SWIFT: PROMPT-ADAPTIVE MEMORY FOR EFFICIENT INTERACTIVE LONG VIDEO GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Streaming long-video generation faces a central challenge in continuous semantic switching, requiring adaptive memory to preserve coherent visual evolution. Current approaches rely on cache rebuilding at prompt boundaries or fixed memory budgets, but they introduce redundant computation and limit flexible semantic adaptation. This limitation arises from a mismatch between cached video history and prompt updates, as memory preserves visual continuity while prompt switches demand rapid semantic adaptation. Motivated by this observation, we present **SWIFT** (Semantic Windowing and Injection for Flexible Transitions), a training-free framework for multi-prompt long-video generation that enables efficient semantic switching while preserving temporal coherence in causal video diffusion models. SWIFT introduces a lightweight *Semantic Injection Cache* that augments cached video memory rather than reconstructing it from scratch at every prompt boundary. To avoid uniformly perturbing all attention channels, we further perform head-wise semantic injection, so that each attention head receives a prompt update proportional to its alignment with the current video state. In addition, we introduce an *Adaptive Dynamic Window* that allocates temporal memory according to prompt phase, using larger local context near switching boundaries and smaller windows during stable segments to reduce average inference cost. To preserve long-range semantic consistency under compressed local attention, we further maintain segment-level semantic anchors that summarize prompt-conditioned video history and reintroduce it as compact memory tokens. Compared with current state-of-the-art methods, SWIFT preserves generation quality while achieving 22.6 FPS on a single H100 GPU, establishing a substantially more efficient solution for multi-prompt long-video generation. Our code is available at `https://anonymous.4open.science/r/SWIFT-5CBB`.

1 INTRODUCTION

Video generation delivers strong visual realism and temporal stability (Qing et al., 2024; Brooks et al., 2022; Yin et al., 2025b; He et al., 2022), with broad relevance to filmmaking (Yuan et al., 2024), digital content creation (Ju et al., 2025), and immersive media (Zhou et al., 2024a). Autoregressive video diffusion (Gu et al., 2025) extends this progress by rolling out frames sequentially for variable-length synthesis (Huang et al., 2025; Zhou et al., 2024b; Yin et al., 2024). Representative efforts include constant memory inference in FIFO-Diffusion (Kim et al., 2024), training-free long-horizon extension in LongDiff (Li et al., 2025), and supervised causal generation with autoregressive Transformers (Yin et al., 2025a). Beyond length extension, semantic control throughout generation is equally important, rather than only conditioning at initialization (Xie et al., 2025b; Feng et al., 2025).

Multi-prompt video generation is more challenging than single-prompt synthesis, as the model must adapt to prompt updates while ensuring visual coherence, stable transitions, and consistent dynamics. Existing memory-based pipelines struggle with balancing semantic adaptation and temporal preservation (Wu et al., 2025b; Liu et al., 2025). Recent studies have begun exploring multi-prompt long video generation through distinct control and memory paradigms (Cui et al., 2026; Wu et al., 2025a). DiTCtrl (Cai et al., 2025a) addresses sequential prompt control through attention level modulation, while LongLive (Yang et al., 2026) resolves prompt switching by recaching memory at semantic boundaries. SynCoS (Kim et al., 2025) extends pretrained text-to-video models for multi-event

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

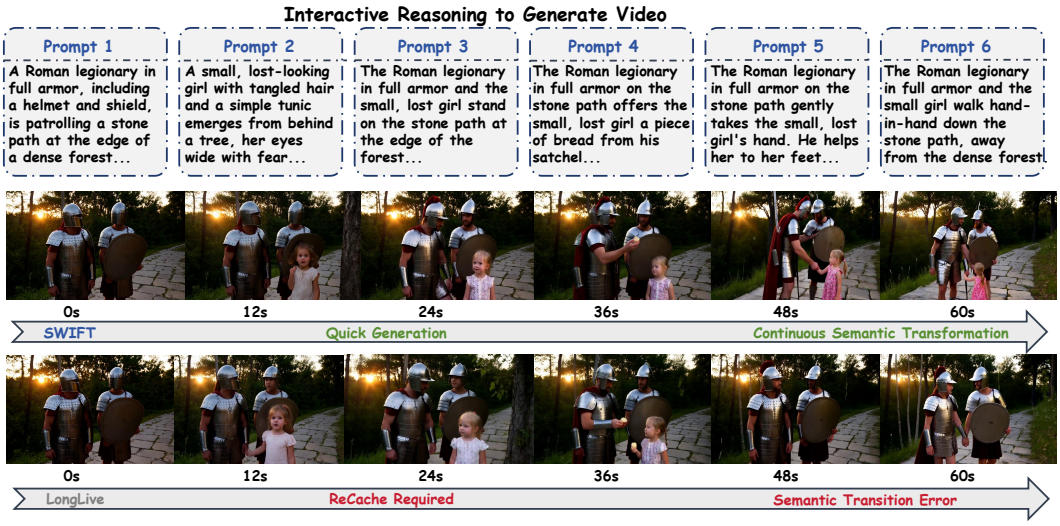

Figure 1: Streaming interactive T2V models, e.g., LongLive (Yang et al., 2026), suffer prompt-transition instability and recaching overhead. SWIFT uses Semantic Injection Cache and Adaptive Dynamic Window, improving transition smoothness, long-range consistency, and inference efficiency.

long video generation by synchronizing local smoothness and global coherence. However, limited responsiveness to newly activated prompts or memory reconstruction overhead still constrains their scalability in long-horizon interactive generation (Lin et al., 2026; Xi et al., 2025). It exposes a central requirement for multi-prompt long-video generation: ***an efficient memory mechanism must accommodate continuous prompt shifts while supporting coherent long-range generation.***

Motivated by these challenges, we propose **SWIFT** (Semantic Windowing and Injection for Flexible Transitions), an inference-time memory framework for multi-prompt long-video generation. SWIFT enables efficient adaptation to continuous prompt transitions while preserving coherent long-range visual evolution (Figure 1). SWIFT integrates two complementary components: (i) ***Semantic Injection Cache***, which injects transition-aware semantic signals into memory at prompt boundaries to improve adaptation without full cache reconstruction; and (ii) ***Adaptive Dynamic Window***, which allocates the temporal read budget according to the current generation phase to balance long-range coherence and inference efficiency. By combining Semantic Injection Cache with Adaptive Dynamic Window, SWIFT improves prompt responsiveness, visual coherence, and efficiency in multi-prompt long-video generation. The main contributions of this paper can be summarized in three parts.

- **New Perspective.** Multi-prompt long-video generation is studied from the perspective of continuous memory adaptation. Prompt switching is addressed through efficient historical-memory updating rather than repeated scratch reconstruction.
- **Unified Method.** We propose SWIFT, an inference-time memory framework for multi-prompt long-video generation. SWIFT couples transition-aware semantic memory refinement with prompt-phase adaptive temporal budgeting, enabling efficient multi-prompt long-video generation with strong controllability and coherence.
- **Empirical Validation.** Extensive experiments are conducted under multi-prompt long-video generation settings. SWIFT achieves strong visual quality and semantic alignment while reaching 22.6 FPS on a single H100 GPU under the multi-prompt setting.

2 Related Works

Long Video Generation. Recent progress in video generation (Lee et al., 2024) has extended synthesis from short clips to longer and more coherent sequences through autoregressive diffusion (Yin et al., 2025a; Gao et al., 2025), diffusion forcing (Song et al., 2025), and memory-based streaming inference (Zhu et al., 2025). Autoregressive and hybrid autoregressive-diffusion methods

generate long videos by iteratively rolling out future chunks while reusing historical context (Xie et al., 2025a), achieving strong temporal continuity and efficient causal generation. Diffusion-forcing methods improve long-horizon rollout by aligning training and inference under partially generated contexts. VideoTetris (Tian et al., 2024) enables compositional text-to-video generation by spatially and temporally composing attention maps to better follow complex textual semantics.

Interactive and Multi-Prompt Video Generation. Interactive and multi-prompt video generation (Cai et al., 2025a; Wu et al., 2025b) requires the model to adapt to changing text conditions while preserving visual continuity across prompt boundaries. Existing approaches commonly decompose the task into multiple short clips (Villegas et al., 2022) and then apply stitching, interpolation, or segment-wise generation, which often weakens temporal coherence over long horizons. StoryMem (Zhang et al., 2025a) further explores multi-shot long video storytelling by maintaining an explicit visual memory bank to improve cross-shot consistency across sequential video segments.

Memory-Efficient Inference for Long-Horizon Generation. To improve the efficiency of long-horizon inference, existing methods commonly restrict attention to local windows (Xie et al., 2025a; Yang et al., 2026) or compact memory states (Cai et al., 2025b) instead of reading the full history densely. These designs provide an effective trade-off between efficiency and long-range context, although most of them rely on fixed memory budgets across different generation stages. SWIFT addresses this challenge with a continuous and dynamic memory mechanism for semantic transition and temporal memory allocation.

3 METHOD

SWIFT (Semantic Windowing and Injection for Flexible Transitions) is an inference-time mechanism for efficient semantic transition and memory cache management in interactive long-video generation under sequential prompt control. Built upon causal autoregressive video diffusion, the method combines two coupled designs: a *Semantic Injection Cache* (section 3.1) for memory updates at prompt transitions via lightweight semantic bridging rather than expensive cache reconstruction, and an *Adaptive Dynamic Window* (section 3.2) for phase-aware control of the effective temporal memory span with segment-level semantic anchors augmenting compressed local context. These designs preserve temporal continuity, improve semantic responsiveness at prompt boundaries, and reduce the average attention cost during long-horizon rollout.

3.1 SEMANTIC INJECTION CACHE

Prompt transitions introduce a mismatch between the newly activated prompt semantics and the historical video memory accumulated under earlier prompts (Yu et al., 2025). Existing methods typically address this issue by recomputing the cache at prompt boundaries (Yang et al., 2026), which improves semantic alignment but incurs substantial additional computation. We introduce a Semantic Injection Cache (Figure 2) that updates memory through the detected semantic transition, enabling prompt adaptation without recomputing the full cache while preserving the visual history accumulated in prior memory states.

Prompt Transition Signatures. For each prompt segment, we first compute a projected prompt signature by aggregating the prompt embeddings after projection into the generator hidden space. Let $p^{(m-1)}$ and $p^{(m)}$ denote the projected signatures of the previous and current prompts, represented in the same head-wise value-cache space as the cached visual summaries. We define the prompt transition signal as:

$$\Delta p^{(m)} = p^{(m)} - p^{(m-1)}. \tag{1}$$

This difference vector captures the semantic displacement induced by the prompt switch. We further quantify the switch magnitude through the cosine-based transition strength

$$\rho^{(m)} = 1 - \cos\Big(p^{(m-1)},\, p^{(m)}\Big), \tag{2}$$

which is later used to modulate the amount of semantic injection. We decompose $\Delta p^{(m)}$ along the local cache tangent m. Under a local first-order cache response model, the closest motion-neutral update is the following orthogonal projection.

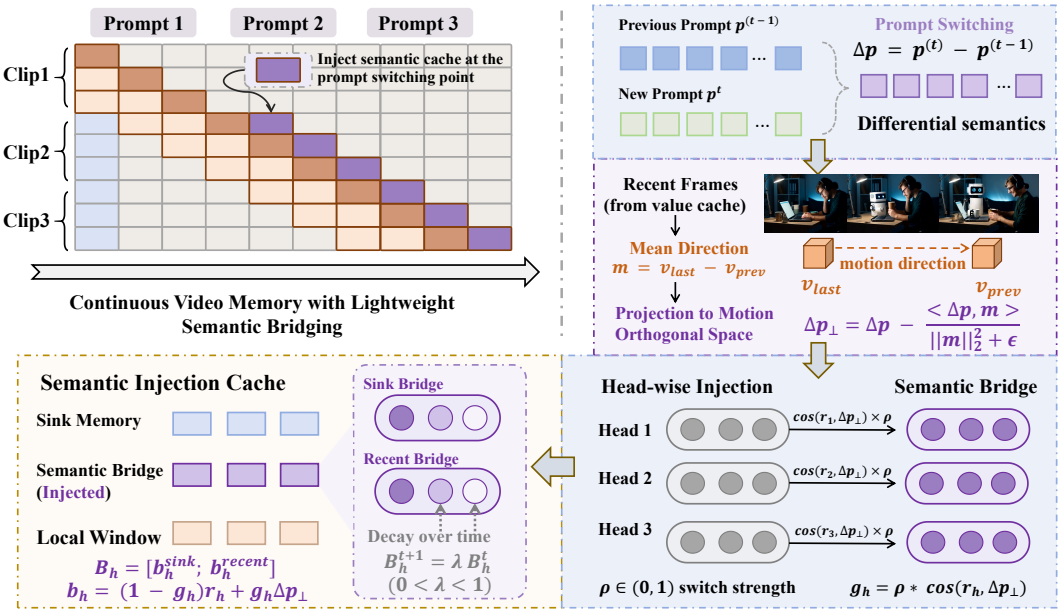

Figure 2: Illustration of **Semantic Injection Cache**. Instead of rebuilding the full video cache at every prompt boundary, SWIFT constructs a lightweight semantic bridge from the prompt transition signal. The transition is first projected onto a motion-orthogonal subspace to avoid interfering with local temporal dynamics, and is then injected into memory through head-wise alignment with recent and sink summaries. The injected bridge is read together with continuous video memory, providing efficient semantic switching while preserving motion continuity.

Proposition 1 (**Motion-neutral semantic projection**). *Let $\mathcal{H}$ be the value-cache representation space, let $\Delta p^{(m)} = p^{(m)} - p^{(m-1)}$ be the prompt displacement, and let $m = v_t - v_{t-1}$ be the local cache tangent. The minimum-distortion update orthogonal to m is*

$$\Delta p_\perp^{(m)} = \Delta p^{(m)} - \frac{\langle \Delta p^{(m)}, m \rangle}{\|m\|_2^2} m. \tag{3}$$

The derivation is given in Appendix D. Thus, semantic injection removes the motion-aligned component while preserving the largest first-order motion-neutral prompt signal. In practice, we use the stabilized form (where $\epsilon > 0$ prevents numerical amplification when the estimated motion magnitude is small).

$$\Delta p_\perp^{(m)} = \Delta p^{(m)} - \frac{\langle \Delta p^{(m)}, m \rangle}{\|m\|_2^2 + \epsilon} m. \tag{4}$$

Head-wise Semantic Injection. We construct the semantic bridge head-wise to reflect heterogeneous memory reading patterns across attention heads, from local dynamics to long-range contextual anchors. For each layer, we extract two summaries from the value cache: a recent summary r from the most recent frames and a sink summary s from the persistent sink region. We measure their alignment with the projected semantic transition by clipped cosine similarity and use the switch strength $\rho^{(m)}$ to define bounded head-wise gates:

$$g_r = \rho^{(m)} \left[\cos\left(r, \Delta p_\perp^{(m)}\right) \right]_0^1, \qquad g_s = \rho^{(m)} \left[\cos\left(s, \Delta p_\perp^{(m)}\right) \right]_0^1. \tag{5}$$

Here, $[\cdot]_0^1$ denotes clipping to $[0, 1]$, so g_r and g_s act as bounded interpolation gates that control the semantic transition injected into the recent and sink-aligned summaries. We then construct two bridge components:

$$B_r = (1 - g_r)\, r + g_r\, \Delta p_\perp^{(m)}, \qquad B_s = (1 - g_s)\, s + g_s\, \Delta p_\perp^{(m)}. \tag{6}$$

The two components are concatenated as a transient bridge memory $B_h = [B_s; B_r]$. Following the implementation, each bridge representation is instantiated as a paired key–value entry, $K_h^{\text{bridge}} =$

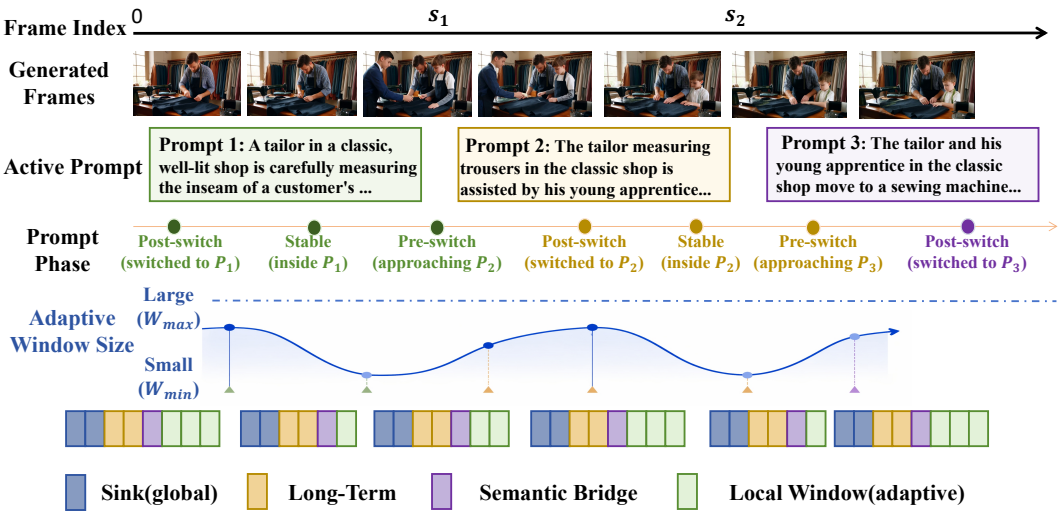

Figure 3: Illustration of **Adaptive Dynamic Window**. SWIFT allocates temporal memory according to prompt phase rather than using a fixed local attention span throughout generation. The effective window expands around prompt transitions for stable semantic handover and shrinks inside stable intervals for efficient rollout. Segment-level semantic anchors compensate for the reduced local context by preserving compact prompt-conditioned summaries of previous segments, thereby lowering average attention cost without sacrificing long-range coherence.

$V_h^{\text{bridge}} = B_h$, and appended to the attention cache. During self-attention, these entries are read together with sink memory, segment anchors, and the adaptive local window, providing prompt-transition guidance without rebuilding the full historical cache. After each generated block, the bridge contribution decays as $B_h^{t+1} = \lambda B_h^t$, allowing newly generated video evidence to gradually dominate once the transition has been absorbed.

3.2 Adaptive Dynamic Window

In long-video generation, the inference cost of causal attention increases with sequence length (Wang et al., 2025; Xi et al., 2025), while recent frames usually contribute more to the next prediction than distant history. Existing local-attention methods therefore use a fixed temporal window for efficiency (Zhang et al., 2025c; Yang et al., 2026), but different inference phases favor different window sizes. To address this issue, an adaptive dynamic window (Figure 3) is introduced to allocate different temporal read budgets across multi-prompt generation phases, improving efficiency in semantically stable intervals and preserving sufficient context around prompt transitions.

Prompt-Phase Window Scheduling. Let t denote the current block position, let s_m denote the start of the active prompt segment, and let s_{m+1} denote the next switch boundary. We define the segment age

$$a_t = \max(0,\, t - s_m), \tag{7}$$

and the distance to the next switch

$$d_t = \begin{cases} \max(0,\, s_{m+1} - t), & \text{if the next switch exists,} \\ +\infty, & \text{otherwise.} \end{cases} \tag{8}$$

The prompt phase of the current generation step is determined by a_t and d_t. Early after a prompt switch, the model requires a larger local window to maintain stable semantic transition. In the stable portion of a prompt segment, the local context demand becomes weaker and the window can be reduced. As generation approaches the next switch, the window expands again to prepare for the upcoming semantic transition. To realize this phase-dependent scheduling, we define a post-switch decay factor and a pre-switch expansion factor

$$w_{\text{post}}(t) = \exp\left(-\frac{a_t}{\tau_{\text{post}}}\right), \quad w_{\text{pre}}(t) = \exp\left(-\frac{d_t}{\tau_{\text{pre}}}\right). \tag{9}$$

The effective phase weight is

$$w_t = \max\{w_{\text{post}}(t),\ w_{\text{pre}}(t)\}\,. \tag{10}$$

Given a maximum window size $W_{\max}$ and a minimum window size $W_{\min}$, the adaptive window is defined as

$$W_t = W_{\min} + (W_{\max} - W_{\min})\, w_t. \tag{11}$$

The physical cache budget remains fixed, and the adaptive mechanism only modulates the effective temporal read range at each step. This design avoids cache reallocation overhead while reducing the average attention cost over long-horizon generation.

Segment-Level Semantic Anchors. A small local window improves efficiency, but it also reduces direct access to long-range semantic context (Zhang et al., 2025c). To compensate for this loss, we introduce segment-level semantic anchors. At the end of each prompt segment, we summarize the recent value cache and combine this visual summary with the projected prompt signature of the same segment. Let $u^{(m)}$ denote the summary of the recent value cache from the completed segment, and let $p^{(m)}$ denote the semantic prompt signature obtained by projecting the prompt into the generator hidden space. We define the segment anchor as

$$A^{(m)} = (1-\alpha)\, u^{(m)} + \alpha\, p^{(m)}, \tag{12}$$

where α controls the balance between visual memory and semantic identity. This anchor provides a compact prompt-conditioned summary of the segment and is injected into attention together with sink memory, semantic bridge entries, and the current local window. Long-range semantic context is therefore preserved without incurring the full cost of dense historical attention.

Unified Memory View. Under the proposed design, the memory used by the generator is no longer a single fixed-size local cache. Instead, it becomes a structured memory system composed of continuous video history, transient semantic bridges, compact segment anchors, and a prompt-phase adaptive local window. The semantic injection cache governs how the model changes its interpretation of history at prompt transitions. The adaptive dynamic window governs where computation is spent over time. The segment anchors preserve long-range semantic context when the local window is compressed. Together, these components enable efficient and coherent multi-prompt long-video generation without relying on repeated full memory reconstruction. The detailed procedures for semantic injection and adaptive window scheduling are summarized in Algorithm 1 and Algorithm 2.

4 Experiments

4.1 Experimental Setup

Implementation Details. SWIFT is instantiated on top of Wan2.1-T2V-1.3B (Wan et al., 2025), inherits the inference protocol of LongLive (Yang et al., 2026), and extends the backbone with our semantic injection and adaptive memory design. SWIFT is compatible with autoregressive video diffusion models that expose reusable attention memory value cache and support blockwise causal rollout.

Evaluation protocol. Following the same setting as LongLive (Yang et al., 2026) and Self-Forcing (Huang et al., 2025), we use Qwen2-72B-Instruct (Yang et al., 2024) to generate the switched-prompt dataset. Each script set contains 6 consecutive 10-second cue sequences, for a total of 100 videos. Each video lasts 60 seconds. We use VBench-Long (Huang et al., 2024) to evaluate the visual quality of all generated videos, focusing on image quality, subject consistency and aesthetics, to compare long-range consistency and visual effects in terms of subject, background and visual aesthetics. For semantic alignment evaluation under sequential prompt control, each generated video is partitioned into 10-second segments according to the prompt schedule, and the CLIP (Radford et al., 2021) score is computed between each segment and its corresponding prompt.

Baselines. We compare SWIFT with several representative state-of-the-art long video generation methods under prompt switching at inference time. The compared methods include MemFlow (Ji et al., 2025), FramePack (Zhang et al., 2025b), Self Forcing (Huang et al., 2025), CausVid (Yin et al., 2025a), and LongLive (Yang et al., 2026). Each script contains 6 consecutive 10-second prompts, yielding 100 generated videos totaling 60 seconds each. Appendix C provides detailed information on SWIFT settings.

Table 1: **CLIP score and FPS comparison under the multi-prompt 60-second setting.** CLIP scores are computed at prompt-aligned intervals.

Method	#Params	Throughput (FPS)↑	CLIP Score↑					
			0–10 s	10–20 s	20–30 s	30–40 s	40–50 s	50–60 s
MemFlow (Ji et al., 2025)	1.3B	18.7	26.29	24.64	23.85	23.53	24.27	<u>23.73</u>
Self Forcing (Huang et al., 2025)	1.3B	17.0	26.24	24.87	23.46	21.92	22.05	21.07
LongLive (Yang et al., 2026)	1.3B	<u>20.7</u>	**26.63**	<u>25.47</u>	<u>24.65</u>	<u>23.69</u>	<u>24.52</u>	23.61
CausVid (Yin et al., 2025a)	1.3B	17.0	26.55	24.93	23.82	22.74	23.16	22.68
FramePack (Zhang et al., 2025b)	13B	6.7	26.51	22.60	22.18	21.53	21.98	21.62
SWIFT	1.3B	**22.6**	<u>26.53</u>	**25.86**	**24.71**	**24.04**	**24.68**	**24.13**

Table 2: **Quantitative comparison under multi-prompt 60-second setting** with representative long video generation models.

Method	Resolution	Aesthetic Quality↑	Background Consistency↑	Imaging Quality↑	Motion Smoothness↑	Subject Consistency↑	Temporal Flickering↑
MemFlow (Ji et al., 2025)	832×480	<u>60.02</u>	96.21	71.69	98.83	<u>97.63</u>	97.67
Self Forcing (Huang et al., 2025)	832×480	58.45	95.74	71.33	98.56	95.35	97.52
LongLive (Yang et al., 2026)	832×480	59.89	96.05	<u>71.74</u>	<u>99.14</u>	97.32	**98.46**
CausVid (Yin et al., 2025a)	832×480	59.77	<u>95.42</u>	71.73	99.02	96.15	98.02
FramePack (Zhang et al., 2025b)	832×480	59.68	**96.77**	71.71	98.87	**97.86**	97.41
SWIFT	832×480	**60.23**	<u>96.28</u>	**72.32**	**99.19**	97.34	<u>98.35</u>

4.2 Multi-prompt Generation Results

We first compare SWIFT with representative existing methods under the multi-prompt setting to evaluate its performance on long-horizon interactive video generation.

Text-video alignment and efficiency. Table 1 further reports clip-wise CLIP scores and inference throughput. SWIFT achieves the highest CLIP scores in five out of six prompt intervals and remains very close to the best result in the first interval, showing consistently stronger semantic alignment immediately after prompt transitions. The semantic injection cache enables SWIFT to reliably sustain strong prompt alignment throughout successive prompt changes by continuously correcting residual semantics in historical memory. At the same time, SWIFT reaches the highest throughput of 22.6 FPS, outperforming LongLive at 20.7 FPS and substantially exceeding the other baselines.

Video quality. Visual quality is comprehensively evaluated using multiple metrics from VBench (Huang et al., 2024), including aesthetic quality, background consistency, imaging quality, motion smoothness, subject consistency, and temporal flickering. These metrics provide a broad assessment of visual fidelity, long range coherence, and temporal stability in generated videos. As shown in Table 2, SWIFT delivers strong overall visual quality under the multi-prompt setting. It achieves the best results in aesthetic quality, imaging quality, and motion smoothness, while remaining competitive in background consistency, subject consistency, and temporal flickering. These results indicate that SWIFT improves visual fidelity and temporal stability without compromising long-range consistency. We attribute this improvement to the proposed memory design, which facilitates semantic transitions while retaining useful historical context. The results demonstrate that SWIFT improves prompt-following ability under sequential prompt control while also providing higher inference efficiency.

4.3 Ablation Studies

We conduct ablation studies on the two core components of the framework, namely the Semantic Injection Cache and the Adaptive Dynamic Window, under the 60-second interactive multi-prompt video generation setting with six consecutive prompts.

Memory Variants. Specifically, variants with different memory mechanisms are constructed based on the design in Section 3.1. *No-Memory* removes the proposed memory augmentation and performs generation without additional transition-aware memory. *Sink* retains only the sink memory for

Table 3: **Ablation study under the multi-prompt 60-second setting**. The CLIP score is computed at intervals aligned with the prompt switching.

Variant	Quality Score↑	Consistency Score↑	Throughput (FPS)↑	CLIP Score↑					
				0–10 s	10–20 s	20–30 s	30–40 s	40–50 s	50–60 s
No-Memory	80.94	90.87	**25.1**	24.83	22.74	21.68	21.35	21.12	20.94
Sink	82.19	92.52	<u>24.5</u>	25.58	23.40	22.45	22.67	22.59	22.82
Sink+SIC	<u>84.26</u>	<u>95.74</u>	20.5	26.11	<u>25.12</u>	<u>23.98</u>	<u>23.77</u>	<u>23.95</u>	<u>23.84</u>
Fixed	83.71	94.96	20.9	**26.67**	24.96	23.81	23.58	23.64	23.42
SWIFT	**84.97**	**96.62**	22.6	<u>26.53</u>	**25.86**	**24.71**	**24.04**	**24.68**	**24.13**

long-range context preservation. *Sink+SIC* further introduces the Semantic Injection Cache on top of sink memory to support semantic adaptation across prompt transitions.

Window Variants. Following the design in Section 3.2, we further compare different cache capacity strategies. Here, *Fixed* uses a constant static local window with size 12 throughout the entire generation process, without adaptive adjustment across prompt phases. Empirical results across all variants collectively underscore the necessity of our integrated memory design in effectively mitigating semantic drift while preserving long-range structural integrity.

Table 3 reports inference throughput together with clip-wise CLIP scores over all six prompt intervals. The full SWIFT achieves the strongest overall semantic alignment performance, which further verifies the effectiveness of combining Semantic Injection Cache with adaptive temporal memory allocation for sustained prompt following under long-horizon multi-prompt generation. High throughput in *No-Memory* and *Sink* is offset by significant degradation in video quality and consistency metrics. Such performance gaps underscore the insufficiency of raw efficiency for sustaining coherent multi-prompt generation over long horizons. Meanwhile, *Sink+SIC* yields performance competitive with SWIFT in visual metrics but exhibits a trade-off through noticeably lower inference throughput. Compared with *Fixed*, SWIFT achieves higher throughput and better overall generation quality, showing the advantage of adaptive temporal memory allocation over a constant window strategy.

As shown in Figure 4, across the visualized frame sequences, the full SWIFT produces more coherent semantic transitions and more stable long-range visual continuation than the ablated variants. This highlights the importance of coupling semantic memory refinement with phase-aware temporal allocation in multi-prompt long-video generation.

4.4 Sensitivity Analysis

Long-horizon video generation requires the model to balance temporal coherence and computational efficiency under constrained attention budgets Li et al. (2025; 2026). Sensitivity analyses are conducted on key hyperparameters of both the Semantic Injection Cache and the Adaptive Dynamic Window, described in Section 3.1 and Section 3.2, respectively. All other hyperparameters remain the same as the default settings. Figure 5 illustrates the sensitivity of representative visual quality metrics to adaptive window hyperparameter settings, including the minimum window size and the phase scheduling factors.

Minimum window size $W_{\min}$. $W_{\min}$ controls the lower bound of the temporal read budget. Very small values degrade quality, whereas excessively large values reduce the efficiency gain with limited additional benefit.

Phase scheduling factors τ_{post} and τ_{pre}. The parameters τ_{post} and τ_{pre} control how fast the temporal window shrinks after a prompt switch and how early it expands before the next boundary. Across a broad range of τ_{post} and τ_{pre} values, the visual quality metrics remain largely stable, indicating that SWIFT is robust to moderate changes in phase scheduling. The result indicates that effective multi-prompt generation can be maintained without highly sensitive hyperparameter tuning.

Semantic mixing coefficient α. α balances visual memory and semantic identity in the segment anchor. The quality metrics remain stable over the tested range, motivating the default value and confirming a broad operating range.



Figure 4: **Qualitative comparison of different memory variants under the multi-prompt 60-second setting.** *Fixed* uses a constant local window. *No-Memory* removes additional transition-aware memory. *Sink* retains only sink memory. *Sink+SIC* adds the Semantic Injection Cache on top of sink memory. *Ours* denotes the full SWIFT model, which achieves more coherent prompt transitions and better long-range visual consistency.

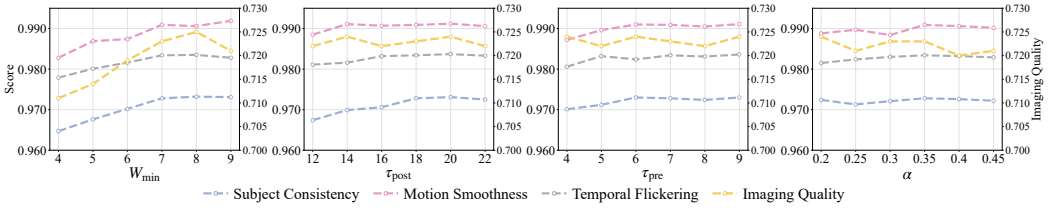

Figure 5: **Sensitivity analysis of the adaptive dynamic window.** The figure reports the variation of representative visual quality metrics under different hyperparameter settings of the adaptive window, including the minimum window size and the phase scheduling factors.

4.5 Additional Experiments

We further analyze temporal injection patterns in Appendix E.1, including one-shot, continuously decayed, and constant injection. Per-block efficiency and scalability across different video lengths are examined in Appendix E.2 and Appendix E.3, respectively. Finally, Appendix E.4 reports a user study on transition consistency, prompt following, and aesthetic quality.

5 Conclusion

In this work, we introduce SWIFT, a training-free memory framework for efficient multi prompt long video generation. SWIFT enables prompt transitions through lightweight semantic injection instead of repeated cache reconstruction. Semantic Injection Cache updates historical video memory with transition-aware prompt signals and preserves motion continuity and scene structure. Adaptive Dynamic Window allocates temporal memory according to prompt phase, using larger context around prompt switches and compact context during stable generation. Segment-level semantic anchors preserve long range prompt conditioned history under reduced local attention. The proposed design avoids expensive full memory rebuilding and provides a more practical path toward scalable interactive long video generation. Extensive experiments show that SWIFT improves prompt responsiveness, temporal coherence, and visual quality under sequential prompt control, and achieves 22.6 FPS on a single H100 GPU in the multi-prompt setting.

REFERENCES

Tim Brooks, Janne Hellsten, Miika Aittala, Ting-Chun Wang, Timo Aila, Jaakko Lehtinen, Ming-Yu Liu, Alexei Efros, and Tero Karras. Generating long videos of dynamic scenes. *Advances in Neural Information Processing Systems*, 35:31769–31781, 2022.

Minghong Cai, Xiaodong Cun, Xiaoyu Li, Wenze Liu, Zhaoyang Zhang, Yong Zhang, Ying Shan, and Xiangyu Yue. Ditctrl: Exploring attention control in multi-modal diffusion transformer for tuning-free multi-prompt longer video generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 7763–7772, 2025a.

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. *arXiv preprint arXiv:2508.21058*, 2025b.

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 *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=DzvPiqh23f.

Weixi Feng, Chao Liu, Sifei Liu, William Yang Wang, Arash Vahdat, and Weili Nie. Blobgen-vid: Compositional text-to-video generation with blob video representations. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 12989–12998, 2025.

Kaifeng Gao, Jiaxin Shi, Hanwang Zhang, Chunping Wang, Jun Xiao, and Long Chen. Ca2-vdm: Efficient autoregressive video diffusion model with causal generation and cache sharing. In *International Conference on Machine Learning*, pp. 18550–18565. PMLR, 2025.

Yuchao Gu, Weijia Mao, and Mike Zheng Shou. Long-context autoregressive video modeling with next-frame prediction. *arXiv preprint arXiv:2503.19325*, 2025.

Yingqing He, Tianyu Yang, Yong Zhang, Ying Shan, and Qifeng Chen. Latent video diffusion models for high-fidelity long video generation. *arXiv preprint arXiv:2211.13221*, 2022.

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. *arXiv preprint arXiv:2506.08009*, 2025.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. Vbench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21807–21818, 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 preprint arXiv:2512.14699*, 2025.

Xuan Ju, Weicai Ye, Quande Liu, Qiulin Wang, Xintao Wang, Pengfei Wan, Di Zhang, Kun Gai, and Qiang Xu. Fulldit: Multi-task video generative foundation model with full attention. *arXiv preprint arXiv:2503.19907*, 2025.

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

Subin Kim, Seoung Wug Oh, Jui-Hsien Wang, Joon-Young Lee, and Jinwoo Shin. Tuning-free multi-event long video generation via synchronized coupled sampling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6418–6429, 2025.

Taegyeong Lee, Soyeong Kwon, and Taehwan Kim. Grid diffusion models for text-to-video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8734–8743, 2024.

Ruibin Li, Tao Yang, Fangzhou Ai, Tianhe Wu, Shilei Wen, Bingyue Peng, and Lei Zhang. Long-horizon streaming video generation via hybrid attention with decoupled distillation. *arXiv preprint arXiv:2604.10103*, 2026.

Zhuoling Li, Hossein Rahmani, Qiuhong Ke, and Jun Liu. Longdiff: Training-free long video generation in one go. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17789–17798, 2025.

Shanchuan Lin, Ceyuan Yang, Hao He, Jianwen Jiang, Yuxi Ren, Xin Xia, Yang Zhao, Xuefeng Xiao, and Lu Jiang. Autoregressive adversarial post-training for real-time interactive video generation. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026.

Zhiheng Liu, Xueqing Deng, Shoufa Chen, Angtian Wang, Qiushan Guo, Mingfei Han, Zeyue Xue, Mengzhao Chen, Ping Luo, and Linjie Yang. Worldweaver: Generating long-horizon video worlds via rich perception. *arXiv preprint arXiv:2508.15720*, 2025.

Zhiwu Qing, Shiwei Zhang, Jiayu Wang, Xiang Wang, Yujie Wei, Yingya Zhang, Changxin Gao, and Nong Sang. Hierarchical spatio-temporal decoupling for text-to-video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6635–6645, 2024.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Kiwhan Song, Boyuan Chen, Max Simchowitz, Yilun Du, Russ Tedrake, and Vincent Sitzmann. History-guided video diffusion. *arXiv preprint arXiv:2502.06764*, 2025.

Ye Tian, Ling Yang, Haotian Yang, Yuan Gao, Yufan Deng, Jingmin Chen, Xintao Wang, Zhaochen Yu, Xin Tao, Pengfei Wan, et al. Videotetris: Towards compositional text-to-video generation. *Advances in Neural Information Processing Systems*, 37:29489–29513, 2024.

Ruben Villegas, Mohammad Babaeizadeh, Pieter-Jan Kindermans, Hernan Moraldo, Han Zhang, Mohammad Taghi Saffar, Santiago Castro, Julius Kunze, and Dumitru Erhan. Phenaki: Variable length video generation from open domain textual description. *arXiv preprint arXiv:2210.02399*, 2022.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Hongjie Wang, Chih-Yao Ma, Yen-Cheng Liu, Ji Hou, Tao Xu, Jialiang Wang, Felix Juefei-Xu, Yaqiao Luo, Peizhao Zhang, Tingbo Hou, et al. Lingen: Towards high-resolution minute-length text-to-video generation with linear computational complexity. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 2578–2588, 2025.

Tong Wu, Shuai Yang, Ryan Po, Yinghao Xu, Ziwei Liu, Dahua Lin, and Gordon Wetzstein. Video world models with long-term spatial memory. *arXiv preprint arXiv:2506.05284*, 2025a.

Ziyi Wu, Aliaksandr Siarohin, Willi Menapace, Ivan Skorokhodov, Yuwei Fang, Varnith Chordia, Igor Gilitschenski, and Sergey Tulyakov. Mind the time: Temporally-controlled multi-event video generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 23989–24000, 2025b.

Haocheng Xi, Shuo Yang, Yilong Zhao, Chenfeng Xu, Muyang Li, Xiuyu Li, Yujun Lin, Han Cai, Jintao Zhang, Dacheng Li, et al. Sparse video-gen: Accelerating video diffusion transformers with spatial-temporal sparsity. In *International Conference on Machine Learning*, pp. 68208–68224. PMLR, 2025.

Desai Xie, Zhan Xu, Yicong Hong, Hao Tan, Difan Liu, Feng Liu, Arie Kaufman, and Yang Zhou. Progressive autoregressive video diffusion models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 6322–6332, 2025a.

Fan Xie, Dan Zeng, Qiaomu Shen, and Bo Tang. A comprehensive survey on text-to-video generation. *Chinese Journal of Electronics*, 34(4):1009–1036, 2025b.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. Qwen2 technical report, 2024. URL https://arxiv.org/abs/2407.10671.

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

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*, pp. 6613–6623, 2024.

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

Zhiyu Yin, Kehai Chen, Xuefeng Bai, Ruili Jiang, Juntao Li, Hongdong Li, Jin Liu, Yang Xiang, Jun Yu, and Min Zhang. A survey: spatiotemporal consistency in video generation. *ACM Computing Surveys*, 2025b.

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

Zhengqing Yuan, Yixin Liu, Yihan Cao, Weixiang Sun, Haolong Jia, Ruoxi Chen, Zhaoxu Li, Bin Lin, Li Yuan, Lifang He, et al. Mora: Enabling generalist video generation via a multi-agent framework. *arXiv preprint arXiv:2403.13248*, 2024.

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 preprint arXiv:2512.19539*, 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.

Peiyuan Zhang, Yongqi Chen, Runlong Su, Hangliang Ding, Ion Stoica, Zhengzhong Liu, and Hao Zhang. Fast video generation with sliding tile attention. In *International Conference on Machine Learning*, pp. 74714–74731. PMLR, 2025c.

Shangchen Zhou, Peiqing Yang, Jianyi Wang, Yihang Luo, and Chen Change Loy. Upscale-a-video: Temporal-consistent diffusion model for real-world video super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2535–2545, 2024a.

Yupeng Zhou, Daquan Zhou, Ming-Ming Cheng, Jiashi Feng, and Qibin Hou. Storydiffusion: Consistent self-attention for long-range image and video generation. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024b. URL https://openreview.net/forum?id=VFqzxhINFU.

Tianrui Zhu, Shiyi Zhang, Zhirui Sun, Jingqi Tian, and Yansong Tang. Memorize-and-generate: Towards long-term consistency in real-time video generation. *arXiv preprint arXiv:2512.18741*, 2025.

Appendix

A Use of Large Language Models **14**

B Broader Impacts **14**

C Detailed Experimental Setup **14**

C.1 Compute Resources 14

C.2 SWIFT Settings 14

C.3 Baselines 15

D Justification of Proposition 1 ..... **15**

D.1 Setup and Assumptions 16

D.2 Proof 16

D.3 Stabilized Implementation 18

E Further Experiments **18**

E.1 Temporal Injection Patterns 18

E.2 Per Block Efficiency Trace 18

E.3 Multi-Prompt Generation with Different Video Lengths 19

E.4 User Study 20

E.5 Case Study 21

F Limitation Analysis **21**

A USE OF LARGE LANGUAGE MODELS

Large language models were used only as auxiliary writing tools during the preparation of this manuscript. Their use was limited to improving grammar, readability, and stylistic consistency of author-written text. They were not used to formulate the research idea, design the method, conduct experiments, analyze results, or derive conclusions. All technical content, experimental settings, reported results, and interpretations were produced and checked by the authors. The authors remain fully responsible for the correctness, integrity, and final content of the manuscript.

B BROADER IMPACTS

SWIFT improves efficient and controllable long-video generation. This capability can support film production, digital content creation, education, and interactive media. The training-free design also reduces extra training cost and lowers the barrier for adapting long-video generators to sequential user prompts.

The same capability may also increase risks from synthetic media. Possible misuse includes misleading video generation, impersonation, and harmful content creation. These risks are shared by many generative video systems. Responsible deployment should include watermarking, provenance tracking, content moderation, and user-facing disclosure. SWIFT is an inference-time method and does not introduce new training data or personal data collection. All text prompts used to generate reasoning were generated by Qwen2-72B-Instruct (Yang et al., 2024), and are harmless, safe, and intended for academic research purposes only.

C DETAILED EXPERIMENTAL SETUP

C.1 COMPUTE RESOURCES

SWIFT is an inference-time framework and requires no additional training for deployment on autoregressive diffusion video generation models. All experiments in this paper are conducted on a single H100 GPU with 80GB memory for inference, which does not impose a high computational requirement in the context of long-video generation.

C.2 SWIFT SETTINGS

Autoregressive generation. All experiments follow the autoregressive inference protocol of LongLive with Wan2.1-T2V-1.3B as the backbone. Each video is generated block by block with $B = 3$ frames per block and $T = 240$ output frames in total. The multi-prompt schedule contains six prompt segments, with switching boundaries at $\mathcal{S} = \{40, 80, 120, 160, 200\}$.

Diffusion inference. The denoising process uses four inference steps, $\mathcal{D} = \{1000, 750, 500, 250\}$, with warped denoising enabled. The timestep shift of the backbone diffusion model is fixed to $\lambda_{\text{ts}} = 5.0$ for all experiments.

Memory configuration. The persistent sink size is set to $N_{\text{sink}} = 3$, and the maximum local attention window is set to $W_{\max} = 12$, corresponding to the `local_attn_size` used in the implementation. For the adaptive dynamic window, we use $W_{\min} = 7$, $\tau_{\text{post}} = 18$, and $\tau_{\text{pre}} = 9$, while the fixed-window ablation uses $W = 12$ throughout generation. Segment-level semantic anchors are enabled by default: the visual summary for each completed segment is computed from the most recent $R_{\text{anchor}} = 6$ frames, and at most $M_{\text{anchor}} = 4$ historical anchors are retained. The semantic mixing coefficient is set to $\alpha = 0.35$, and the anchor injection scale is set to $\gamma_{\text{anchor}} = 0.8$. At each switching boundary $s_m \in \mathcal{S}$, the Semantic Injection Cache constructs transient bridge tokens from $\Delta p_{\perp}^{(m)}$, which are inserted into memory and decayed during subsequent generation steps. The semantic bridge decay factor is set to $\lambda = 0.85$

Algorithm 1: Semantic Injection Cache

Input : Previous prompt signature $p^{(m-1)}$, current prompt signature $p^{(m)}$; value cache of the current layer; recent summary r; sink summary s

Output : Transient semantic bridge entries B_s and B_r

$\Delta p^{(m)} \leftarrow$ Eq. (1)
$\rho^{(m)} \leftarrow$ Eq. (2)
for *each layer* **do**
  Estimate local motion direction m from value cache
  $\Delta p_{\perp}^{(m)} \leftarrow$ Eq. (4)
  Extract recent summary r and sink summary s
  for *each head* **do**
    $(g_r, g_s) \leftarrow$ Eq. (5)
    $(B_r, B_s) \leftarrow$ Eq. (6)
  Append B_s and B_r as transient semantic entries
Read from sink memory, segment anchors, bridge entries, and local window
return B_s, B_r

Algorithm 2: Adaptive Dynamic Window

Input : Current block position t; active segment start s_m; next switch boundary s_{m+1}; window limits $W_{\min}, W_{\max}$; recent value summary $u^{(m)}$; prompt signature $p^{(m)}$

Output : Adaptive window size W_t and segment anchor $A^{(m)}$

Determine active prompt segment for t
$a_t \leftarrow \max(0, t - s_m)$
Compute distance d_t to next switch boundary s_{m+1}
Compute $w_{\text{post}}(t)$ and $w_{\text{pre}}(t)$
$w_t \leftarrow$ Eq. (10)
$W_t \leftarrow$ Eq. (11)
Select effective local window according to W_t
if *current prompt segment ends* **then**
  Summarize recent value cache as $u^{(m)}$
  $A^{(m)} \leftarrow$ Eq. (12)
  Append $A^{(m)}$ into structured memory
Read from sink memory, bridge entries, anchors, and adaptive local window
return $W_t, A^{(m)}$

C.3 Baselines

We compare SWIFT with several state-of-the-art inference-time methods for interactive long-video generation. For a fair comparison, all baseline hyperparameters are kept consistent with the default configurations reported in their original papers.

MemFlow (Ji et al., 2025) is a dynamic memory-retrieval framework for efficient long-context video generation. It retrieves prompt-relevant historical frames to update the memory bank before each generation chunk, and activates only the most relevant memory tokens during attention, thereby enabling consistent long-video generation with limited computational overhead.

FramePack (Zhang et al., 2025b) is a context-packing framework designed for efficient long video generation that optimizes computational throughput and memory usage by packing multiple video frames into a unified context representation.

CausVid (Yin et al., 2025a) is a causal autoregressive video diffusion framework for fast streaming generation. It converts a pretrained bidirectional diffusion transformer into an autoregressive generator and applies video distribution matching distillation to obtain a few-step causal student.

Self-Forcing (Huang et al., 2025) is an autoregressive video diffusion training paradigm that mitigates the exposure bias (train-test gap) by training the model on its own generated outputs using key-value (KV) caching and holistic video-level supervision.

LongLive (Yang et al., 2026) is a real-time interactive video generation framework that introduces a *KV-recache* mechanism to recalculate cached states during prompt transitions, ensuring semantic adherence to new instructions while preserving temporal motion continuity.

D Justification of Proposition 1

This appendix provides the derivation supporting Proposition 1. We first formalize the local cache response model used to define motion-neutral semantic injection, and then derive the closed-form projection as the locally optimal prompt-preserving update.

D.1 Setup and Assumptions

Let $\mathcal{H}$ be the Hilbert space of value-cache representations, equipped with inner product $\langle \cdot, \cdot \rangle$ and induced norm $\| \cdot \|_2$. During autoregressive generation, the cached video representation evolves as

$$v_1, v_2, \ldots, v_t \in \mathcal{H}. \tag{13}$$

At a prompt switch from segment $m-1$ to segment m, the projected prompt displacement is

$$\Delta p^{(m)} = p^{(m)} - p^{(m-1)} \in \mathcal{H}. \tag{14}$$

The local cache tangent is estimated from consecutive value-cache summaries:

$$m = v_t - v_{t-1}, \qquad m \neq 0. \tag{15}$$

Local cache tangent. We assume the cache trajectory is locally first-order smooth around the current state. For a small temporal step Δt, this gives

$$v_t = v_{t-1} + \dot{v}_{t-1}\Delta t + O(\Delta t^2). \tag{16}$$

Thus, $m = v_t - v_{t-1}$ provides a finite-difference estimate of the dominant local tangent direction of short-term cache dynamics. We denote the corresponding motion tangent subspace by

$$\mathcal{T}_t = \mathrm{span}\{m\}, \tag{17}$$

and its orthogonal complement by

$$\mathcal{N}_t = \mathcal{T}_t^{\perp} = \{z \in \mathcal{H} : \langle z, m \rangle = 0\}. \tag{18}$$

First-order motion response. For a small semantic injection $v_t \mapsto v_t + \eta \Delta x$, its tangential effect on the local cache trajectory is measured by the projection onto $\mathcal{T}_t$. The orthogonal projection onto $\mathcal{T}_t$ is

$$\Pi_{\mathcal{T}_t}(z) = \frac{\langle z, m \rangle}{\|m\|_2^2} m. \tag{19}$$

Therefore, the first-order tangential variation caused by the injection is

$$\begin{aligned} \Pi_{\mathcal{T}_t}(v_t + \eta \Delta x) - \Pi_{\mathcal{T}_t}(v_t) &= \eta \Pi_{\mathcal{T}_t}(\Delta x) \\ &= \eta \frac{\langle \Delta x, m \rangle}{\|m\|_2^2} m. \end{aligned} \tag{20}$$

Its magnitude is proportional to $|\langle \Delta x, m \rangle|$. Hence, an update does not perturb the local motion tangent to first order if and only if

$$\Pi_{\mathcal{T}_t}(\Delta x) = 0 \quad \Longleftrightarrow \quad \langle \Delta x, m \rangle = 0. \tag{21}$$

This defines the admissible motion-neutral subspace $\mathcal{N}_t$.

Prompt-preserving semantic injection. The raw prompt displacement $\Delta p^{(m)}$ represents the desired semantic change at the prompt boundary. Since only updates in $\mathcal{N}_t$ are motion-neutral to first order, the injected semantic update should retain as much of $\Delta p^{(m)}$ as possible while remaining in $\mathcal{N}_t$. This leads to the minimum-distortion problem

$$\Delta p_{\perp}^{(m)} = \arg\min_{\Delta x \in \mathcal{H}} \|\Delta x - \Delta p^{(m)}\|_2^2 \quad \text{s.t.} \quad \langle \Delta x, m \rangle = 0. \tag{22}$$

Equivalently, the solution is the motion-neutral component of $\Delta p^{(m)}$ that maximally preserves the prompt displacement under the first-order motion-neutrality constraint.

D.2 Proof

Proof of Proposition 1. For compactness, denote

$$q := \Delta p^{(m)}. \tag{23}$$

The feasible set of Eq. equation 22 is

$$\mathcal{C} := \{z \in \mathcal{H} : \langle z, m\rangle = 0\} = \mathcal{N}_t = m^{\perp}. \tag{24}$$

Since $m \neq 0$, $\mathcal{C}$ is a closed linear subspace of the Hilbert space $\mathcal{H}$. By the Hilbert projection theorem, every $q \in \mathcal{H}$ admits a unique closest point in $\mathcal{C}$. Hence the minimizer of Eq. equation 22 exists and is unique, and is the orthogonal projection of q onto $\mathcal{C}$.

It remains to compute this projection. Since

$$\mathcal{C}^{\perp} = \mathrm{span}\{m\}, \tag{25}$$

there exists a unique orthogonal decomposition

$$q = z_{\perp} + \alpha m, \qquad z_{\perp} \in \mathcal{C}, \quad \alpha m \in \mathcal{C}^{\perp}. \tag{26}$$

Taking the inner product of Eq. equation 26 with m gives

$$\langle q, m\rangle = \langle z_{\perp}, m\rangle + \alpha \|m\|_2^2. \tag{27}$$

Because $z_{\perp} \in \mathcal{C}$, we have $\langle z_{\perp}, m\rangle = 0$. Therefore,

$$\alpha = \frac{\langle q, m\rangle}{\|m\|_2^2}. \tag{28}$$

Substituting this value back into Eq. equation 26 yields

$$z_{\perp} = q - \frac{\langle q, m\rangle}{\|m\|_2^2} m. \tag{29}$$

Returning to $q = \Delta p^{(m)}$, we obtain

$$\Delta p_{\perp}^{(m)} = \Delta p^{(m)} - \frac{\langle \Delta p^{(m)}, m\rangle}{\|m\|_2^2} m. \tag{30}$$

We next verify the minimum-distortion optimality. For any feasible $\Delta x \in \mathcal{C}$, since $q - z_{\perp} \in \mathcal{C}^{\perp}$ and $\Delta x - z_{\perp} \in \mathcal{C}$, the two terms are orthogonal:

$$\langle \Delta x - z_{\perp}, q - z_{\perp}\rangle = 0. \tag{31}$$

Hence the Pythagorean identity gives

$$\begin{aligned} \|\Delta x - q\|_2^2 &= \|(\Delta x - z_{\perp}) - (q - z_{\perp})\|_2^2 \\ &= \|\Delta x - z_{\perp}\|_2^2 + \|q - z_{\perp}\|_2^2 \\ &\geq \|q - z_{\perp}\|_2^2. \end{aligned} \tag{32}$$

Equality holds if and only if $\Delta x = z_{\perp}$. Therefore, $z_{\perp}$ is the unique minimum-distortion update satisfying $\langle \Delta x, m\rangle = 0$.

Finally, we show that the same solution also maximally preserves the prompt transition within the motion-neutral subspace. For any $\Delta x \in \mathcal{C}$, using $q = z_{\perp} + \alpha m$ and $\langle \Delta x, m\rangle = 0$, we have

$$\langle \Delta x, q\rangle = \langle \Delta x, z_{\perp}\rangle. \tag{33}$$

Thus, by the Cauchy–Schwarz inequality, any motion-neutral update with no larger energy than $z_{\perp}$ satisfies

$$\langle \Delta x, q\rangle = \langle \Delta x, z_{\perp}\rangle \leq \|\Delta x\|_2 \|z_{\perp}\|_2 \leq \|z_{\perp}\|_2^2 = \langle z_{\perp}, q\rangle. \tag{34}$$

The upper bound is achieved by $\Delta x = z_{\perp}$. Hence $z_{\perp} = \Delta p_{\perp}^{(m)}$ is not only the closest admissible vector to the raw prompt displacement, but also the locally optimal motion-neutral update that preserves the largest first-order prompt component under the same energy budget. This proves Eq. equation 30 and completes the proof. □

D.3 Stabilized Implementation

The exact projection in Eq. equation 30 requires division by $\|m\|_2^2$. When the finite-difference motion estimate has very small magnitude, this denominator may become numerically unstable. We therefore use the stabilized implementation

$$\widehat{\Delta p}_{\perp}^{(m)} = \Delta p^{(m)} - \frac{\langle \Delta p^{(m)}, m \rangle}{\|m\|_2^2 + \epsilon} m, \qquad \epsilon > 0. \tag{35}$$

This update converges to the exact projection as $\epsilon \to 0$:

$$\lim_{\epsilon \to 0} \widehat{\Delta p}_{\perp}^{(m)} = \Delta p_{\perp}^{(m)}. \tag{36}$$

Its residual first-order motion response is

$$\begin{aligned} \left\langle \widehat{\Delta p}_{\perp}^{(m)}, m \right\rangle &= \left\langle \Delta p^{(m)} - \frac{\langle \Delta p^{(m)}, m \rangle}{\|m\|_2^2 + \epsilon} m, m \right\rangle \\ &= \frac{\epsilon}{\|m\|_2^2 + \epsilon} \langle \Delta p^{(m)}, m \rangle. \end{aligned} \tag{37}$$

Thus, the stabilized update preserves the same projection principle, approaches exact motion orthogonality as ϵ decreases, and avoids numerical amplification when the estimated local motion direction is unreliable.

E Further Experiments

This section provides supplementary ablation studies and detailed analyses regarding the architectural components of SWIFT. These evaluations offer further insights into the system's performance across various interactive generation scenarios.

E.1 Temporal Injection Patterns

We investigate the influence of different injection schedules for the Semantic Injection Cache detailed in Section 3.1 by comparing one-shot, continuously decayed, and constant injection modes. Specifically, one-shot injection inserts the full semantic update at the switching boundary without decay, continuously decayed injection denotes the default semantic injection strategy, and constant injection progressively inserts a fixed proportion of the semantic update over time. This analysis quantifies the trade-off between immediate semantic adaptation and the preservation of long-term temporal stability. Experiments are conducted in the multi-prompt setting by generating 100 videos of 60 seconds, and the results are evaluated using VBench (Huang et al., 2024).

Table 4 shows the quantitative results in terms of aesthetic quality, background consistency, imaging quality, motion smoothness, subject consistency, and temporal flickering. Compared with one-shot injection and constant injection, the continuously decayed injection strategy achieves the best overall visual quality, as it provides sufficient semantic guidance immediately after each prompt transition while gradually reducing the strength of injected bridge tokens in subsequent steps. This design prevents abrupt semantic overwriting caused by constant injection and avoids the insufficient long-range adaptation of one-shot injection, thereby yielding more coherent subject appearance, smoother motion, and reduced temporal flickering across prompt switches.

Table 5 reports the segment-wise CLIP scores under different temporal injection schedules. The results show that continuously decayed injection maintains stronger prompt alignment after semantic transitions, indicating that gradual decay enables more effective semantic adaptation than one-shot or constant injection. Figure 6 presents representative generated frames for the one-shot, constant, and continuously decayed injection schedules under the same multi-prompt sequence.

E.2 Per Block Efficiency Trace

We further analyze the runtime behavior of SWIFT at the autoregressive block level. The profiler records block latency, effective memory read budget, active window size, allocated GPU memory,

Table 4: **Quantitative comparison under multi-prompt 60-second setting** with different injection schedules for the Semantic Injection Cache detailed in Section 3.1, including one-shot, continuously decayed, and constant injection modes.

Injection Schedule	Aesthetic Quality↑	Background Consistency↑	Imaging Quality↑	Motion Smoothness↑	Subject Consistency↑	Temporal Flickering↑
One-shot Injection	59.96	96.01	71.61	99.06	97.03	98.28
Constant Injection	60.03	95.95	71.59	99.09	96.91	98.31
Continuously Decayed Injection	**60.23**	**96.28**	**72.32**	**99.19**	**97.34**	**98.35**

Table 5: **Quantitative comparison of temporal injection schedules.** We evaluate one-shot, constant, and continuously decayed injection under the multi-prompt 60-second setting. The CLIP (Radford et al., 2021) score is computed for each 10-second segment according to the corresponding prompt semantics, where continuously decayed injection achieves the best overall alignment across prompt transitions.

Injection Schedule	CLIP Score↑					
	0–10 s	10–20 s	20–30 s	30–40 s	40–50 s	50–60 s
One-shot Injection	**26.60**	25.42	23.74	23.85	24.18	23.67
Constant Injection	26.32	25.77	23.18	23.76	24.25	23.89
Continuously Decayed Injection	26.53	**25.86**	**24.71**	**24.04**	**24.68**	**24.13**

and peak allocated GPU memory. Prompt switch boundaries are also logged. These metrics measure the local cost of semantic transition and the memory budget used during stable rollout. We use an autoregressive generation baseline with a fixed window size and ReCache at prompt boundaries as the comparison setting.

Figure 7 shows the efficiency trace per block. SWIFT shows mild latency increases near prompt switch boundaries. The average latency is 132.7 ms, and the average boundary latency is 144.8 ms. The result indicates stable prompt switching without severe runtime spikes. The fixed window reference uses a constant large local window across all blocks. This strategy keeps a high read budget during both transition phases and stable phases. In contrast, SWIFT increases the read budget near prompt boundaries and reduces it during stable segments. The effective read budget averages 15.1K tokens and ranges from 12.5K to 18.7K tokens. GPU memory remains nearly constant, with 17.56 GiB average usage and 18.70 GiB peak usage.

The efficiency advantage of SWIFT comes from prompt phase aware memory allocation. A larger window supplies sufficient context for semantic handover near prompt switches. A smaller window reduces redundant attention computation during stable generation. Segment anchors preserve compact long range semantics under the reduced local window. This design improves runtime efficiency over the fixed window setting while retaining stable memory usage.

Table 6: **Quantitative comparison under different multi-prompt video lengths.** We compare SWIFT with LongLive across 30, 45, 60, 75, and 90 seconds using VBench quality and consistency scores.

Method	**30s**		**45s**		**60s**		**75s**		**90s**	
	Quality Score↑	Consistency Score↑	Quality Score↑	Consistency Score↑	Quality Score↑	Consistency Score↑	Quality Score↑	Consistency Score↑	Quality Score↑	Consistency Score↑
LongLive (Yang et al., 2026)	83.64	96.08	83.76	96.31	83.88	96.48	84.47	96.56	84.58	96.67
SWIFT	**84.72**	**96.43**	**84.83**	**96.51**	**84.97**	**96.62**	**85.01**	**96.74**	**85.06**	**96.89**

E.3 Multi-Prompt Generation with Different Video Lengths

We further evaluate SWIFT under different generation lengths to examine its scalability in multi-prompt long-video generation. Specifically, we vary the target video duration across 30, 45, 60, 75, and 90 seconds while keeping the six-prompt interactive setting. The prompt switching boundaries are uniformly scaled according to the target duration, and all remaining model configurations, inference hyperparameters, and evaluation protocols are kept unchanged.



Figure 6: **Qualitative examples of temporal injection schedules.** We visualize generated frames from one-shot, constant, and continuously decayed injection under the same multi-prompt sequence. One-shot injection causes abrupt semantic changes, constant injection shows weaker adaptation after prompt switches, and continuously decayed injection provides smoother semantic transitions while preserving temporal consistency.

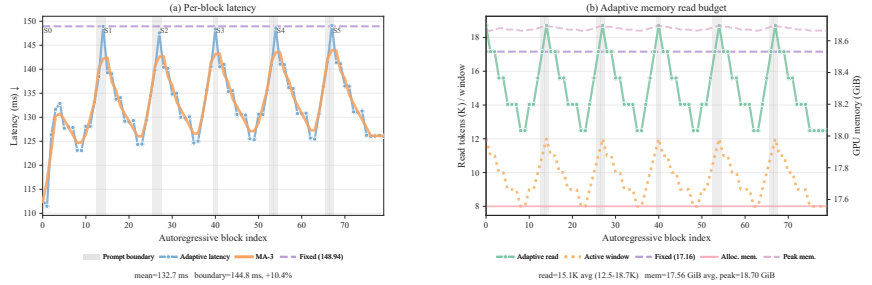

Figure 7: **Per-block efficiency of SWIFT.** Gray bands mark prompt-switch boundaries, and dashed lines denote fixed-window references. SWIFT shows only mild latency increases at prompt transitions, while its adaptive memory schedule expands the read budget near boundaries and contracts it in stable segments, with nearly constant GPU memory usage.

Table 6 reports the video quality metrics on the VBench benchmark across different generation lengths. Compared with LongLive, SWIFT consistently maintains better overall visual quality across varying temporal horizons, demonstrating the scalability and robustness of the proposed memory mechanism for multi-prompt long-video generation. Figure 11 shows representative frames from the 30-second six-prompt setting, where SWIFT preserves the main subject and interaction structure more consistently across prompt transitions. Under the shorter 30-second generation setting, rapid semantic injection enables SWIFT to respond more effectively to dense prompt switches while maintaining coherent subject appearance and scene continuity.

E.4 User Study

We conduct a user study to complement the automatic evaluation. We randomly select 30 multi-prompt generation cases from the 60-second evaluation set. Ten participants compare SWIFT with LongLive (Yang et al., 2026) and MemFlow (Ji et al., 2025) in pairwise evaluations. The display order is randomized and the method identity is hidden. Each comparison considers three criteria: transition consistency, prompt following, and aesthetic quality. Participants select Win, Tie, or Lose for SWIFT against each baseline. The reported percentages are computed over 300 pairwise judgments for each baseline and criterion.

Table 7 shows a consistent human preference for SWIFT across all three criteria. The strongest gains appear in prompt following and transition consistency. SWIFT receives 61.0% and 65.7% win rates in prompt following against LongLive and MemFlow, respectively. It also reaches 58.7% and 64.3% win rates in transition consistency. These results indicate more stable semantic handover and stronger responsiveness to successive prompt updates. Aesthetic quality remains competitive, with substantial tie rates for both comparisons. The study therefore supports the automatic results and confirms the advantage of SWIFT in interactive multi-prompt generation.

Table 7: Human preference results for SWIFT against LongLive and MemFlow. Win, Tie, and Lose denote the percentages of pairwise judgments favoring SWIFT, judging both methods comparable, or favoring the corresponding baseline.

Criterion	Baseline	Win (%)	Tie (%)	Lose (%)
Transition Consistency	LongLive	**58.7**	29.0	12.3
	MemFlow	**64.3**	25.0	10.7
Prompt Following	LongLive	**61.0**	27.3	11.7
	MemFlow	**65.7**	23.3	11.0
Aesthetic Quality	LongLive	**49.7**	34.0	16.3
	MemFlow	**54.3**	31.0	14.7

E.5 Case Study

We present additional qualitative results in Figure 9 and Figure 10 to illustrate the videos generated by SWIFT from multi-prompt inputs. These examples show that SWIFT can follow sequential prompt changes while preserving coherent visual progression across long video generation.

F Limitation Analysis

SWIFT is a training-free inference-time framework built on a pretrained causal video diffusion backbone, so its quality remains bounded by the backbone's visual fidelity, motion prior, and instruction-following ability. Lightweight cache injection and prompt-phase adaptive memory allocation improve semantic transition and memory utilization, but do not introduce new visual concepts or correct intrinsic backbone failures. Difficult cases involving rare objects, fine-grained physical interactions, crowded scenes, or large camera motion may still produce artifacts or inconsistent details. Future work can combine SWIFT with supervised transition data, stronger base video generators, and learned memory scheduling to improve robustness. As shown in Figure 11, SWIFT responds to the newly introduced research partner, but fine-grained facial structure and identity consistency degrade in several frames. This illustrates that training-free semantic memory adaptation improves prompt transition and temporal continuity without fully eliminating backbone-level artifacts under detailed human interaction and compositional prompt changes.



Figure 9: **Qualitative example of 60-second multi-prompt video generation.** We visualize representative frames generated by SWIFT under a six-prompt sequence.

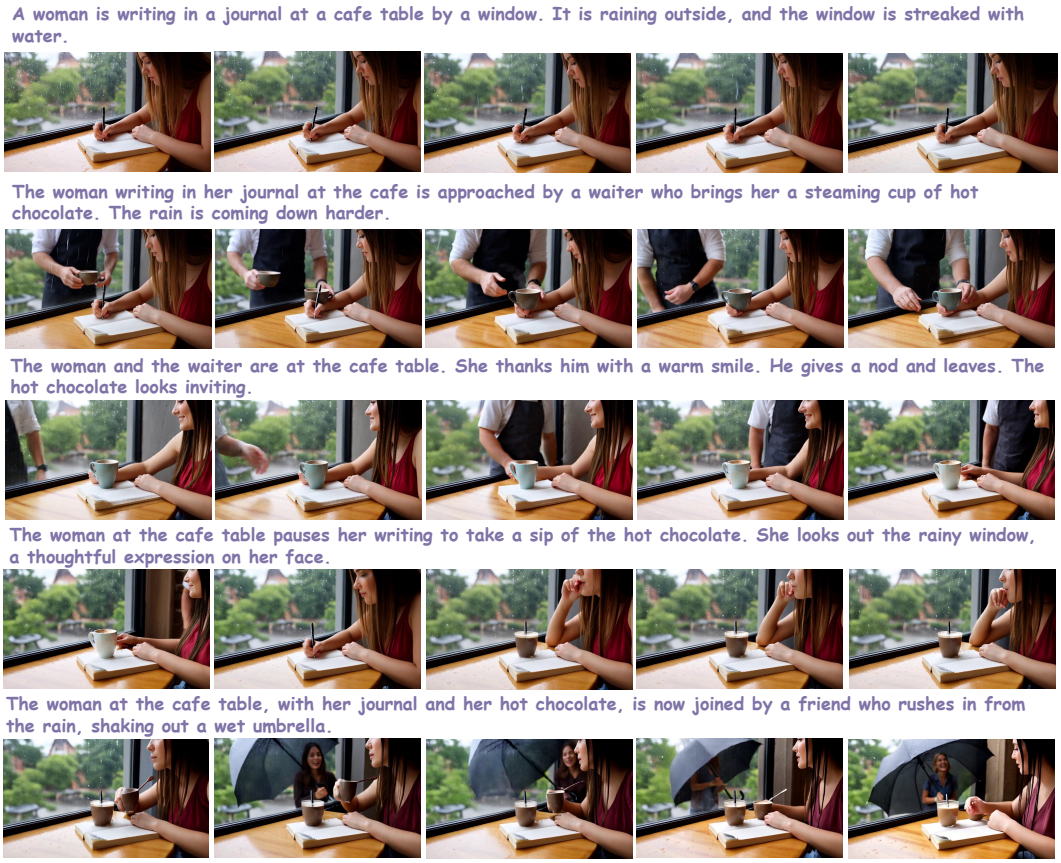

Figure 10: **Qualitative example of 60-second multi-prompt video generation.** We visualize representative frames generated by SWIFT under a six-prompt sequence.



Figure 11: **Qualitative example of 30-second multi-prompt video generation.** We visualize representative frames generated by SWIFT and LongLive under the same six-prompt sequence.

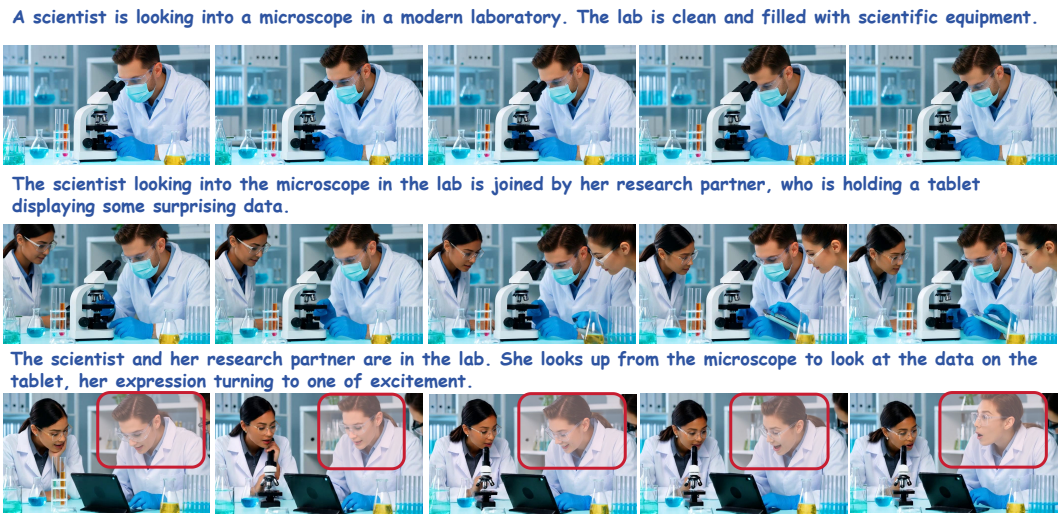

Figure 11: **Failure case of SWIFT under complex multi-prompt generation.** Although SWIFT preserves the laboratory scene and maintains a coherent visual layout across prompt transitions, it can still inherit limitations from the pretrained video diffusion backbone.