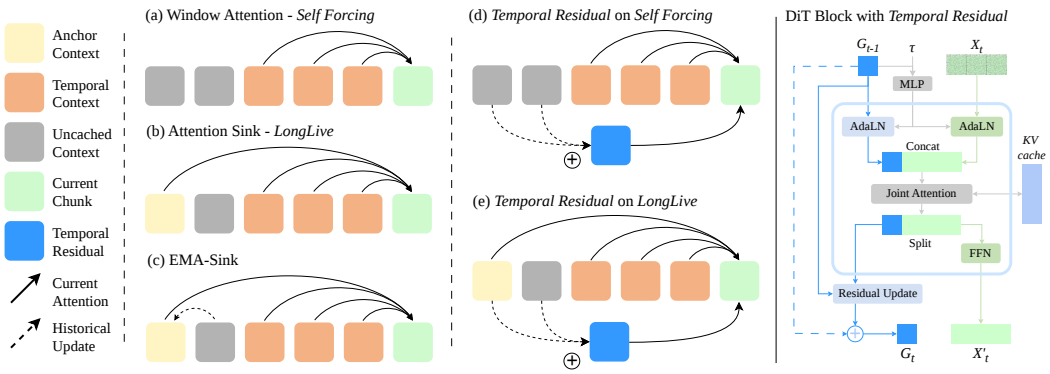

Figure 2: **Context schedule comparison and the TR block structure.** Left: Existing Methods (a-c) and TR added to Self Forcing and LongLive (d,e); Section 3.1 explains their cache policies. Right: the TR block design where the previous global state and timestep condition AdaLN; state and vision tokens attend jointly with host KV, then split into residual state-update and vision-FFN paths.

3.1 EVOLVING GLOBAL STATE

A bounded KV window preserves detailed recent features but loses access to earlier content as generation proceeds. TR complements it with a compact state in each DiT block. This state persists across the video and evolves with generated content; it is a learned summary rather than a stored frame or a fixed visual anchor.

Let an autoregressive generator produce N latent chunks $x_{1:N}$, each containing F frames with P vision tokens per frame. We index chunks by $t \in \{1, \dots, N\}$, denoising passes by $s \in \{1, \dots, S\}$, and DiT blocks by $\ell \in \{1, \dots, L\}$. At pass s of chunk t, the generator predicts $\hat{x}_{t,s}$ from noisy latent $z_{t,s}$, diffusion timestep τ_s, prompt c_t, and retained context. The host scheduler renoises intermediate predictions. Its *exit pass* $s^\star$ commits $x_t = \hat{x}_{t,s^\star}$; $s^\star = S$ at inference and is sampled during self-rollout training (Huang et al., 2025a). A timestep-zero pass then caches clean features of x_t.

The host retains up to n_{loc} local latent frames, including the current chunk, and optionally n_{sink} opening frames, giving at most $(n_{\text{sink}} + n_{\text{loc}})P$ vision tokens in its attention context. TR keeps a

separate state $G_t^\ell \in \mathbb{R}^{n_g \times d}$ for each block, where n_g is the number of state tokens and d is the block width. We use $n_g = 1$ by default and initialize $G_0^\ell = 0$. Every pass of chunk t reads the same G_{t-1}^ℓ; only committing the chunk produces G_t^ℓ. Figure 2 omits the block index on G for clarity.

Figure 2(a–c) contrasts Self Forcing's sink-free window, LongLive's fixed opening sink, and an EMA-sink that averages evicted KV features. TR leaves each host cache intact in panels (d,e). Its state is neither another sink slot nor an average of evicted features: a learned residual update incorporates evidence from the committed chunk (Section 3.3).

3.2 Dual-Path Memory Injection

TR injects history through the host's existing DiT blocks rather than replacing their attention. Since a single state token can be diluted among many visual tokens, TR combines two read paths: block-wide adaptive layer normalization (AdaLN) modulation and token-selective attention. At denoising pass s of chunk t, block ℓ receives six host AdaLN vectors $m_s^\ell \in \mathbb{R}^{6\times d}$: shift, scale, and residual-output scale for self-attention and for the feed-forward network (FFN). Let e_{τ_s} be the host's timestep embedding, Pool mean-pool the state tokens, and $[\,;\,]$ concatenate features. A bottleneck MLP produces the memory-dependent correction

$$A_{t,s}^\ell = M_{\mathrm{mod}}^\ell\big([e_{\tau_s}; \mathrm{Pool}(G_{t-1}^\ell)]\big), \qquad \widetilde{m}_{t,s}^\ell = m_s^\ell + A_{t,s}^\ell. \tag{1}$$

The superscript (j) selects one of the six vectors in $A_{t,s}^\ell$ or $\widetilde{m}_{t,s}^\ell$; vectors 1–3 serve self-attention and 4–6 serve the FFN. Thus the same persistent state can condition different denoising stages without changing the host's prompt conditioning.

The attention path makes the state directly accessible to individual visual tokens. With host feature normalization Norm and channelwise multiplication $\odot$ (broadcast over state tokens), we form

$$\widehat{G}_{t,s}^\ell = \mathrm{Norm}(G_{t-1}^\ell) \odot (1 + A_{t,s}^{\ell,(2)}) + A_{t,s}^{\ell,(1)}. \tag{2}$$

TR reuses the host projections Q^ℓ, K^ℓ, V^ℓ, including query–key normalization where applicable. Let R_{p_t} denote the host rotary embedding (Su et al., 2021) at spatial origin and the first frame of chunk t, with zero-based frame index $p_t = F(t-1)$, and let γ^ℓ be a learned scalar. The state projections are

$$q_{G,t,s}^\ell = \gamma^\ell R_{p_t} Q^\ell(\widehat{G}_{t,s}^\ell), \quad k_{G,t,s}^\ell = \gamma^\ell R_{p_t} K^\ell(\widehat{G}_{t,s}^\ell), \quad v_{G,t,s}^\ell = \gamma^\ell V^\ell(\widehat{G}_{t,s}^\ell). \tag{3}$$

They are recomputed at each denoising pass and are never stored in the host KV cache.

Let $U_{t,s}^\ell \in \mathbb{R}^{FP\times d}$ be the AdaLN-normalized current-chunk visual tokens, with projections $Q_{U,t,s}^\ell, K_{U,t,s}^\ell, V_{U,t,s}^\ell$. The host context $K_{C,t,s}^\ell, V_{C,t,s}^\ell$ includes the current chunk, retained local tokens, and any sink tokens. Writing Attn for masked scaled dot-product attention, W_O^ℓ for the host output projection, and $[\,;\,]$ for token-axis concatenation, the two paths produce

$$y_{t,s}^\ell = W_O^\ell\,\mathrm{Attn}\big(Q_{U,t,s}^\ell, [k_{G,t,s}^\ell; K_{C,t,s}^\ell], [v_{G,t,s}^\ell; V_{C,t,s}^\ell]\big), \tag{4}$$

$$g_{t,s}^\ell = W_O^\ell\,\mathrm{Attn}\big(q_{G,t,s}^\ell, [k_{G,t,s}^\ell; K_{U,t,s}^\ell], [v_{G,t,s}^\ell; V_{U,t,s}^\ell]\big). \tag{5}$$

As shown in Figure 2, visual queries read the state together with host context, whereas state queries read only themselves and the current chunk. The visual stream receives the gated residual $\widetilde{m}_{t,s}^{\ell,(3)} \odot y_{t,s}^\ell$ before continuing through the host FFN. The state readout $g_{t,s}^\ell$ supplies evidence for the subsequent write; it bypasses the visual FFN and KV cache.

3.3 Chunk-Synchronous Evolution Schedule

The read paths condition a proposal, but an unfinished denoising pass is not yet a new video event. Writing after each pass would fold several noisy revisions of the same chunk into memory. TR instead holds G_{t-1}^ℓ fixed throughout denoising. At the exit pass $s^\star$, it combines that state with the state-query readout $g_{t,s^\star}^\ell$ to form write evidence H_t^ℓ and a residual increment Δ_t^ℓ. Here LN is affine layer normalization, sg stops gradients through its argument, M_{upd}^ℓ is a tokenwise bottleneck MLP,

and $\lambda^\ell \in \mathbb{R}^d$ is a learned per-channel step size shared across state tokens:

$$H_t^\ell = G_{t-1}^\ell + A_{t,s^\star}^{\ell,(3)} \odot g_{t,s^\star}^\ell, \tag{6}$$

$$\Delta_t^\ell = \lambda^\ell \odot M_{\text{upd}}^\ell\big(\text{LN}\big(\text{sg}(H_t^\ell)\big)\big), \qquad G_t^\ell = G_{t-1}^\ell + \Delta_t^\ell. \tag{7}$$

The residual form carries history forward while allowing revision; a cache-window shift does not erase a prior update, though the state remains a lossy summary.

Detaching H_t^ℓ cuts the backward path from the write into the DiT features that produced its evidence, while the residual connection provides a direct state-gradient path from G_t^ℓ to G_{t-1}^ℓ. During training, the relay described below realizes this path without retaining historical denoising graphs (Appendix B).

For the inference algorithm, $G_t = \{G_t^\ell\}_{\ell=1}^L$ collects all block states and KV denotes the host cache. A working variable z holds the current $z_{t,s}$. One generator pass f_θ, with host and TR parameters θ, returns a latent estimate and $E_{t,s} = \{(A_{t,s}^{\ell,(3)}, g_{t,s}^\ell)\}_{\ell=1}^L$. The operator Write applies Eqs. 6–7 to this evidence. Renoise follows the host scheduler, and $z \sim \mathcal{N}(0, I)$ draws an initial noisy latent with identity covariance I. $\text{Recache}_{\text{no write}}$ recomputes clean KV at timestep zero without updating G_t. At a prompt change, Switch refreshes prompt-dependent KV according to the host; TR's default *Carry* policy preserves G_{t-1}. These definitions distinguish a committed new chunk from other forward passes.

Algorithm 1: Temporal Residual inference

Require: prompts $c_{1:N}$, steps $\tau_{1:S}$, host cache policy
$G_0 \leftarrow \{0\}_{\ell=1}^L$; KV $\leftarrow \varnothing$
for $t = 1, \ldots, N$ **do**
  if $t > 1$ and $c_t \neq c_{t-1}$ **then**
    $(\text{KV}, G_{t-1}) \leftarrow \text{Switch}(\text{KV}, G_{t-1}, c_t)$
  end if
  $z \sim \mathcal{N}(0, I)$
  for $s = 1, \ldots, S$ **do**
    $(\hat{x}_{t,s}, E_{t,s}) \leftarrow f_\theta(z, \tau_s, c_t; \text{KV}, G_{t-1})$
    if $s < S$ **then**
      $z \leftarrow \text{Renoise}(\hat{x}_{t,s}, \tau_{s+1})$
    end if
  end for
  $x_t \leftarrow \hat{x}_{t,S}$; $G_t \leftarrow \text{Write}(G_{t-1}, E_{t,S})$
  $\text{KV} \leftarrow \text{Recache}_{\text{no write}}(x_t, c_t, \text{KV}, G_t)$
end for
return $x_{1:N}$

Algorithm 1 makes the two clocks explicit: every noisy pass of chunk t reads G_{t-1}, and only exit-pass evidence $E_{t,S}$ advances it to G_t. The clean recache reads G_t but cannot write again. Thus the state follows committed chunks rather than denoising or recaching calls.

A prompt switch exposes a related distinction between cached features and historical state. LongLive recaches recent frames under the new text to remove stale prompt-conditioned KV features (Yang et al., 2025). That cache refresh does not mean the preceding visual history ceased to exist. Our default *Carry* policy keeps G across the switch and leaves recaching read-only for state. To test the role of continuity, the *Rebuild* ablation instead resets G and replays recent frames under the new prompt, writing state updates while discarding replay cache writes. The chosen policy is used consistently in training and inference.

3.4 Lightweight Parameter Overhead

TR adds 118.4M trainable parameters, an 8.3% increase over the 1.42B-parameter host. Its one-token-per-block global state is reused across chunks without enlarging the host's local KV window. Thus, TR provides persistent, evolving history while preserving the bounded-context scaling that makes autoregressive long-video generation efficient. Appendix B gives the exact parameter count.

3.5 Post-Training

To integrate the new history channel while preserving the host's learned generation behavior, we initialize the output layers of both TR MLPs, the attention scale γ^ℓ, and G_0^ℓ to zero; input layers use Xavier initialization and $\lambda^\ell = 1$. Thus modulation and state updates begin at zero. The additional attention entries still change softmax normalization, so this is near-host rather than exactly function-preserving initialization (Appendix B).

TR and the host train jointly on self-generated rollout segments under the existing distribution matching distillation objective (Yin et al., 2024b;a), supervised by a frozen teacher and a critic fitted to generated samples. No memory-specific loss or additional reward is used.

Table 1: **Consistency with sustained dynamics in 60-second generation.** VBench-Long scores on MovieGenBench (percent). TR improves all three consistency-related scores and both visual-quality scores over each host; gray subscripts give the change relative to the corresponding host. At similar consistency to Rolling Forcing, both TR models retain higher dynamic degree. Bold marks non-dynamic column maxima; dynamics is interpreted jointly with consistency, as explained in Section 4.2.

Method	Subject consist.↑	Backgr. consist.↑	Motion smooth.↑	Dynamic degree	Aesthetic quality↑	Imaging quality↑
Causal Forcing (Zhu et al., 2026)	95.82	95.24	97.47	79.85	56.50	67.48
Rolling Forcing (Liu et al., 2025)	97.77	96.55	98.68	31.08	59.45	**70.58**
SkyReels-V2 (Chen et al., 2025)	86.09	90.27	**98.94**	55.53	57.07	54.15
EMA-sink (Lu et al., 2026)	96.67	95.79	97.72	68.70	60.19	68.08
Self Forcing (Huang et al., 2025a)	96.05	95.70	97.50	57.77	58.28	66.59
LongLive (Yang et al., 2025)	97.82	96.58	98.83	35.18	60.97	68.38
TR+Self Forcing (ours)	97.85$_{+1.80}$	96.51$_{+0.81}$	98.86$_{+1.36}$	49.91$_{-7.86}$	59.95$_{+1.67}$	70.39$_{+3.80}$
TR+LongLive (ours)	**97.92**$_{+0.10}$	**96.61**$_{+0.03}$	98.85$_{+0.02}$	52.85$_{+17.67}$	**61.34**$_{+0.37}$	69.01$_{+0.63}$

Each supervised chunk is backpropagated immediately, freeing its DiT graph and leaving a detached boundary G as a gradient-tracking leaf. A reverse relay replays detached write evidence through TR updates, including loss-free chunks, depositing gradients at preceding boundaries. With recurrence detachment disabled, this recovers full-stream state-path and TR-parameter gradients; KV writes remain graph-free.

We apply the same TR module to two Wan2.1-based generators, retaining each host's cache policy and training objective. Self Forcing adapts the full generator with its sink-free rolling window; LongLive uses its opening-frame sink and LoRA (Hu et al., 2022) streaming recipe, including prompt switches. This pairing tests whether one history interface can improve generators with different ways of retaining local context. Architecture and host-specific configs are in Appendix A.

4 Experiments

We test whether a fixed-size evolving global state can extend bounded-context generation without trading away meaningful scene evolution. The decisive evidence is transfer across two different popular hosts: Self Forcing (Huang et al., 2025a) uses a moving window without a sink, whereas LongLive (Yang et al., 2025) also retains an opening anchor. We compare each host with its *Temporal Residual* (TR) extension on minute-long videos, examine how the memory design shapes consistency and dynamics, and report the added parameters and human preferences.

4.1 Evaluation protocol

We generate 60-second videos on extended MovieGenBench prompts (Polyak et al., 2024), using the same prompts across models, a 480-pixel short edge, and five samples per prompt. The evaluated subset contains 256 randomly selected prompts from MovieGenBench (Polyak et al., 2024). VBench-Long (Huang et al., 2024; 2025b) evaluates 30 clips per video in `long_custom_input` mode. Outputs are passed directly to the evaluator without an additional pre-evaluation re-encoding. We report its six supported custom-prompt dimensions: subject consistency, background consistency, motion smoothness, dynamic degree, aesthetic quality, and imaging quality. We primarily compare TR with its Self Forcing (Huang et al., 2025a) and LongLive (Yang et al., 2025) hosts in Table 1, and also alongside the recent autoregressive and streaming methods in Section 2.

4.2 Quantitative Results

TR improves the consistency-dynamics balance on both hosts (Table 1 and Figure 4(a)). Dynamic Degree measures visual change, which can also arise from drift or flicker; it is most meaningful when subject consistency, background consistency, and motion smoothness are comparable. Self Forcing starts with substantial visual change, some of which comes from appearance drift over long rollouts. TR raises subject consistency from 96.05 to 97.85 and improves the other consistency and

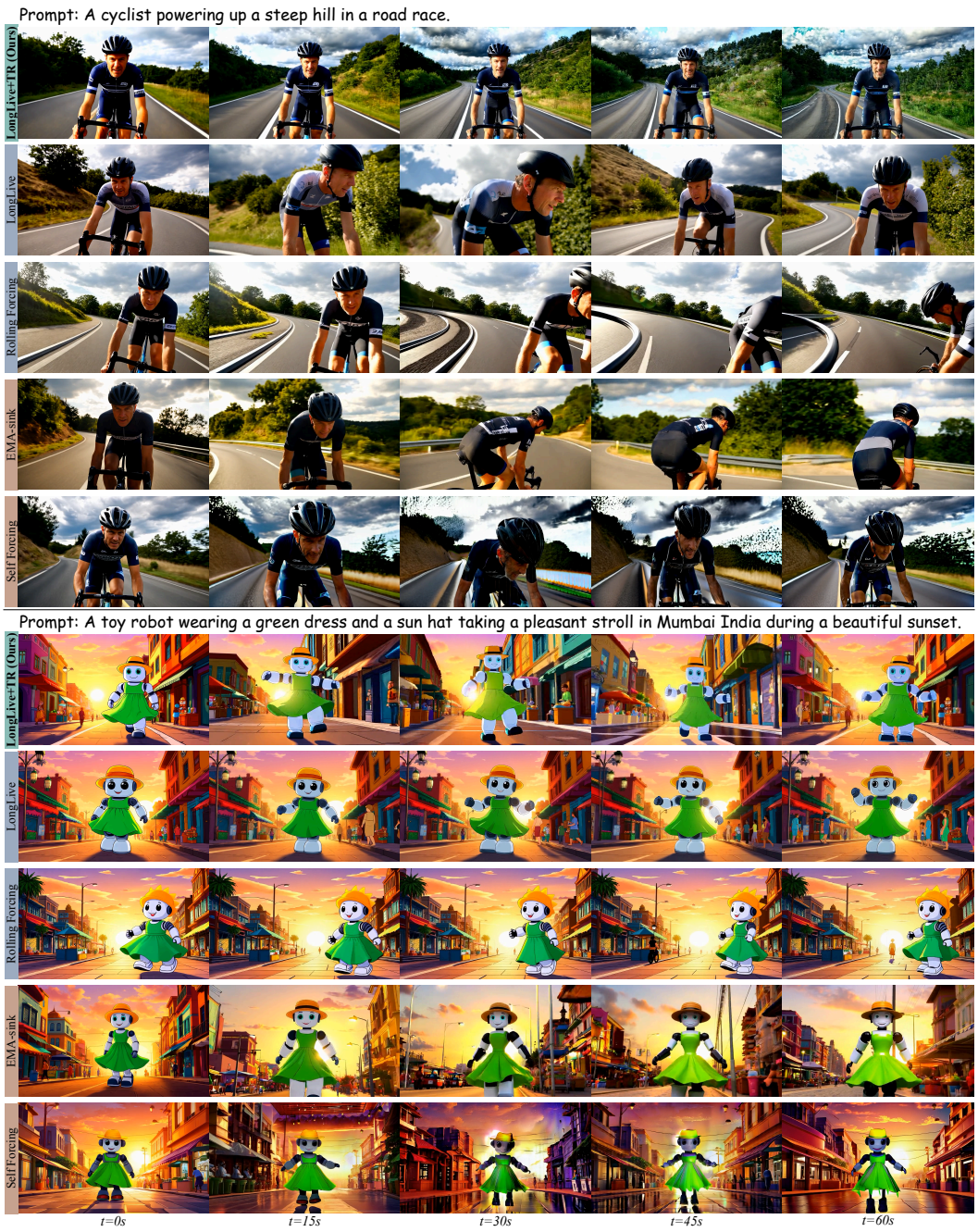

Figure 3: **Visual comparison over 60 seconds.** Five timestamps are shown for each method on two shared prompts. Row-label backgrounds match the operating regions in Figure 4(a): teal for TR, blue for LongLive and Rolling Forcing, and warm orange for EMA-sink and Self Forcing. More generated videos are provided in the Supplementary Material for comparison.

visual-quality scores. Its Dynamic Degree moves from 57.77 to 49.91 as drift is reduced, yet remains substantial rather than collapsing toward a static scene; at comparable consistency, Rolling Forcing scores 31.08.

LongLive already has high consistency, but its opening anchor can leave the scene with limited motion. Adding TR slightly improves all three consistency-related scores while raising Dynamic Degree from 35.18 to 52.85. The result is a more balanced video: the scene continues to change without losing its subject or background. The frame sequences in Figure 3 illustrate these complementary

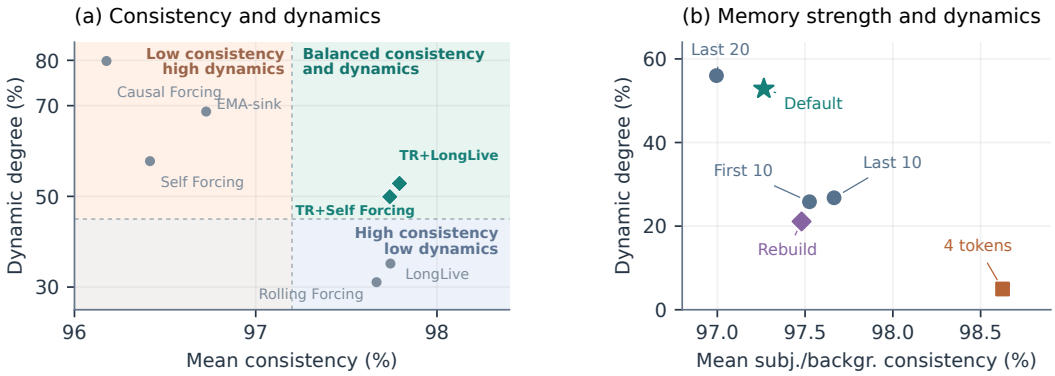

Figure 4: **Useful consistency leaves room for change.** Left: both TR instantiations retain a higher dynamic degree than LongLive and Rolling Forcing at nearby subject/background/smoothness scores. Right: the ablations reveal that very high consistency can coincide with little detected motion. Points are aggregate configurations, not a controlled fixed-consistency curve; Table 2 supplies the other temporal and visual-quality dimensions.

gains, while the human preferences in Figure 5 support the joint improvement in consistency and meaningful motion.

4.3 Qualitative Results and User Study

Figure 3 makes the joint consistency-dynamics interpretation in Figure 4(a) visible. Across two prompts, TR keeps the cyclist's kit and the robot's green dress and sun hat recognizable as their poses and surroundings evolve over a minute. By contrast, LongLive's robot remains near its initial centered pose, while Rolling Forcing's cyclist moves partly out of view. The greater changes in appearance or framing in the Self Forcing and EMA-sink rows likewise show why a high raw dynamic score at lower consistency cannot, by itself, establish better motion. These frame strips complement the full videos, where the continuity of the action can be assessed.

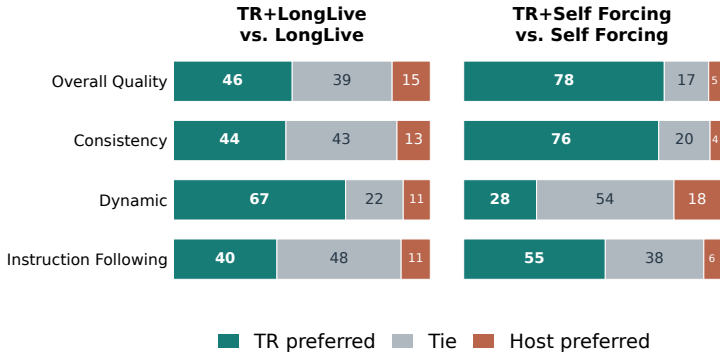

Figure 5: **Human preference on 60-second video pairs.** Bars show TR wins, ties, and host wins in rounded percentages. Exact counts appear in Table 4.

Human preferences further support this picture of improved consistency with sustained motion. Thirty-seven valid participants each rated 20 matched 60-second video pairs, yielding 740 paired trials. Figure 5 summarizes their judgments across four criteria; Appendix C gives the full counts and protocol.

5 Ablation Analysis

We study (i) the effects of varying the token numbers of the *Temporal Residual*'s global state, (ii) the influence of different layers where the state is injected, and (iii) the global memory update strategies to adapt TR to LongLive's prompt switch design. We vary TR's LongLive (Yang et al., 2025) version's config of state-token count, layer coverage, and the state policy at prompt switches around the default of one token in every block with Carry. Table 2 reports all six dimensions; Figure 4(b)

displays mean subject/background consistency against dynamic degree. As in the main comparison, the latter should be interpreted only alongside consistency and smoothness.

Table 2: **TR ablations on LongLive.** Memory strength and placement affect whether high consistency coexists with substantial dynamics. Scores are percentages. The reference uses one token per block and Carry. Blocks are zero-based; parameters count only TR. Bold marks non-dynamic column maxima, not an overall ranking.

Variant	Blocks	Params	Subject consist.	Backgr. consist.	Motion smooth.	Dynamic degree	Aesthetic quality	Imaging quality
Default	0–29	118.4M	97.92	96.61	98.85	52.85	61.34	69.01
State-token count								
4 tokens	0–29	118.4M	**99.27**	**97.98**	**99.55**	4.96	**62.02**	69.73
Layer coverage								
Last 20	10–29	79.0M	97.49	96.50	98.71	56.00	60.83	66.64
Last 10	20–29	39.5M	98.23	97.10	99.21	26.78	61.99	67.56
First 10	0–9	39.5M	98.12	96.93	99.12	25.83	61.72	**71.02**
State policy at a prompt switch								
Rebuild	0–29	118.4M	98.02	96.94	99.36	21.12	59.47	65.94

Increasing the number of state tokens does not simply increase useful memory. The four-token variant has the highest consistency scores, yet its dynamic degree falls to 4.96: stable appearance now comes with little visual evolution. The one-token default retains high consistency while leaving substantially more movement. Because the variants share projection and MLP weights, this change cannot be explained by extra trainable parameters; it reflects how repeated state entries participate in attention. A compact state can therefore be preferable to a stronger historical pull.

Where the state enters the DiT also changes the balance. Restricting TR to the last 20 blocks leaves slightly more measured motion but weakens consistency and visual quality; restricting it to only 10 blocks raises some consistency scores while sharply reducing dynamics. The response is not monotonic in layer count, so coverage cannot be treated as a simple strength knob. A single token distributed across all blocks gives the most useful joint behavior in these experiments. Early- and late-only placement also differ in imaging quality, suggesting that historical guidance affects rendering differently across the backbone.

At prompt switches, the difference between a persistent state and a reconstructed local summary becomes visible. Carry preserves the accumulated state while the host refreshes prompt-conditioned cache features; Rebuild resets it and reconstructs it from recent frames. Rebuild slightly raises consistency but substantially reduces dynamic degree and visual quality. This result supports the role of a history channel that survives a change of instruction instead of being recreated from whichever frames remain in the local window. Because each policy is used in both training and inference, the ablation compares complete adaptation recipes.

Together, our ablations support a restrained form of persistence: one distributed state token, advanced with committed content and carried across prompt changes, helps the host retain history while leaving local generation room to evolve.

6 Conclusion

Temporal Residual provides a lightweight route from bounded local context to evolving global memory conditioning. It gives bounded-context autoregressive video DiTs a compact history channel that evolves with committed content. A per-block state, dual-path injection, and chunk-synchronous updates extend access to past appearance without enlarging the host's local KV window over time. Brief post-training with LongLive and Self Forcing as base models demonstrates the benefits of the added *Temporal Residual*. The ablations explain why this joint objective matters: stronger memory can favor highly stable but nearly static generation, whereas compact conditioning distributed across the backbone supports a more useful balance.

AI USE STATEMENT

We used generative AI tools to learn relevant background concepts, polish the writing, and assist with code implementation. We also used LLMs to process the data used for the experiments. The authors take responsibility for the final text, code, and claims.

REPRODUCIBILITY STATEMENT

We describe the model architecture, training procedure, and evaluation protocol in the main paper and appendix. We will release the code publicly.

ETHICS STATEMENT

The method does not target a sensitive application. Our user study collected anonymous preferences on generated videos. The participants agreed to participate. As with other generative video systems, this technology could be misused to create misleading or harmful content.

REFERENCES

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

Guibin Chen, Dixuan Lin, Jiangping Yang, Chunze Lin, Junchen Zhu, Mingyuan Fan, Hao Zhang, Sheng Chen, et al. SkyReels-V2: Infinite-length Film Generative Model. *arXiv preprint arXiv:2504.13074*, 2025. URL https://arxiv.org/abs/2504.13074.

Justin Cui, Jie Wu, Ming Li, Tao Yang, Xiaojie Li, Rui Wang, Andrew Bai, Yuanhao Ban, et al. LoL: Longer than Longer, Scaling Video Generation to Hour. *arXiv preprint arXiv:2601.16914*, 2026. URL https://arxiv.org/abs/2601.16914.

Yusuf Dalva and Pinar Yanardag. AdaState: Self-Evolving Anchors for Streaming Video Generation. *arXiv preprint arXiv:2605.30349*, 2026. URL https://arxiv.org/abs/2605.30349.

Yu Gao, Haoyuan Guo, Tuyen Hoang, Weilin Huang, Lu Jiang, Fangyuan Kong, Huixia Li, Jiashi Li, et al. Seedance 1.0: Exploring the Boundaries of Video Generation Models. *arXiv preprint arXiv:2506.09113*, 2025. URL https://arxiv.org/abs/2506.09113.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2106.09685.

Qixin Hu, Shuai Yang, Wei Huang, Song Han, and Yukang Chen. LongLive-RAG: A General Retrieval-Augmented Framework for Long Video Generation. *arXiv preprint arXiv:2606.02553*, 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. In *Advances in Neural Information Processing Systems*, 2025a. URL https://arxiv.org/abs/2506.08009.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, et al. VBench: Comprehensive Benchmark Suite for Video Generative Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. URL https://arxiv.org/abs/2311.17982.

Ziqi Huang, Fan Zhang, Xiaojie Xu, Yinan He, Jiashuo Yu, Ziyue Dong, Qianli Ma, Nattapol Chanpaisit, et al. VBench++: Comprehensive and Versatile Benchmark Suite for Video Generative Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025b. doi: 10.1109/TPAMI.2025.3633890. URL https://doi.org/10.1109/TPAMI.2025.3633890.

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

Kunhao Liu, Wenbo Hu, Jiale Xu, Ying Shan, and Shijian Lu. Rolling Forcing: Autoregressive Long Video Diffusion in Real Time. *arXiv preprint arXiv:2509.25161*, 2025. URL https://arxiv.org/abs/2509.25161.

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

Yunhong Lu, Yanhong Zeng, Haobo Li, Hao Ouyang, Qiuyu Wang, Ka Leong Cheng, Jiapeng Zhu, Hengyuan Cao, Zhipeng Zhang, Xing Zhu, et al. Reward forcing: Efficient streaming video generation with rewarded distribution matching distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 34385–34397, 2026.

William Peebles and Saining Xie. Scalable Diffusion Models with Transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023. URL https://arxiv.org/abs/2212.09748.

Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Animesh Sinha, Ann Lee, Apoorv Vyas, Bowen Shi, et al. Movie Gen: A Cast of Media Foundation Models. *arXiv preprint arXiv:2410.13720*, 2024. URL https://arxiv.org/abs/2410.13720.

Seedance, Team, Heyi Chen, Siyan Chen, Xin Chen, Yanfei Chen, Ying Chen, Zhuo Chen, Feng Cheng, et al. Seedance 1.5 pro: A Native Audio-Visual Joint Generation Foundation Model. *arXiv preprint arXiv:2512.13507*, 2025. URL https://arxiv.org/abs/2512.13507.

Seedance, Team, De Chen, Liyang Chen, Xin Chen, Ying Chen, Zhuo Chen, Zhuowei Chen, Feng Cheng, et al. Seedance 2.0: Advancing Video Generation for World Complexity. *arXiv preprint arXiv:2604.14148*, 2026. URL https://arxiv.org/abs/2604.14148.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. RoFormer: Enhanced Transformer with Rotary Position Embedding. *arXiv preprint arXiv:2104.09864*, 2021. URL https://arxiv.org/abs/2104.09864.

Wan, Team, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, et al. Wan: Open and Advanced Large-Scale Video Generative Models. *arXiv preprint arXiv:2503.20314*, 2025. URL https://arxiv.org/abs/2503.20314.

Shuai Yang, Wei Huang, Ruihang Chu, Yicheng Xiao, Yuyang Zhao, Xianbang Wang, Muyang Li, Enze Xie, et al. LongLive: Real-time Interactive Long Video Generation. *arXiv preprint arXiv:2509.22622*, 2025. URL https://arxiv.org/abs/2509.22622.

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 *Advances in Neural Information Processing Systems*, 2024a. URL https://arxiv.org/abs/2405.14867.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T. Freeman, and Taesung Park. One-step Diffusion with Distribution Matching Distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024b. URL https://arxiv.org/abs/2311.18828.

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

Yifei Yu, Xiaoshan Wu, Xinting Hu, Tao Hu, Yangtian Sun, Xiaoyang Lyu, Bo Wang, Lin Ma, et al. VideoSSM: Autoregressive Long Video Generation with Hybrid State-Space Memory. *arXiv preprint arXiv:2512.04519*, 2025. URL https://arxiv.org/abs/2512.04519.

Lin Zhao, Yushu Wu, Yifan Gong, Yanzhi Wang, and Pu Zhao. OmniMem: Scalable and Adaptive Memory Retrieval for Long Video Generation. *arXiv preprint arXiv:2605.30519*, 2026. URL https://arxiv.org/abs/2605.30519.

Hongzhou Zhu, Min Zhao, Guande He, Hang Su, Chongxuan Li, and Jun Zhu. Causal Forcing: Autoregressive Diffusion Distillation Done Right for High-Quality Real-Time Interactive Video Generation. *arXiv preprint arXiv:2602.02214*, 2026. URL https://arxiv.org/abs/2602.02214.

A Implementation and Training Details

Table 3 details the two adaptation recipes. Both variants initialize from released host generators and train TR together with the indicated host parameters. TR has its own optimizer group and is excluded from LoRA. The teacher is frozen; the critic is fitted to generated samples. One generator update is performed for every five critic updates. Thus, the 2,000 iteration schedules correspond to 400 generator updates, under the stated update ratio.

Table 3: Training settings for the two TR instantiations. All frame counts in this table refer to latent frames.

Setting	TR+Self Forcing	TR+LongLive
Generator backbone	Wan2.1-T2V-1.3B	Wan2.1-T2V-1.3B
Teacher	Wan2.1-T2V-14B	Wan2.1-T2V-14B
Blocks / width	30 / 1536	30 / 1536
Chunk / denoising passes	3 / 4	3 / 4
Sink / local-frame budget	0 / 21	3 / 9
TR tokens / block	1	1
TR MLP bottleneck	256	256
Generator tuning	Full parameters	Rank-256 LoRA
TR tuning	Full parameters	Full parameters
Generator and TR learning rate	2×10^{-6}	10^{-5}
Critic learning rate	4×10^{-7}	2×10^{-6}
Iterations	2,000	2,000
Generator update interval	Every fifth iteration	Every fifth iteration
Generator EMA decay	0.99	0.99
Prompt regime	Single prompt per rollout	One switch per stream
Long-history regime	93–240 frames on half of iterations	Streams up to 240 frames
Supervised span	Last 21 frames on extended rollouts	New frames of one segment per generator update
Streaming segments	Extended self-rollouts	First 21 frames, then 18 new frames per segment
Prompt dataset	120,055 single prompts	120,055 prompt pairs

The nominal denoising timesteps are $\{1000, 750, 500, 250\}$. A shift factor of five maps a normalized time t to $5t/(1+4t)$, giving approximately $\{1000, 938, 833, 625\}$ on the 1,000-step scale. Training samples an exit pass and skips later passes, while inference uses all four. A timestep-zero forward pass then computes the emitted chunk's clean KV features using the updated state, with state writing disabled.

LongLive streams carry both cache and state between segments. A single prompt switch is sampled at a segment boundary from the source-switch prompt pairs. Carry is the default for this switch. For extended Self Forcing rollouts, the prefix supplies long-history conditioning without retaining its computation graph; only the final 21 latent frames are supervised. Earlier writes inside a differentiable span can be trained by later state reads in that same span.

LongLive adapts the host and critic with rank-256 LoRA, while TR is trained in full with a separate parameter group. The remaining optimization settings follow the LongLive official codebase.

B Resource Accounting and State Properties

B.1 Parameter and storage costs

For width $d = 1536$ and bottleneck $b = 256$, a modulation MLP with dimensions $2d \to b \to 6d$ and biases has

$$8db + b + 6d = 3{,}155{,}200 \tag{8}$$

parameters. The $d \to b \to d$ update MLP has $2db+b+d = 788{,}224$. The affine layer normalization, per-channel step size, and scalar attention scale add $2d + d + 1 = 4{,}609$, yielding 3,948,033 per block and 118,440,990 over 30 blocks. Shared tokenwise maps explain why four state tokens have

the same parameter count as one. Restricting TR to 20 or 10 blocks gives 78,960,660 or 39,480,330 parameters.

At two bytes per value, the default state occupies $2Ln_gd = 92{,}160$ bytes; four tokens occupy 368,640 bytes. For LongLive's cache, keys and values together require

$$2\ \text{(K,V)} \times L \times (n_{\text{sink}} + n_{\text{loc}}) \times P \times d \times 2\ \text{bytes} = 3{,}450{,}470{,}400\ \text{bytes}. \tag{9}$$

The recurrent-state/cache ratio is approximately 0.00267%. These are tensor-storage estimates using decimal KB, MB, and GB, not peak device-memory measurements. TR weights require 236,881,980 additional bytes in bfloat16; optimizer states and training activations incur separate costs.

B.2 Computation and initialization

For $F = 3$, $P = 1560$, and $n_g = 1$, the $T = 4680$ current tokens introduce $T + n_g(T + n_g) = 9361$ additional query–key pairs. A full $3 + 9$ frame window has $T \times 12P = 87{,}609{,}600$ pairs before TR, so the pair-count increase is approximately 0.0107%. This calculation excludes the modulation and update MLPs, projections, normalization, and implementation overhead.

Residual initialization removes the new value contributions but still adds attention entries. For a vision query with original softmax denominator Z, adding n_g zero-logit, zero-value keys scales its pre-projection output by $Z/(Z + n_g)$. Consequently, zero initialization is near the host when $Z \gg n_g$, rather than exactly function-preserving. This is distinct from the modulation and write branches, whose zero-initialized output layers produce exactly zero corrections.

B.3 Forward memory and backward credit

The additive recurrence gives $G_k = G_0 + \sum_{j=1}^{k} \Delta_j$. However, the nonlinear increments can alter or cancel previously stored content; this identity is not a retrieval guarantee. Writing $\Delta_k = f_\phi(\text{sg}(G_{k-1} + a_k \odot g_k))$, the backward rule induced by stop-gradient satisfies

$$\left.\frac{\partial G_k}{\partial G_{k-1}}\right|_{\text{autodiff}} = I, \qquad \frac{\partial \Delta_k}{\partial \phi} = \frac{\partial f_\phi(\text{sg}(G_{k-1} + a_k \odot g_k))}{\partial \phi}. \tag{10}$$

Thus an earlier write can receive credit through residual bases until a state detachment boundary. The true sensitivity of the forward map to a perturbation need not be the identity: stopping gradients changes differentiation, not the value computed by the forward recurrence. In particular, there is no guaranteed bound on the magnitude of G_k.

B.4 Interpreting multiple state tokens

Under identical zero initialization, shared tokenwise operations, and identical positional coordinates, all state-token rows are symmetric. Without a token-specific asymmetry, those operations preserve equal rows. Repeating a key/value pair n_g times changes its total attention mass from e^a to $n_g e^a$, even if the rows contain identical information. Accordingly, the four-token experiment tests the effect of more state entries, and cannot alone establish increased diversity of stored information.

C Evaluation and User Study

The evaluation uses common extended prompts. Thirty-seven valid anonymous participants each viewed 20 approximately 60-second, prompt-matched video pairs: 10 TR+LongLive versus LongLive and 10 TR+Self Forcing versus Self Forcing. Model identities were masked, and pair order and A/B placement were randomized. Each pair was rated separately for consistency, dynamics, instruction following, and overall quality, with a tie option. All 37 participants rated the 20 pairs, yielding 370 judgments per host comparison and criterion. Table 4 reports the specific counts. We define tie-adjusted TR preference as $(W + T/2)/(W + T + L)$, where W, T, and L are TR wins, ties, and host wins. These four judgments on the same pair are not independent observations.

Table 4: **Human preference in 60-second paired comparisons.** Thirty-seven participants rated ten video pairs per host comparison. The final column credits a tie as half a win for TR and is shown as a percentage.

Host	Criterion	TR wins	Ties	Host wins	TR preference
LongLive	Consistency	162	160	48	65.4
	Dynamic	248	83	39	78.2
	Instruction following	149	179	42	64.5
	Overall quality	171	144	55	65.7
Self Forcing	Consistency	281	74	15	85.9
	Dynamic	105	198	67	55.1
	Instruction following	204	142	24	74.3
	Overall quality	289	64	17	86.8