Deep Forcing: Training-Free Long Video Generation with Deep Sink and Participative Compression

Jung Yi Wooseok Jang Paul Hyunbin Cho Jisu Nam Heeji Yoon Seungryong Kim

KAIST AI

https://cvlab-kaist.github.io/DeepForcing

Figure 1. Our ***training-free*** approach, **Deep Forcing**, achieves comparable visual quality to ***training-based*** baselines, such as Rolling Forcing [17] and LongLive [31]. Notably, Deep Forcing enables minute-long video generation while maintaining visual quality and dynamics ***without*** requiring any additional training.

Abstract

Recent advances in autoregressive video diffusion have enabled real-time frame streaming, yet existing solutions still suffer from temporal repetition, drift, and motion deceleration. We find that naïvely applying StreamingLLM-style attention sinks to video diffusion leads to fidelity degradation and motion stagnation. To overcome this, we introduce ***Deep Forcing****, which consists of two* ***training-free*** *mechanisms that address this without any fine-tuning. Specifically, 1)* ***Deep Sink*** *dedicates half of the sliding window to persistent sink tokens and re-aligns their temporal RoPE phase to the current timeline, stabilizing global context during long rollouts. 2)* ***Participative Compression*** *performs importance-aware KV cache pruning that preserves only tokens actively participating in recent attention while safely discarding redundant and degraded history, minimizing error accumulation under out-of-distribution length generation. Together, these components enable* ***over 12× extrapolation*** *(e.g.* ***5s-trained → 60s+*** *generation) with bet-*

ter imaging quality than LongLive, better aesthetic quality than RollingForcing, almost maintaining overall consistency, and substantial gains in dynamic degree, all while maintaining real-time generation. Our results demonstrate that training-free KV-cache management can match or exceed training-based approaches for autoregressively streaming long-video generation.

1. Introduction

Recent advances in video diffusion models [10, 25, 33] have demonstrated remarkable capabilities in synthesizing short video clips (e.g, 50–81 frames) with both high visual fidelity and coherent motion dynamics.

Building upon this progress, emerging interactive systems, such as world models [1, 18, 34], now require autoregressive video generation (e.g., 1-2 minutes), where frames are streamed sequentially in real time for immediate downstream applications [18, 21, 34]. Unlike conventional offline video generation, which synthesizes entire clips at once, autoregressive video generation operates in an online manner, with each frame generated, emitted, and consumed instantaneously. Self Forcing [12] and its variants [8, 17, 31] have become standard in this field, leveraging a causal attention mask and the Key-Value (KV) cache from previous frames.

However, this existing autoregressive formulation is inherently susceptible to *error accumulation over long horizons*, as each predicted frame depends on previously generated and potentially imperfect frames [12, 27, 35]. Such accumulation error leads to fidelity degradation, in which visual quality deteriorates as colors drift toward oversaturation, textures blur, and fine details disappear.

To mitigate this, several works [4, 22] introduce history corruption by adding noise to previous frames during training. While this improves robustness to noisy generated histories at inference, a discrepancy remains between generated noise and artificially injected noise, leaving models vulnerable to long-horizon drift. Self Forcing [12] aims to reduce this gap by training on self-generated histories, but their heavy reliance on past frames' KV caches still leads to accumulated errors.

On the other hand, recent LLM studies [9, 16, 20], inspired by StreamingLLM [29], introduce an *attention sink*, in which newly generated tokens attend strongly to a small set of initial global tokens, helping stabilize the attention distribution and improve overall performance.

Motivated by these insights, we propose **Deep Forcing**, a novel **tuning-free** method that addresses error accumulation in long-horizon video generation. Our approach is able to generate minute-long video that maintain both visual fidelity and motion stability, while requiring no fine-tuning, outperforming even training-based approaches [17, 31] (Fig. 1).

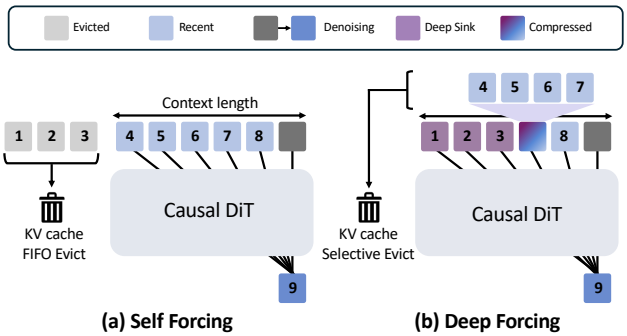

Figure 2. **Comparison of KV Cache Management.** (a) **Self Forcing** [12] adopts a FIFO policy that discards the earliest tokens regardless of their importance, often losing critical context and degrading generation quality. In contrast, our (b) **Deep Forcing** performs selective eviction by preserving **Deep Sink** tokens and applying KV-cache compression, effectively mitigating visual degradation during long-horizon generation.

Specifically, we observe that the pre-trained Self Forcing [12] inherently exhibits strong attention sink behavior, not only attending to the first few tokens but also strongly attending to intermediate tokens. Building on this finding, we first introduce **Deep Sink**, which (i) maintains a large deep-sink ratio (typically 40–60%) and (ii) dynamically adjusts the Relative Positional Embedding (RoPE) for long video generation. Second, we further present **Participative Compression** that retains only the most informative tokens in the KV cache by selecting them based on their importance to queries from recent frames. This removes redundancy and significantly reduces fidelity degradation caused by noise accumulation from outdated tokens.

We implement our method on top of the pre-trained Self Forcing. Comprehensive evaluations using VBench [13], user studies, and VLM assessments demonstrate that our training-free approach significantly enhances the baseline without any fine-tuning. We also achieve state-of-the-art performance on several metrics, even surpassing training-based methods [17, 31]. Our ablation studies further validate the effectiveness of each design choice.

Our contribution is summarized as follows:

- We propose a **tuning-free** autoregressive video generation framework, dubbed **Deep Forcing**, that significantly mitigates error accumulation in long-horizon generation.
- We introduce **Deep Sink**, which stabilizes long-horizon generation by leveraging the inherent attention sink behavior of Self Forcing [12] while adjusting the relative positional gap.
- We present **Participative Compression**, a lightweight KV selection mechanism that removes redundant tokens.
- Our training-free method achieves state-of-the-art performance on VBench and user studies, surpassing existing training-based approaches.

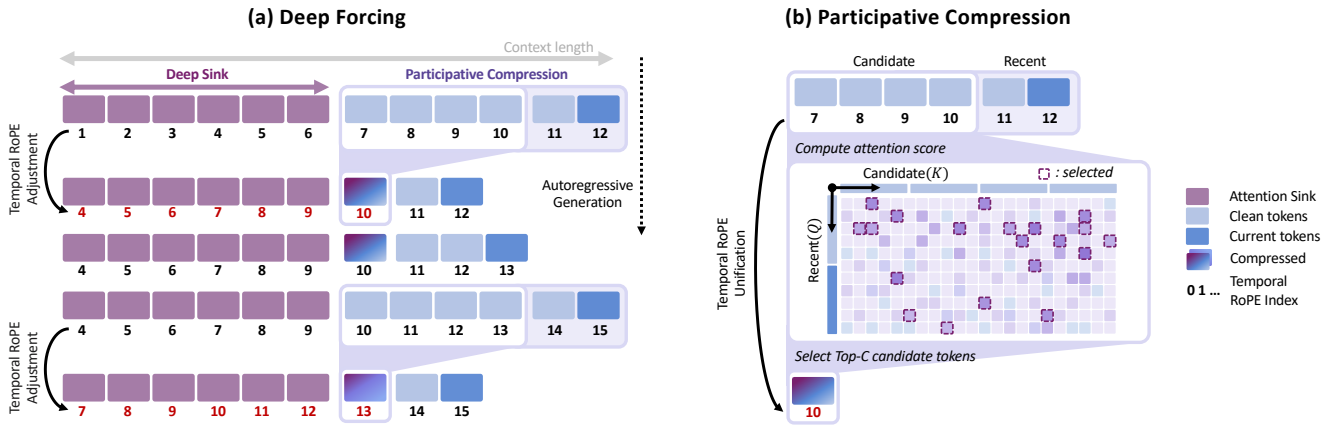

Figure 3. **Overview of Deep Forcing. (a) Deep Forcing** maintains a substantially enlarged attention sink (Deep Sink) covering approximately half the context window, combined with Participative Compression for the remaining rolling portion. Temporal RoPE adjustment aligns the sink tokens' temporal indices with current frames to maintain temporal coherence. **(b) Participative Compression** computes query-averaged attention scores between recent tokens and candidate tokens, selecting the top-C most important tokens to retain in the compressed cache while evicting redundant tokens.

2. Related Work

Autoregressive Video Diffusion. A growing line of work [4, 12, 14, 24, 35] combines diffusion modeling with autoregressive (AR) prediction to support long-horizon or streaming video generation. MAGI-1 [24] generates videos chunk-by-chunk autoregressively with progressive denoising, enabling streaming generation. CausVid [35] converts a pre-trained bidirectional diffusion transformer into a causal AR generator with KV caching. Building on these ideas, Self Forcing [12] addresses the train–inference mismatch by conditioning the model on its own generated frames. Rolling Forcing [17] proposes expanding the diffusion window, and LongLive [31] incorporates KV recaching to maintain visual continuity while ensuring prompt adherence across scene transitions. In contrast, our method is fully tuning-free. We show that the pre-trained Self Forcing already has attention-sink behavior and demonstrate how to leverage it effectively to surpass existing training-based methods.

Attention Sink. Recent work has revealed that attention in autoregressive models concentrates disproportionately on initial tokens, termed attention sinks [29]. StreamingLLM [29] showed that retaining these sink tokens within a sliding window enables robust generation beyond the training context length.

Building on this insight, recent autoregressive video models [17, 31] maintain the first three frames as attention sinks via model distillation or fine-tuning. We demonstrate that the pre-trained autoregressive video diffusion model [12] exhibits inherent attention sink behavior that can be effectively leveraged without training, requiring deeper context preservation.

KV Cache Compression. The linearly growing KV cache in autoregressive generation motivates compression strategies that reduce memory footprint while preserving generation quality. As the cache grows, attention becomes distributed across increasingly many tokens, diluting focus on critical context and degrading output quality. To address this, recent works employ attention-based token selection for long-context LLM generation. H2O [37] and SnapKV [16] preserve important tokens based on cumulative attention scores and observation windows, respectively. D2O [26] dynamically allocates budgets across layers, while MorphKV [9] maintains constant-size caches through correlation-aware ranking. While these methods target language models, similar memory constraints arise in autoregressive video diffusion, where temporal context must be efficiently maintained across frames. We extend these principles through Participative Compression.

3. Preliminaries

Autoregressive Video Diffusion. Autoregressive video diffusion models [5, 12, 24] produce each frame or chunk conditioned on previously generated frames within a denoising diffusion process.

Given a video sequence of N frames $x^{1:N} = (x^1, x^2, \ldots, x^N)$, the autoregressive model applies the chain rule to factorize the joint distribution as

$$p(x^{1:N}) = \prod_{i=1}^{N} p(x^i \mid x^{<i}). \tag{1}$$

A diffusion model parameterizes each conditional $p(x^i \mid x^{<i})$, generating the i-th frame by conditioning on previously generated frames $x^{<i} = (x^1, x^2, \ldots, x^{i-1})$.

Self Forcing. Self Forcing [12] generates videos in an autoregressive manner using a rolling KV cache mechanism, producing frames or frame chunks sequentially, enabling efficient long video generation. Each frame is encoded into latent tokens through the VAE encoder. The method maintains a fixed-size cache of length L that stores key-value pairs corresponding to the most recent L frames. When the cache reaches capacity, the oldest entry is evicted to accommodate new frames, thereby maintaining a sliding context window over the L most recent frames. During generation, self-attention is computed between queries from the frame(s) currently being generated and the keys and values of cached context frames. Specifically, Self Forcing employs a 4-step denoising process with noise schedule $\{t_0 = 0, t_1 = 250, t_2 = 500, t_3 = 750, t_4 = 1{,}000\}$ ($T = 4$), totaling 5 noise levels. Each frame i is denoised iteratively across these timesteps. At denoising step j, the model processes a noisy intermediate frame $x^i_{t_j}$, conditioned on the KV cache of previously generated clean frames. The predicted clean frame is then perturbed with Gaussian noise at a lower noise level t_{j-1} via the forward diffusion process Ψ, producing $x^i_{t_{j-1}}$ for the next denoising iteration. Formally, this process is expressed as:

$$x^i_{t_{j-1}} = \Psi\big(G_\theta(x^i_{t_j}, t_j, KV), t_{j-1}\big), \tag{2}$$

where $x^i_{t_4} \sim \mathcal{N}(0, I)$ and KV denotes the KV cache from previously generated frames.

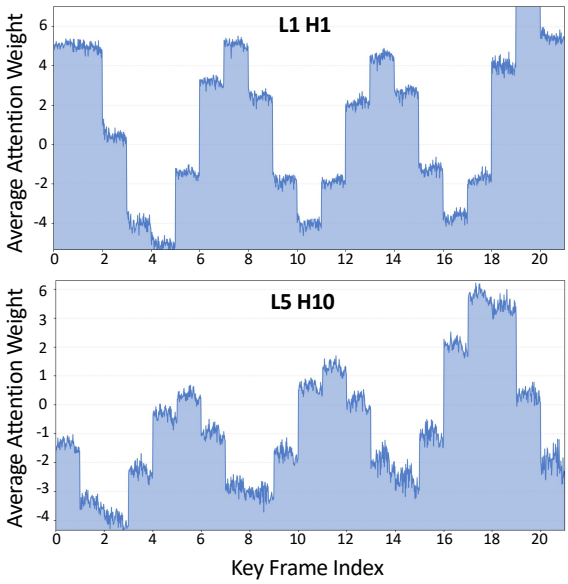

Figure 4. **Attention weight distribution across earlier frames.** Query-averaged attention showing how the last chunk (frames 19-21) attends to earlier KV cache entries (frames 0-18). We visualize two representative attention heads from different layers—L1H1 (layer 1, head 1) and L5H10 (layer 5, head 10)—demonstrating that substantial attention is maintained across the entire context window, not just initial frames. See Appendix H for additional heads analysis.

4. Method

4.1. Overview

We propose a novel training-free method to mitigate error accumulation in long-horizon video generation. Drawing inspiration from the attention sink mechanism in large language models (LLMs) [29], our work begins by thoroughly investigating the attention mechanism within the pre-trained Self Forcing [12]. Based on this investigation, we introduce two core components: **Deep Sink**, which maintains approximately half of the sliding window as attention sinks, and **Participative Compression**, which selectively retains important tokens in the KV cache, while evicting redundant ones. Our method is illustrated in Fig. 3

4.2. Deep Sink

Motivation. Self Forcing [12] employs a sliding window to autoregressively extrapolate video frames. However, because the model is distilled from short video clips (e.g., 5-second segments), its frame fidelity deteriorates significantly when generating sequences that extend far beyond its training domain.

This degradation is a known challenge in autoregressive systems. In the LLM domain, the attention sink mechanism [29] was introduced as a simple yet effective technique to mitigate performance drift during sliding-window inference. While several works [17, 31] have investigated adapting the attention sink mechanism to redistribute attention probabilities and stabilize LLM performance, no prior work has explored how to achieve a similar stabilizing effect in autoregressive video diffusion models in a **training-free** manner.

To address this gap, we first analyze the attention behavior of the pre-trained Self Forcing. As illustrated in Fig. 4, we specifically examine how newly generated latent frames attend to earlier frames in the KV cache. Contrary to the conventional understanding [29] that only a small set of initial KV tokens (latent frames) needs to be retained, our analysis reveals: most attention heads allocate substantial weight to not only the earliest tokens, but also assign comparable attention to the middle of the sequence.

Deepening Sink Size. Based on this observation, we hypothesize that more tokens up to the middle of the initial sequence are essential for high-quality video generation. To evaluate this hypothesis, we measured the influence of different attention sink sizes on the generation quality of long videos. To rigorously assess long-horizon generation quality, we first define our key metrics from VBench [13]: Overall Consistency and Aesthetic Quality, which use Vi-

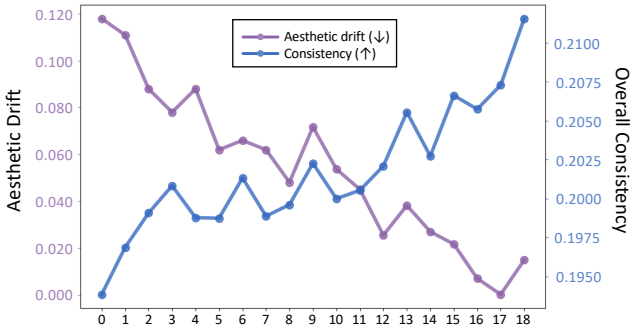

Figure 5. **Ablation study on Deep Sink depth.** We evaluate the effect of sink depth on video quality using Aesthetic Drift (↓) and Overall Consistency (↑) metrics on 50-second videos from the first 21 prompts in MovieGen [19].

CLIP [28] and LAION aesthetic predictor [15], respectively. Following standard practice in long video generation [17, 35, 36], we compute $\Delta_{\text{Drift}}^{\text{Quality}}$ as the absolute difference in aesthetic quality between the first and the last five seconds of each 50-second generated video. Our results demonstrate a clear trend (Fig. 5): as the sink frame size increases, Overall Consistency improves and Aesthetic Quality Drift ($\Delta_{\text{Drift}}^{\text{Quality}}$) decreases. This finding suggests that intermediate frames function as crucial anchors, effectively maintaining both temporal coherence and visual fidelity throughout the long generation process. Consequently, we find that effective attention sinking in Self Forcing [12] emerges from deep, extended temporal anchoring, a mechanism that differs from the shallow, initial-frame fixation used in StreamingLLM [29].

Temporal RoPE Adjustment. RoPE [23] is widely used as the positional embedding in video diffusion models, and recent architectures [6, 24, 25, 33] commonly adopt 3D RoPE, which encodes temporal, height, and width dimensions separately. However, attention sinks in video require the model to attend to past frames, and directly applying 3D RoPE under this setting leads to large temporal discrepancies, where tokens at vastly different timestamps (e.g., $t = 1$ vs. $t = 200$) are forced to attend to each other. This breaks the continuity of video and results in (1) flickering, (2) fidelity degradation, and (3) roll-back, where previously sinked frames are regenerated (a detailed analysis is provided in the Appendix A). To address this, we propose selectively adjusting only the temporal dimensions while preserving the original spatial encoding.

Specifically, we selectively modify the temporal RoPE index by applying a temporal offset to the attention sink's temporal index. This reduces the temporal gap between the attention sink and the remaining tokens, while preserving the spatial indices unchanged.

We divide the key and value caches K and V in the current sliding window into two parts: the sink ($K_{\text{sink}}, V_{\text{sink}}$) for deep sink tokens and the tail ($K_{\text{tail}}, V_{\text{tail}}$) for the rest.

$$K = \left[K_{\text{sink}} \,\|\, K_{\text{tail}} \right], \tag{3}$$

$$V = \left[V_{\text{sink}} \,\|\, V_{\text{tail}} \right], \tag{4}$$

where $\left[\cdot \,\|\, \cdot \right]$ denotes concatenation.

Let s_{tail} denote the first frame index of the tail and let s_{sink} be the last frame index of the deep sink.

We then define Δ_{sink}, which is the temporal gap between s_{tail} and s_{sink}, as follows:

$$\Delta_{\text{sink}} = s_{\text{tail}} - s_{\text{sink}}. \tag{5}$$

We apply Δ_{sink} to $K_{\text{sink}}^{(\text{time})}$, which is temporal component of K_{sink}, using the RoPE temporal frequency vector $\boldsymbol{\omega}_t$:

$$K_{\text{sink}}^{(\text{time})} \leftarrow K_{\text{sink}}^{(\text{time})} \odot \exp\left(i\, \boldsymbol{\omega}_t\, \Delta_{\text{sink}}\right), \tag{6}$$

where i is the imaginary unit, $\odot$ denotes element-wise multiplication. This further rotates K_{sink} to align the relative temporal positions of sink and tail tokens.

4.3. Participative Compression

Motivation. While Deep Sink effectively mitigates fidelity degradation compared to the baseline Self Forcing [12], it alone cannot fully alleviate quality degradation in minute-long video generation. When extrapolating from 5-second training clips to sequences more than 12× longer, a critical issue emerges: *degeneration*, where visual fidelity and overall quality progressively deteriorate. This phenomenon is well-documented in autoregressive long-context generation [9, 11]: when generating beyond training length, indiscriminate token retention causes attention to dilute across both relevant and irrelevant context, introducing compounding noise. Beyond its training distribution, the growing KV cache retains increasingly irrelevant tokens, further diluting attention.

Recent analysis of video diffusion models reveals that attention concentrates on a small subset of semantically critical tokens, with the majority contributing minimally to generation [32]—suggesting that pruning low-attention tokens can substantially reduce computation with limited impact on quality. Building on this insight and importance-aware compression [9, 16, 37], we propose **Participative Compression**, which dynamically identifies and retains only contextually relevant tokens while pruning those that could contribute to attention dilution and error accumulation.

Overview. Self Forcing [12] implements the rolling KV cache by evicting the earliest frame when the cache is filled. In comparison, Participative Compression (PC) operates at the token level, selectively removing redundant tokens by

ranking them according to their aggregated attention scores from recent frames, rather than using a simple FIFO (First-In, First-Out) policy as illustrated in Fig 2.

PC introduces two key hyperparameters: (1) *Budget* (N), the target number of tokens to retain after compression, and (2) *Recent* (R), the number of tokens from the most recent frames that are excluded from compression, in addition to the S sink frame tokens that are always preserved. PC is applied when the sliding window reaches maximum length M tokens, compressing the cache to size $N \leq M$. The compression operates on $K_{\text{cand}}, V_{\text{cand}}$, which contain all tokens except those from the first S and the most recent R.

- **Recent**: $K_{\text{rct}}, V_{\text{rct}}$ containing tokens from the most recent R frames, excluded from compression to preserve local coherence.
- **Candidate**: $K_{\text{cand}}, V_{\text{cand}}$ containing all intermediate tokens between the Sink and Recent, subject to compression.

For each token in $K_{\text{cand}}, V_{\text{cand}}$, PC computes an importance score by summing its attention weights from all recent R frames—tokens frequently attended to are deemed critical for maintaining temporal coherence. PC then selects the top $C = N - R - S$ tokens with the highest importance scores to form $K_{\text{top}}, V_{\text{top}}$. The final KV cache contains N tokens: S sink tokens, C compressed tokens, and R recent tokens.

Top-C Selection. PC selectively retains the C most important candidate tokens based on their relevance to current generation, evicting those not selected by the Top-C operator. To determine which tokens to retain, PC computes attention scores between the recent queries (Q_{rct}) and candidate keys (K_{cand}). We aggregate these scores across all recent queries by summing along the query dimension, producing a unified importance score ϕ_j for each candidate key:

$$\phi_j = \sum_{r=1}^{R} \mathbf{q}_r^\top \mathbf{k}_j, \tag{7}$$

where j indexes the candidate keys, $\mathbf{q}_r$ denotes the r-th query in Q_{rct}, and $\mathbf{k}_j$ denotes the j-th key in K_{cand}. Higher ϕ_j indicates higher importance for current generation. We then form the importance vector $\boldsymbol{\phi} = [\phi_1, \phi_2, \ldots, \phi_{|K_{\text{cand}}|}]$ and select the Top-C tokens with the highest scores:

$$K_{\text{top}} = \text{Top-C}(\boldsymbol{\phi}). \tag{8}$$

Finally, the compressed cache is formed by concatenating the preserved components in temporal order:

$$K_{\text{compressed}} = \left[K_{\text{sink}} \parallel K_{\text{top}} \parallel K_{\text{rct}} \right], \tag{9}$$

where K_{rct} contain keys from the first S and most recent R, respectively. Values (V_{top}) are processed identically. This yields a compact cache structure combining long-term initial context (Sink), selectively important intermediate tokens (Top-C), and fresh recent context (Recent), all within a fixed budget of N.

Algorithm 1 Participative Compression with Deep Sink

Input: KV cache $[K, V]$ of size M; Sink size S; Recent R; Top-C capacity C; Timestep t; first time step T;

if $M \geq$ MAX_WINDOW_LENGTH **and** $t = T$ **then**
  // Partition cache into three regions
  $\mathcal{I}_{\text{sink}} \leftarrow [0, S)$ ▷ First S frames
  $\mathcal{I}_{\text{rct}} \leftarrow [M - R, M)$ ▷ Last R frames
  $\mathcal{I}_{\text{cand}} \leftarrow [S, M - R)$ ▷ Candidate tokens/frames
  if $|\mathcal{I}_{\text{cand}}| > 0$ **and** $C > 0$ **then**
    // Compute importance scores (Eq. 7)
    $Q_{\text{rct}} \leftarrow Q[\mathcal{I}_{\text{rct}}]$ ▷ Recent queries
    $K_{\text{cand}} \leftarrow K[\mathcal{I}_{\text{cand}}]$ ▷ Candidate keys
    for $j = 1$ to $|\mathcal{I}_{\text{cand}}|$ **do**
      $\phi_j \leftarrow \sum_{r=1}^{R} \mathbf{q}_r^\top \mathbf{k}_j$ ▷ Aggregate attention
    // Select top-C tokens (Eq. 8)
    $\boldsymbol{\phi} \leftarrow [\phi_1, \phi_2, \ldots, \phi_{|\mathcal{I}_{\text{cand}}|}]$
    $\mathcal{I}_{\text{top}} \leftarrow \text{TopC}(\boldsymbol{\phi})$ ▷ Select C highest
    // Temporal RoPE Unification (Section 4.3)
    $\Delta_{\text{top}} \leftarrow s^{\text{top}} - s^{\text{top}}_{\text{base}}$
    $K^{(\text{time})}_{\text{top}} \leftarrow K^{(\text{time})}_{\text{top}} \odot \exp(i\boldsymbol{\omega}_t \Delta_{\text{top}})$
  else
    $\mathcal{I}_{\text{top}} \leftarrow \varnothing$
  // Assemble compressed cache (Eq. 9)
  $K_{\text{compressed}} \leftarrow [K_{\text{sink}} \parallel K_{\text{top}} \parallel K_{\text{rct}}]$
  $V_{\text{compressed}} \leftarrow [V_{\text{sink}} \parallel V_{\text{top}} \parallel V_{\text{rct}}]$
  return $K_{\text{compressed}}, V_{\text{compressed}}$
else
  return K, V ▷ No compression

Temporal RoPE Unification. After selecting the Top-C tokens, we apply RoPE adjustment to maintain temporal dimension consistency, following the same approach as Deep Sink (Section 4.2). We adjust only the temporal dimension of the Top-C keys' RoPE while preserving their spatial information intact.

Let s^{top} denote the desired absolute temporal position where the Top-C block should be aligned, and let $s^{\text{top}}_{\text{base}}$ represent the current temporal position of each cached Top-C key. We compute the temporal adjustment:

$$\Delta_{\text{top}} = s^{\text{top}} - s^{\text{top}}_{\text{base}}. \tag{10}$$

This temporal shift Δ_{top} is then applied to $K^{(\text{time})}_{\text{top}}$, which is temporal component of K_{top}, re-aligning each Top-C key

Table 1. Quantitative comparison on long video generation. We evaluate Deep Forcing against open-source autoregressive video diffusion generation baselines on 30-second and 60-second videos across multiple quality metrics on VBench-Long [13].

Model	Throughput (FPS) ↑	Dynamic Degree ↑	Motion Smoothness ↑	Overall Consistency ↑	Imaging Quality ↑	Aesthetic Quality ↑	Subject Consistency ↑	Background Consistency ↑
Trained with Attention Sink	**30 seconds**							
Rolling Forcing [17]	15.79	30.71	98.75	20.99	70.58	60.24	98.12	96.91
LongLive [31]	18.16	45.55	98.76	20.16	69.07	61.51	97.97	96.83
*Trained **without** Attention Sink*								
CausVid [35]	15.78	47.21	98.08	19.15	66.36	59.77	97.92	96.77
Self Forcing [12]	15.78	36.62	98.63	20.50	68.58	59.44	97.34	96.47
Deep Forcing (Ours)	15.75	57.56	98.27	20.54	69.31	60.68	97.34	96.48
Trained with Attention Sink	**60 seconds**							
Rolling Forcing [17]	15.79	31.35	98.69	20.64	70.25	59.75	97.97	96.76
LongLive [31]	18.16	43.49	98.75	20.29	69.11	61.29	97.85	96.74
*Trained **without** Attention Sink*								
CausVid [35]	15.78	46.44	98.09	18.78	65.84	59.42	97.81	96.75
Self Forcing [12]	15.78	31.98	98.21	18.63	66.33	56.45	96.82	96.31
Deep Forcing (Ours)	15.75	57.19	98.23	20.38	69.27	59.86	96.96	96.32

using the complex phase rotation defined by the RoPE temporal frequencies $\boldsymbol{\omega}_t$:

$$K_{\text{top}}^{(\text{time})} \leftarrow K_{\text{top}}^{(\text{time})} \odot \exp\big(i\,\boldsymbol{\omega}_t\,\Delta_{\text{top}}\big). \tag{11}$$

where i is the imaginary unit, and $\odot$ denotes element-wise multiplication.

This rotation adjusts the temporal positioning of K_{top} to create a continuous temporal sequence across all three cache components (Sink, Top-C, Recent), preventing temporal discontinuities that would otherwise cause fidelity degradation, flickering, and roll-back artifacts as demonstrated in Appendix A.

Efficiency. The complexity of Participative Compression (PC) might initially suggest a significant computational overhead. However, its computational burden is minimized due to its sparse activation criteria. PC is only engaged under two specific conditions: When the sliding context window is completely filled, and during the first diffusion time step ($t = T$). Even though the Top-C selection mechanism involves gathering and sorting tokens, our efficiency analysis justifies this operation in the Appendix E.

5. Experiments

5.1. Experimental settings

Implementation details. We implement chunk-wise Self Forcing [12] as our base model. We evaluate both quantitatively and qualitatively using Deep Sink (DS) combined with Participative Compression (PC). We set the hyperparameters for Deep Sink and Participative Compression as follows: sink size $S = 10$, budget $N = 16$, and recent $R = 4$. We compare against baseline autoregressive video diffusion models including CausVid [35], Self Forcing [12], Rolling Forcing [17], and LongLive [31].

Evaluation. We conduct evaluations under two settings. First, we evaluate long video generation on VBench-Long [13] using 128 prompts from MovieGen [19], following the same prompt selection protocol as Self Forcing++ [8]. Each prompt is refined using Qwen/Qwen2.5-7B-Instruct [30] following Self Forcing [12]. Second, we conduct a user preference study to evaluate color consistency, dynamic motion, subject consistency, and overall quality. Additional implementation details are provided in the Appendix G. Third, we evaluate visual stability using the state-of-the-art vision-language model (VLM) Gemini 2.5-Pro [7], following the protocol of Self Forcing++ [8].

5.2. Results in Long Video Generation

Quantitative results. Our quantitative results are presented in Table 1. Despite being a training-free method built upon Self Forcing, which was not trained for long video generation, our approach achieves performance comparable to methods explicitly distilled or trained for long videos [17, 31]. As shown in Table 1, we achieve superior performance in overall consistency and imaging quality compared to LongLive [31], and better aesthetic quality than Rolling Forcing [17].

Notably, our method also excels in dynamic degree, producing more dynamic motions than trained methods [17, 31], despite not being explicitly optimized for this aspect. We attribute this to our training-free approach, which avoids the potential motion constraints introduced when models are explicitly trained to anchor with attention sinks.

Qualitative results. The qualitative results in Figure 8 demonstrate strong visual quality, with our training-free method producing high-fidelity frames comparable to or better than training-based baselines. The results visually confirm our quantitative performance, where we achieve

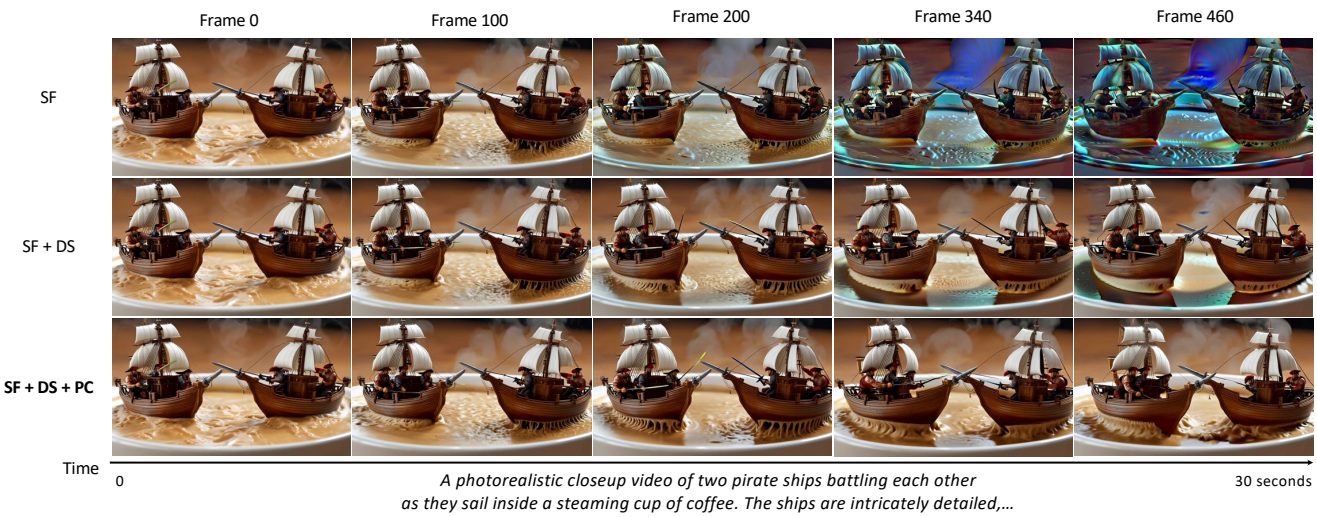

Figure 6. **Qualitative ablation results over 30-second generation:** Comparison of Self Forcing (SF) [12], SF with Deep Sink (SF+DS), and SF with both Deep Sink and Participative Compression (Deep Forcing). Baseline SF exhibits severe color drift. SF+DS improves stability but shows residual artifacts. Deep Forcing maintains consistent visual quality.

competitive scores without any fine-tuning. Notably, our videos exhibit more dynamic motion in both camera and subject movement, yielding more visually expressive results compared to existing approaches.

Although subject consistency is lower in VBench-Long metrics, the bottom example in Figure 8 demonstrates that our training-free approach maintains better overall quality with limited degradation compared to training-based methods. Additional qualitative results are provided in Appendix F.

User study. To further validate these observations, we conducted a user study with 24 participants evaluating multiple aspects of the generated videos. The user study was conducted following the Two-Alternative Forced Choice (2AFC) protocol, where users are instructed to choose which is better between two videos (from Deep Forcing and a baseline), in regard to color consistency, dynamic motion, subject consistency, and overall quality. As shown in Table 2, participants demonstrated a clear preference for Deep Forcing over the baselines across all evaluated aspects. This includes a high preference in terms of subject consistency, highlighting Deep Forcing's ability to retain the subject with minimal identity drift throughout the video. These corroborate our qualitative assessment that perceptual quality remains high despite lower VBench-Long [13] subject consistency scores.

VLM evaluation. For further comparison with the baselines, we evaluate visual stability using the state-of-the-art vision–language model (VLM) Gemini 2.5-Pro [7]. Following the protocol of Self Forcing++ [8], we use the same prompt to ask the VLM to score each 30-second video in terms of exposure stability and degradation. Then we normalize the resulting scores 100. As shown in Tab. 3, our training-free method achieves visual stability comparable to that of recent methods [17, 31].

Table 2. **User study results.** Values represent *percentage of votes favoring our Deep Forcing* over the baselines.

Method	Color Cons.	Dyn. Motion	Subject Cons.	Overall Quality
CausVid	98.9%	95.8%	96.8%	100%
Self Forcing	85.9%	86.9%	84.8%	87.9%
LongLive	71.2%	83.5%	72.2%	72.2%
Rolling Forcing	76.7%	76.7%	80.0%	78.9%

Table 3. **Visual stability compared with the baselines.** Methods are additionally categorized by whether they are trained with an attention sink.

Method	Attention Sink Training	Visual Stability
CausVid [35]	No	42.84
Self Forcing [12]	No	43.94
Rolling Forcing [17]	Yes	72.6
LongLive [31]	Yes	78.58
Deep Forcing (Ours)	No	75.44

5.3. Ablation studies

We conducted ablation studies to evaluate the contributions of each method. We measure relevant VBench-Long metrics on 30 second videos.

Effect of Deep Sink & Participative Compression. We evaluate three variants: naive Self-Forcing [12], Self-

Table 4. **Ablation on our components**: Deep Sink (DS) and Participative Compression (PC).

Method	Dynamic Degree	Overall Consistency	Image Quality
SF [12] (Baseline)	36.62	20.50	68.58
SF [12] + DS	48.58	20.54	68.54
SF [12] + DS + PC (**Ours**)	57.56	20.54	69.31

Forcing with only Deep Sink with sink length $S = 10$ frames, and Self-Forcing with both Deep Sink ($S = 10$ frames) and Participative Compression ($N = 16$, $R = 4$). As shown in Table 6, Deep Forcing demonstrates progressive improvements in dynamic degree, overall consistency, and image quality as components are added. Notably, dynamic degree improves substantially through the Deep Forcing framework, enabling the generation of significantly more dynamic scenes compared to baseline methods.

While image quality shows a slight decrease at 30 seconds, we have already demonstrated Deep Sink's positive impact on 50-second videos in Section 4.2.

Ablation Visualization. Figure 6 visualizes our ablation study results. When generating long videos with Self Forcing (SF) alone (top row), error accumulation leads to severe fidelity degradation and visual quality deteriorates as colors drift toward over-saturation. Adding Deep Sink (SF + DS) substantially reduces fidelity degradation and maintains more consistent colors. However, subtle artifacts persist at frame 460, including slight color shift in the coffee and texture blur in the ship details. When both Deep Sink and Participative Compression are applied (**Deep Forcing**), noticeable degradation is effectively eliminated. This validates that our complete framework successfully mitigates long-horizon error accumulation while preserving both overall visual quality and fine-grained details.

Top-C Visualization. Figure 7 visualizes a subset of Top-C tokens selected during the first rolling step, spatially aligned to their positions in Frame 37 when generating Frame 82. The yellow highlighted regions indicate the spatial positions of tokens selected as Top-C within the frame. These highlighted regions reveal semantic alignment with contextually important content: the robot's body and background architecture, the octopus tentacles and crab, and the circular coffee cup structure. This demonstrates that our method identifies and retains semantically salient regions critical for maintaining contextual coherence in subsequent generation.

6. Conclusion

We introduced Deep Forcing, a training-free approach for autoregressive long video generation that effectively mitigates error accumulation through two key components: **Deep Sink** and **Participative Compression**. Our method achieves state-of-the-art performance on VBench-Long, user studies, and VLM evaluation without any fine-tuning, even surpassing training-based methods. By exploiting the inherent deep attention sink behavior in pre-trained Self Forcing, we enable minute-long video generation while preserving both visual fidelity and motion dynamics. This training-free paradigm offers a practical and efficient solution for long video generation with autoregressive video diffusion.

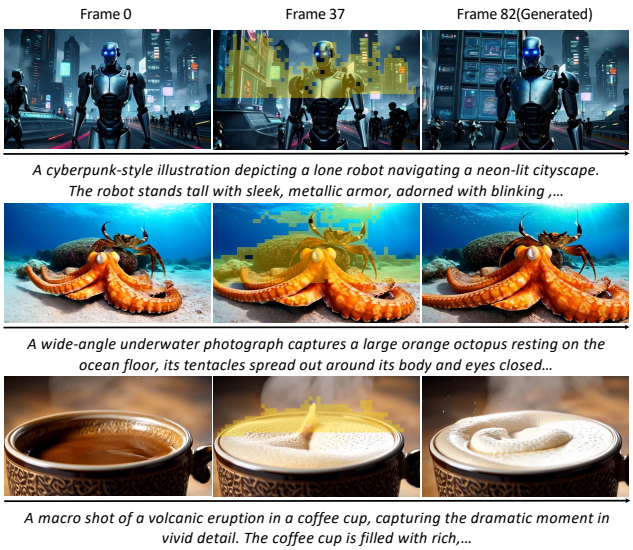

Figure 7. **Visualization of Top-C token selection.** For each example, Frame 37 **(middle)** shows the Top-C tokens selected for generating Frame 82 **(right)**. Yellow highlights indicate the spatial locations of tokens chosen as Top-C. Our method effectively identifies and preserves regions that are critical for maintaining contextual coherence during subsequent generation.

Limitations and Future Works. While our training-free approach substantially improves long-horizon stability, several limitations remain. Operating at inference time on a frozen backbone, our method is constrained by the pre-trained model's capacity and biases. Additionally, our approach lacks explicit long-term memory, potentially causing gradual drift in extremely long sequences with repeated occlusions. Future work could integrate hierarchical memory modules and extend to broader video generation settings.

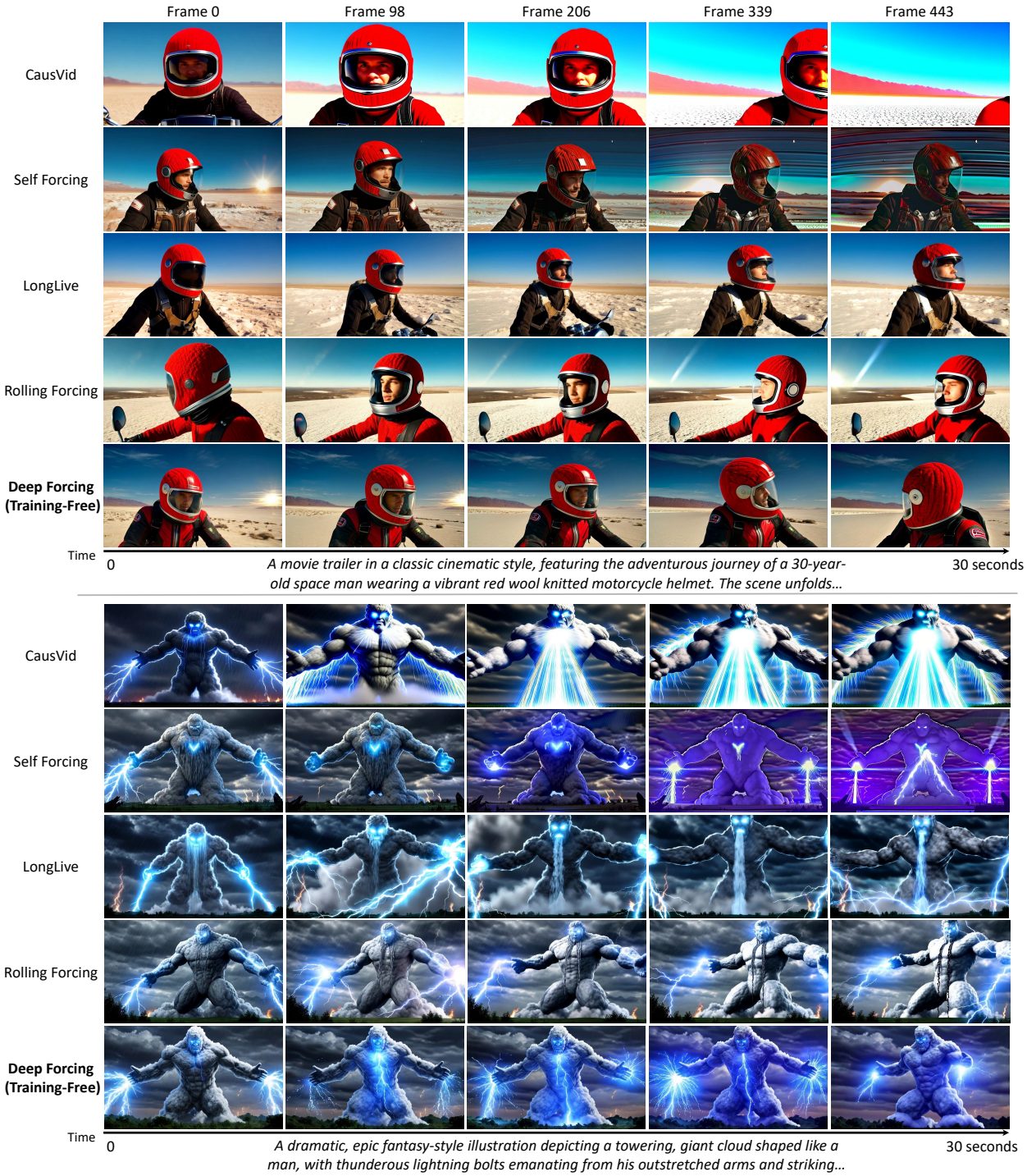

Figure 8. **Qualitative results on 30-second videos.** Frame-by-frame comparison across different methods for two representative prompts. Deep Forcing (training-free) achieves temporal consistency and visual quality comparable to training-based baselines (CausVid [35], Self Forcing [12], LongLive [31], Rolling Forcing [17]) while generating more dynamic content with greater subject consistency.

References

[1] Rtfm: A real-time frame model https://www.worldlabs.ai/blog/rtfm, 2025. 2

[2] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023. 5

[3] Hila Chefer, Uriel Singer, Amit Zohar, Yuval Kirstain, Adam Polyak, Yaniv Taigman, Lior Wolf, and Shelly Sheynin. Videojam: Joint appearance-motion representations for enhanced motion generation in video models. *arXiv preprint arXiv:2502.02492*, 2025. 5

[4] Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. *Advances in Neural Information Processing Systems*, 37:24081–24125, 2024. 2, 3

[5] Guibin Chen, Dixuan Lin, Jiangping Yang, Chunze Lin, Junchen Zhu, Mingyuan Fan, Hao Zhang, Sheng Chen, Zheng Chen, Chengcheng Ma, Weiming Xiong, Wei Wang, Nuo Pang, Kang Kang, Zhiheng Xu, Yuzhe Jin, Yupeng Liang, Yubing Song, Peng Zhao, Boyuan Xu, Di Qiu, Debang Li, Zhengcong Fei, Yang Li, and Yahui Zhou. Skyreels-v2: Infinite-length film generative model, 2025. 3

[6] Junsong Chen, Yuyang Zhao, Jincheng Yu, Ruihang Chu, Junyu Chen, Shuai Yang, Xianbang Wang, Yicheng Pan, Daquan Zhou, Huan Ling, et al. Sana-video: Efficient video generation with block linear diffusion transformer. *arXiv preprint arXiv:2509.24695*, 2025. 5

[7] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025. 7, 8

[8] 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. *arXiv preprint arXiv:2510.02283*, 2025. 2, 7, 8

[9] Ravi Ghadia, Avinash Kumar, Gaurav Jain, Prashant J. Nair, and Poulami Das. Dialogue without limits: Constant-sized kv caches for extended responses in llms. *ArXiv*, abs/2503.00979, 2025. 2, 3, 5

[10] Yoav HaCohen, Nisan Chiprut, Benny Brazowski, Daniel Shalem, Dudu Moshe, Eitan Richardson, Eran Levin, Guy Shiran, Nir Zabari, Ori Gordon, Poriya Panet, Sapir Weissbuch, Victor Kulikov, Yaki Bitterman, Zeev Melumian, and Ofir Bibi. Ltx-video: Realtime video latent diffusion. *arXiv preprint arXiv:2501.00103*, 2024. 2

[11] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. *ArXiv*, abs/1904.09751, 2019. 5

[12] 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. 2, 3, 4, 5, 7, 8, 9, 10, 6

[13] 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*, pages 21807–21818, 2024. 2, 4, 7, 8

[14] Yang Jin, Zhicheng Sun, Ningyuan Li, Kun Xu, Hao Jiang, Nan Zhuang, Quzhe Huang, Yang Song, Yadong Mu, and Zhouchen Lin. Pyramidal flow matching for efficient video generative modeling. *arXiv preprint arXiv:2410.05954*, 2024. 3

[15] LAION-AI. aesthetic-predictor. `https://github.com/LAION-AI/aesthetic-predictor`,. [Accessed 14-11-2025]. 5

[16] Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. Snapkv: Llm knows what you are looking for before generation. *arXiv preprint arXiv:2404.14469*, 2024. 2, 3, 5

[17] 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. 1, 2, 3, 4, 5, 7, 8, 10, 6

[18] Jack Parker-Holder and Shlomi Fruchter. Genie 3: A new frontier for world models, 2025. 2

[19] Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Animesh Sinha, Ann Lee, Apoorv Vyas, Bowen Shi, Chih-Yao Ma, Ching-Yao Chuang, et al. Movie gen: A cast of media foundation models. *arXiv preprint arXiv:2410.13720*, 2024. 5, 7

[20] Dachuan Shi, Yonggan Fu, Xiangchi Yuan, Zhongzhi Yu, Haoran You, Sixu Li, Xin Dong, Jan Kautz, Pavlo Molchanov, and Yingyan Celine Lin. Lacache: Ladder-shaped kv caching for efficient long-context modeling of large language models. In *Forty-second International Conference on Machine Learning*, 2025. 2

[21] Joonghyuk Shin, Zhengqi Li, Richard Zhang, Jun-Yan Zhu, Jaesik Park, Eli Schechtman, and Xun Huang. Motionstream: Real-time video generation with interactive motion controls. *arXiv preprint arXiv:2511.01266*, 2025. 2

[22] Kiwhan Song, Boyuan Chen, Max Simchowitz, Yilun Du, Russ Tedrake, and Vincent Sitzmann. History-guided video diffusion. *arXiv preprint arXiv:2502.06764*, 2025. 2

[23] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. 5

[24] Hansi Teng, Hongyu Jia, Lei Sun, Lingzhi Li, Maolin Li, Mingqiu Tang, Shuai Han, Tianning Zhang, WQ Zhang, Weifeng Luo, et al. Magi-1: Autoregressive video generation at scale. *arXiv preprint arXiv:2505.13211*, 2025. 3, 5

[25] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei

Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025. 2, 5

[26] Zhongwei Wan, Xinjian Wu, Yu Zhang, Yi Xin, Chaofan Tao, Zhihong Zhu, Xin Wang, Siqi Luo, Jing Xiong, Longyue Wang, et al. D2o: Dynamic discriminative operations for efficient long-context inference of large language models. *arXiv preprint arXiv:2406.13035*, 2024. 3

[27] Jing Wang, Fengzhuo Zhang, Xiaoli Li, Vincent YF Tan, Tianyu Pang, Chao Du, Aixin Sun, and Zhuoran Yang. Error analyses of auto-regressive video diffusion models: A unified framework. *arXiv preprint arXiv:2503.10704*, 2025. 2

[28] Yi Wang, Yinan He, Yizhuo Li, Kunchang Li, Jiashuo Yu, Xin Ma, Xinhao Li, Guo Chen, Xinyuan Chen, Yaohui Wang, et al. Internvid: A large-scale video-text dataset for multimodal understanding and generation. *arXiv preprint arXiv:2307.06942*, 2023. 5

[29] Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. *arXiv preprint arXiv:2309.17453*, 2023. 2, 3, 4, 5

[30] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. 7

[31] Shuai Yang, Wei Huang, Ruihang Chu, Yicheng Xiao, Yuyang Zhao, Xianbang Wang, Muyang Li, Enze Xie, Yingcong Chen, Yao Lu, et al. Longlive: Real-time interactive long video generation. *arXiv preprint arXiv:2509.22622*, 2025. 1, 2, 3, 4, 7, 8, 10, 5, 6

[32] Shuo Yang, Haocheng Xi, Yilong Zhao, Muyang Li, Jintao Zhang, Han Cai, Yujun Lin, Xiuyu Li, Chenfeng Xu, Kelly Peng, et al. Sparse videogen2: Accelerate video generation with sparse attention via semantic-aware permutation. *arXiv preprint arXiv:2505.18875*, 2025. 5

[33] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024. 2, 5

[34] Deheng Ye, Fangyun Zhou, Jiacheng Lv, Jianqi Ma, Jun Zhang, Junyan Lv, Junyou Li, Minwen Deng, Mingyu Yang, Qiang Fu, et al. Yan: Foundational interactive video generation. *arXiv preprint arXiv:2508.08601*, 2025. 2

[35] 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 Computer Vision and Pattern Recognition Conference*, pages 22963–22974, 2025. 2, 3, 5, 7, 8, 10, 6

[36] Lvmin Zhang and Maneesh Agrawala. Packing input frame context in next-frame prediction models for video generation. *arXiv preprint arXiv:2504.12626*, 2025. 5

[37] Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, et al. H2o: Heavy-hitter oracle for efficient generative inference of large language models. *Advances in Neural Information Processing Systems*, 36: 34661–34710, 2023. 3, 5

Appendix

In this appendix, we provide comprehensive details, including:

- Comparison with other sink mechanisms [17, 31] (Appendix A)
- Qualitative results on different sink sizes (Appendix B)
- Participative Compression details (Appendix C)
- Denoising query is not just noise (Appendix D)
- FPS measurements (Appendix E)
- More qualitative results (Appendix F)
- User study protocol (Appendix G)
- Additional attention visualization (Appendix H)

A. The Tale of Three Sinks

Concurrent works [17, 31] propose different attention sink mechanisms for Self Forcing-like architectures, typically with model training or distillation. In this section, we compare these approaches in a training-free setting to evaluate their effectiveness when applied directly to pretrained Self Forcing model.

Specifically, we compare three attention sink strategies: LongLive [31], Rolling Forcing [17], and Ours. LongLive applies attention sinks to the first **3 frames** without RoPE adjustment. Rolling Forcing also uses **3 frame** sinks but incorporates (1) storing raw keys and (2) dynamically reapplying RoPE when rolling occurs. Qualitative comparisons are presented in Figure 9.

Figure 9 shows that our method achieves substantially better results. The LongLive attention sink, which does not adjust RoPE, exhibits progressive failure modes: **fidelity degradation** appears at frame 800, followed by **flickering** artifacts at frame 801, and culminating in **roll-back** behavior at frame 802 where the generation reverts to early sinked frames. These issues also occur in LongLive [31], which was explicitly trained on long videos using this attention sink mechanism (Fig. 1).

Although Rolling Forcing attention sink [17] employs Dynamic RoPE, which reapplies positional encodings to the entire cached key set, it still exhibits severe fidelity degradation at frames 800–801.

This comparison demonstrates that both deep sink and RoPE adjustment are essential for long video generation.

B. Qualitative Results on Different Sink Size

Figure 10 presents qualitative comparisons across sink sizes ranging from 0 to 18 frames, as analyzed in Section 4.2. We evaluate two diverse prompts on 60-second generation.

With no attention sink (Sink 0), severe fidelity degradation emerges rapidly, where the monster's texture deteriorates and colors shift noticeably by frame 230, with complete quality collapse by frame 690. Similarly, the SUV scene exhibits significant fidelity degradation. As the sink size increases to 4 and 9 frames, degradation is progressively reduced but remains visible in fine details.

Once the sink size exceeds 10 frames (Sink 14), fidelity degradation is substantially reduced. However, excessively large sinks (Sink 18) exhibit repetitive generation where early frames are over-preserved.

These results validate our optimal sink range of 10–15 frames (40–60% of the sliding window). While Deep Sink substantially mitigates degradation, it alone proves insufficient to maintain visual fidelity throughout minute-long generation across diverse scenes, as demonstrated in our extended evaluations (Section 5.3).

C. Participative Compression Details

In this section, we provide additional details and analysis of Participative Compression (PC) beyond what was presented in Section 4.3.

Each layer maintains its own KV cache, which undergoes compression as follows:

$$\begin{aligned} \left[K_{\text{sink}} \parallel K_{\text{cand}} \parallel K_{\text{rct}} \right] &\rightarrow \left[K_{\text{sink}} \parallel K_{\text{top}} \parallel K_{\text{rct}} \right], \\ \left[V_{\text{sink}} \parallel V_{\text{cand}} \parallel V_{\text{rct}} \right] &\rightarrow \left[V_{\text{sink}} \parallel V_{\text{top}} \parallel V_{\text{rct}} \right]. \end{aligned} \tag{12}$$

PC compresses only the intermediate tokens between the sink and recent. This design preserves both the initial context ($K_{\text{sink}}, V_{\text{sink}}$) and recent context ($K_{\text{rct}}, V_{\text{rct}}$) without modification, while compressing only the candidate tokens ($K_{\text{cand}}, V_{\text{cand}}$) to retain the most important visual and contextual information ($K_{\text{top}}, V_{\text{top}}$).

This compression occurs independently in each layer. Importantly, PC is applied only at the first diffusion timestep ($t = 1000$) when the cache reaches over its maximum window length. The tokens selected at this initial timestep remain fixed throughout subsequent denoising steps.

Figure 11 validates this design choice by demonstrating that attention patterns remain consistent across timesteps. The tokens deemed important when generating frames 19–21 exhibit similar importance scores across different diffusion timesteps ($t = 1000, 750, 500, 250$), confirming that the Top-C selection at $t = 1000$ captures tokens that remain contextually relevant throughout the entire denoising process.

Participative Compression Ablation. As shown in Figure 3 in the main paper, Participative Compression (PC) can leverage both current denoising query tokens and clean past query tokens from previously generated frames to select the Top-C candidates. We evaluate the effect of using each type independently versus combining them together.

Table 5 compares these three strategies. When Top-C is selected using only clean past tokens (Only Past), the

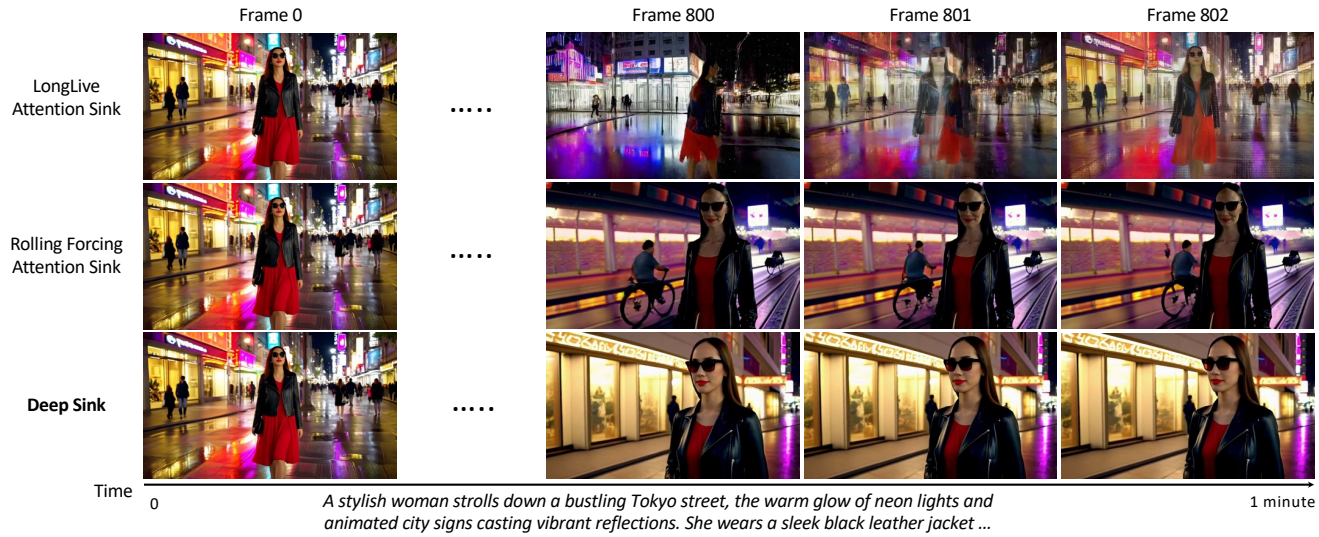

Figure 9. **Qualitative results on different Attention Sink.** The result shows that Deep Sink substantially outperforms both LongLive-style [31] and Rolling Forcing-style [17] attention sinks.

Table 5. **Ablation on Participative Compression.**

Method	Motion Smoothness	Overall Consistency	Image Quality
Only Denoising	97.86	20.44	68.24
Only Past	97.91	20.47	68.54
Both	98.27	20.54	69.31

method achieves an image quality of 68.54 and overall consistency of 20.47. When selection relies solely on currently denoising tokens (Only Denoising), the noisy nature of these queries at the initial timestep ($t = 1000$) leads to slightly lower image quality (68.24) and motion smoothness (97.86), likely due to unstable token selection at the initial denoising step. Combining both query types (Both) achieves the highest scores across all metrics, including motion smoothness, overall consistency, and image quality. The clean past queries appear to provide relatively stable importance estimates, while the current denoising queries help ensure the selected tokens remain relevant to the immediate generation context, suggesting complementary benefits from their combination.

D. Denoising Query Is Not Just Random Noise

While the denoising queries at the initial timestep ($t = 1000$) are inherently noisy, they may still carry meaningful signal for identifying important tokens. Figure 11 suggests this by showing consistent attention patterns across timesteps, but to more conclusively demonstrate the effectiveness of noisy queries, we directly compare Top-C selection based on denoising queries versus Gaussian random selection. For denoising query-based selection, we compute QK^T using only the currently denoising query tokens, then select the Top-C candidates. For Gaussian random selection, we assign each candidate token a score sampled from $\mathcal{N}(0, 1)$ and select the Top-C based on these random scores.

Figure 12 illustrates the stark difference. Random selection exhibits severe scene repetition and context loss, as randomly chosen anchors fail to preserve coherent contextual information. In contrast, denoising query-based selection generates context-aware videos with notably better subject consistency and context.

Figure 13 further validates this through token selection frequency heatmaps over 1-minute generation, where color intensity (white to dark purple) indicates selection frequency. The x-axis spans tokens 0–32,760: tokens 0–15,600 are Deep Sink, 15,600–28,080 are compression candidates, and 28,080+ are recent. Gaussian random selection (top) distributes uniformly across candidates, while denoising query-based selection (bottom) concentrates heavily on specific positions, particularly immediately after the sink boundary (15,600).

Notably, these high-frequency positions do not correspond to fixed frames, as tokens are evicted during compression, subsequent tokens shift into these slots. The concentration at positions near 15,600 indicates that these positional slots are consistently selected regardless of frame identity, as they bridge established context (sink) and current generation (recent), serving as semantically important anchors. This positional selectivity demonstrates meaningful contextual relationships rather than arbitrary noise.

We hypothesize this effectiveness stems from: (1) Self Forcing's 4-step distilled diffusion enabling rapid convergence to meaningful attention patterns despite noisy queries at $t = 1000$, and (2) per-layer KV caching allowing independent selection of semantically important tokens based

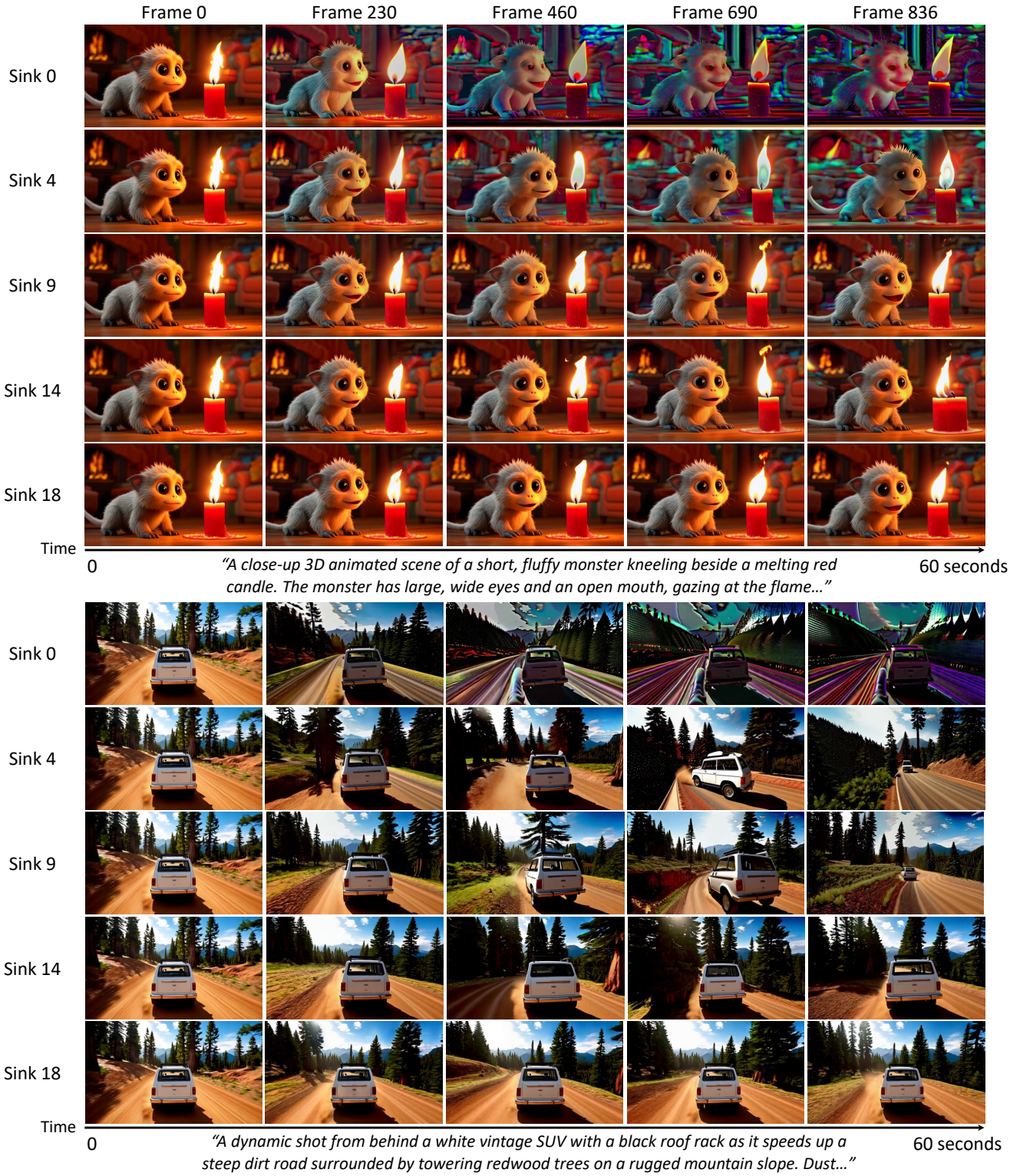

Figure 10. **Qualitative comparison of different sink sizes on 60-second videos.** As the sink size decreases, degradation becomes more severe. Once the sink size exceeds 10 frames, degradation is substantially reduced.

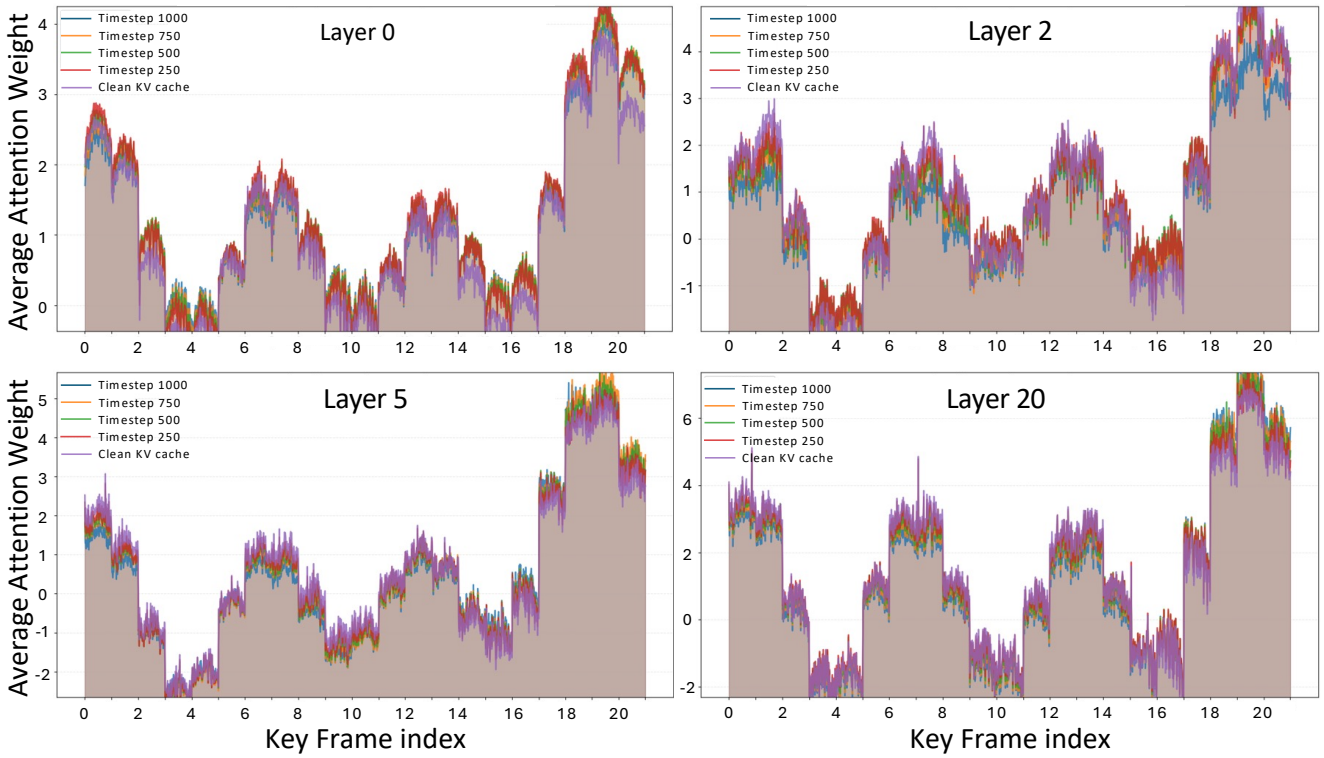

Figure 11. **Attention weight consistency across diffusion timesteps.** Query-averaged attention weights showing how each key frame is attended when generating the last chunk (frames 19–21) at different denoising timesteps. The consistent attention patterns across timesteps (1000, 750, 500, 250, and the final clean KV cache) demonstrate that Top-C tokens selected at the initial timestep ($t = 1000$) remain valid and contextually relevant throughout the entire denoising process.

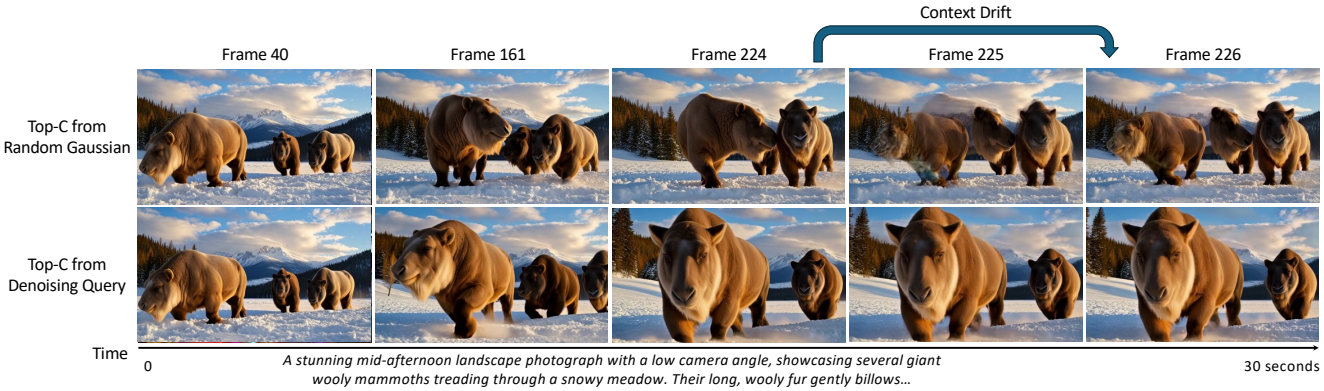

Figure 12. **Qualitative comparison: Random Top-C vs. Denoising Query Top-C.** Gaussian random selection causes severe artifacts during compression - faces abruptly rotate, heads appear floating in mid-air, and random context drift occurs, resulting in incoherent scene transitions. In contrast, denoising query-based selection maintains subject consistency with natural emergent camera movements and preserves contextual coherence throughout the generation.

on layer-specific contextual relevance.

E. FPS measurements

We evaluate inference throughput on a single NVIDIA H100 GPU when generating 60-second videos. Table 6 demonstrates that Deep Forcing maintains throughput comparable to baseline Self Forcing, achieving 15.75 FPS versus 15.78 FPS. Despite the computational overhead of compression, our method balances two competing factors: (1) compressing from 21 to 16 frames requires additional computation, but (2) generating subsequent frames with only 16 cached frames incurs lower attention costs compared to full

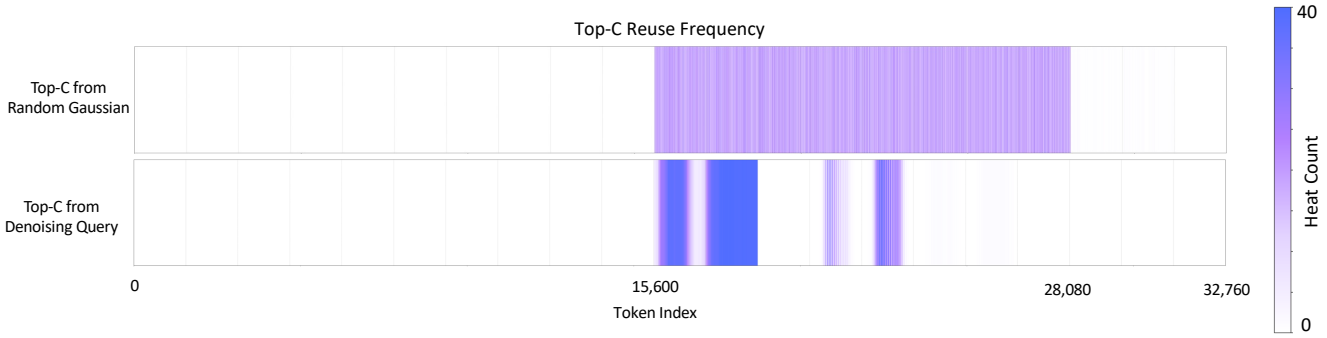

Figure 13. **Token-wise Top-C selection frequency heatmap during 1-minute generation.** Color intensity ranges from white (rarely selected) to dark purple (frequently selected as Top-C), indicating how often each token is reused throughout the generation. The x-axis spans tokens 0–32,760, where 0–15,600 are Deep Sink tokens, 15,600–28,080 are candidates for compression, and 28,080+ are recent tokens. Gaussian random selection (top) distributes selections uniformly across candidate tokens, whereas denoising query-based selection (bottom) concentrates heavily on specific semantically important tokens—particularly those immediately after the sink boundary—that effectively bridge established and newly formed context.

Table 6. **Throughput Comparison on a single H100 GPU. Latency is measured after first rolling.**

Method	FPS	Latency(Min/Max)
Self Forcing [12]	15.78	0.770 / 0.776s
Deep Forcing (Ours)	15.75	0.747 / 0.797s

attention over 21 frames.

The latency range after first rolling in Table 6 reflects this trade-off. While Deep Forcing exhibits a wider latency range (0.747s to 0.797s) compared to the baseline (0.770s to 0.776s), the average latencies are nearly identical, demonstrating that compression overhead is effectively balanced by reduced attention costs. In practice, throughput oscillates between compression phases (slightly slower) and generation phases (slightly faster) as the cache alternates between 21 and 16 frames. These fluctuations average to nearly identical performance as the baseline, demonstrating that our compression mechanism effectively amortizes its overhead, enabling long-horizon generation with minimal performance penalty.

F. More Qualitative Results

Additional qualitative results of our method are presented in Figure 14, and Figure 15. These examples clearly show that our **training-free** Deep Sink and Participative Compression framework produces results on par with training-based methods.

G. User Study Protocol

To perform human evaluation, we conducted a user study based on the Two-Alternative Forced Choice (2AFC) protocol [2, 3]. For each question, participants were presented with two videos generated from the same prompt and instructed to choose which video they preferred according to four evaluation criteria: (1) Color Consistency - Which video maintains more consistent color and exposure throughout, without sudden shifts in brightness, saturation, or color tone? (2) Dynamic Motion - Which video exhibits more dynamic and varied motion, including both subject movement and camera movement? (3) Subject Consistency - Which video maintains better visual consistency of the main subject throughout its duration? (Consider comparing the beginning and end of each video.) (4) Overall Quality - Overall, which video appears more realistic, natural, and of higher quality?

Each participant evaluated 16 video pairs comparing Deep Forcing against each of the four baselines (CausVid [35], Self Forcing [12], LongLive [31], and Rolling Forcing [17]). For each baseline, participants were shown 4 pairwise comparisons using different prompts, with all 16 prompts being non-overlapping within each participant's session. These prompts were randomly sampled from a pool of 20 total prompts. With 24 total participants, this yielded 384 total video comparisons (24 participants $\times$ 16 pairs), the results of which are shown in Table 2. The presentation order of videos was randomized, and participants were not informed which model generated each video. This design ensured balanced evaluation across all baseline models while avoiding prompt repetition within individual sessions. The user interface is shown in Figure 17.

H. Additional Attention Visualization

We provide additional attention head visualizations (Fig.16) beyond those shown in Fig.4 from Section 4.2. This deep attention pattern with substantial weight on both initial and intermediate tokens, emerges consistently and pervasively across layers and heads, rather than being only one or two specific heads, supporting the hypothesis that deep sinks are fundamental to Self Forcing [12].

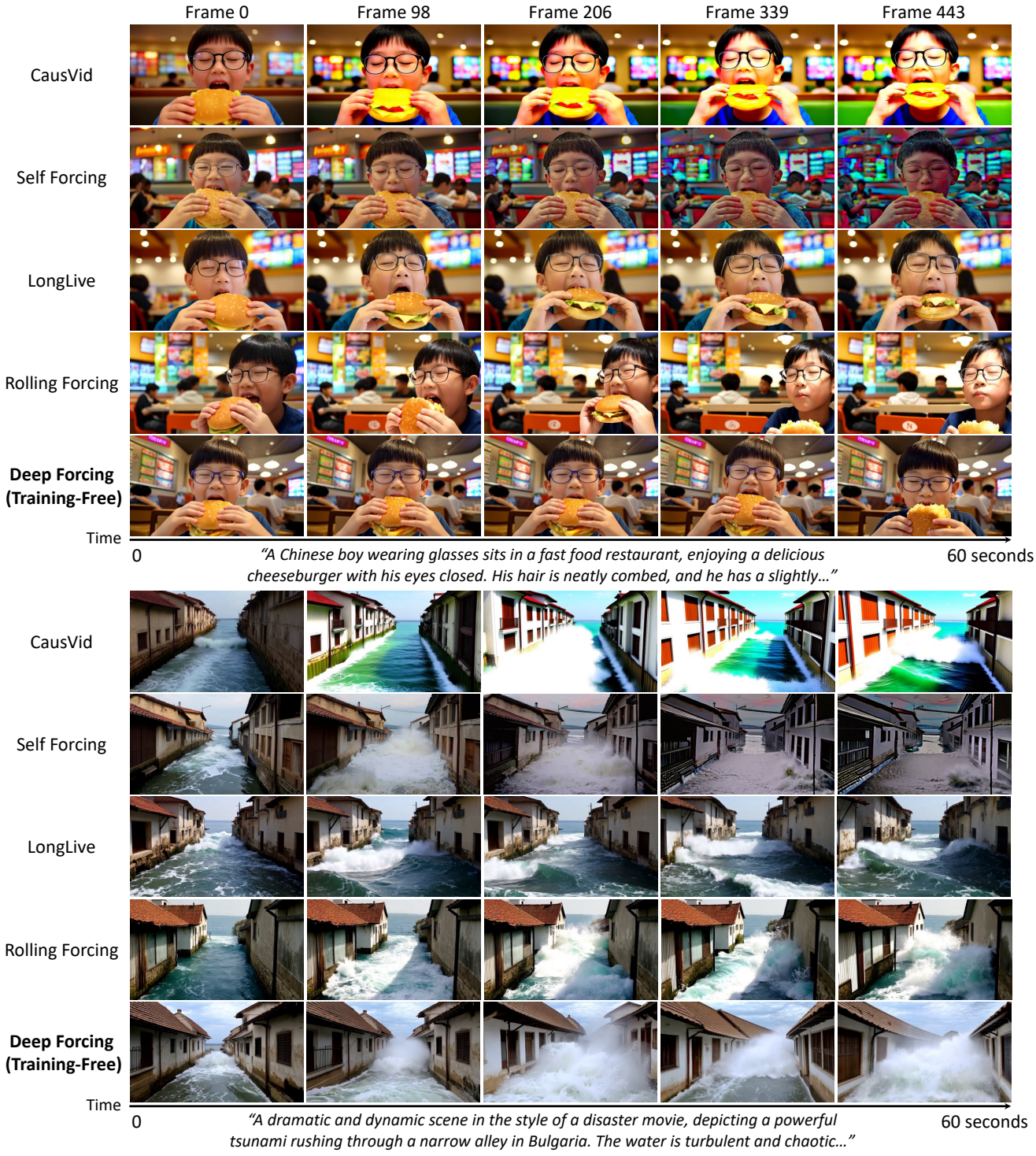

Figure 14. **Qualitative results on 30-second videos.** Frame-by-frame comparison across different methods for two representative prompts. Deep Forcing (training-free) achieves temporal consistency and visual quality comparable to training-based baselines (CausVid [35], Self Forcing [12], LongLive [31], Rolling Forcing [17]) while generating more dynamic content with greater subject consistency.

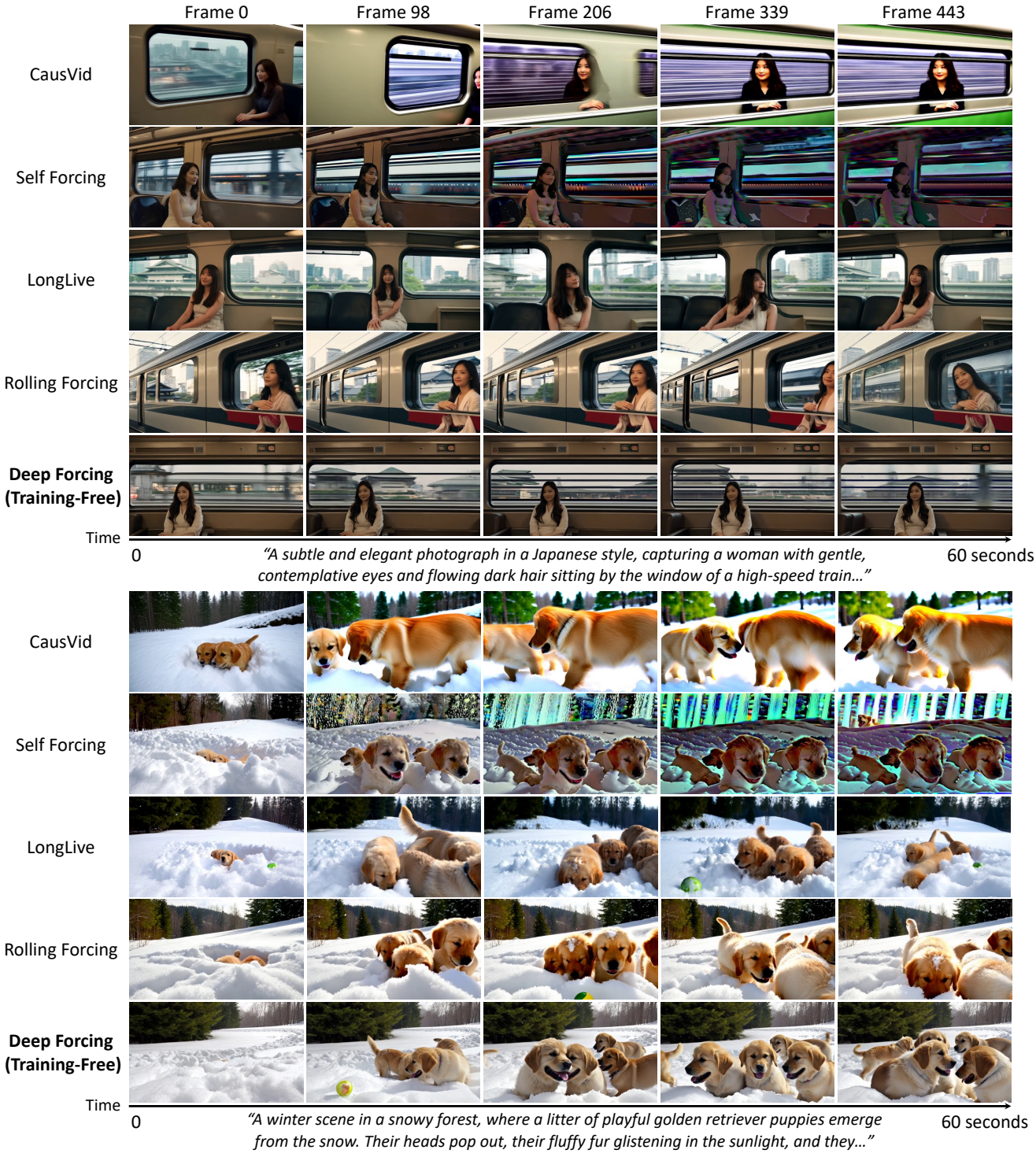

Figure 15. **Qualitative results on 60-second videos.** Frame-by-frame comparison across different methods for two representative prompts. Deep Forcing (training-free) achieves temporal consistency and visual quality comparable to training-based baselines (CausVid [35], Self Forcing [12], LongLive [31], Rolling Forcing [17]) while generating more dynamic content with greater subject consistency.

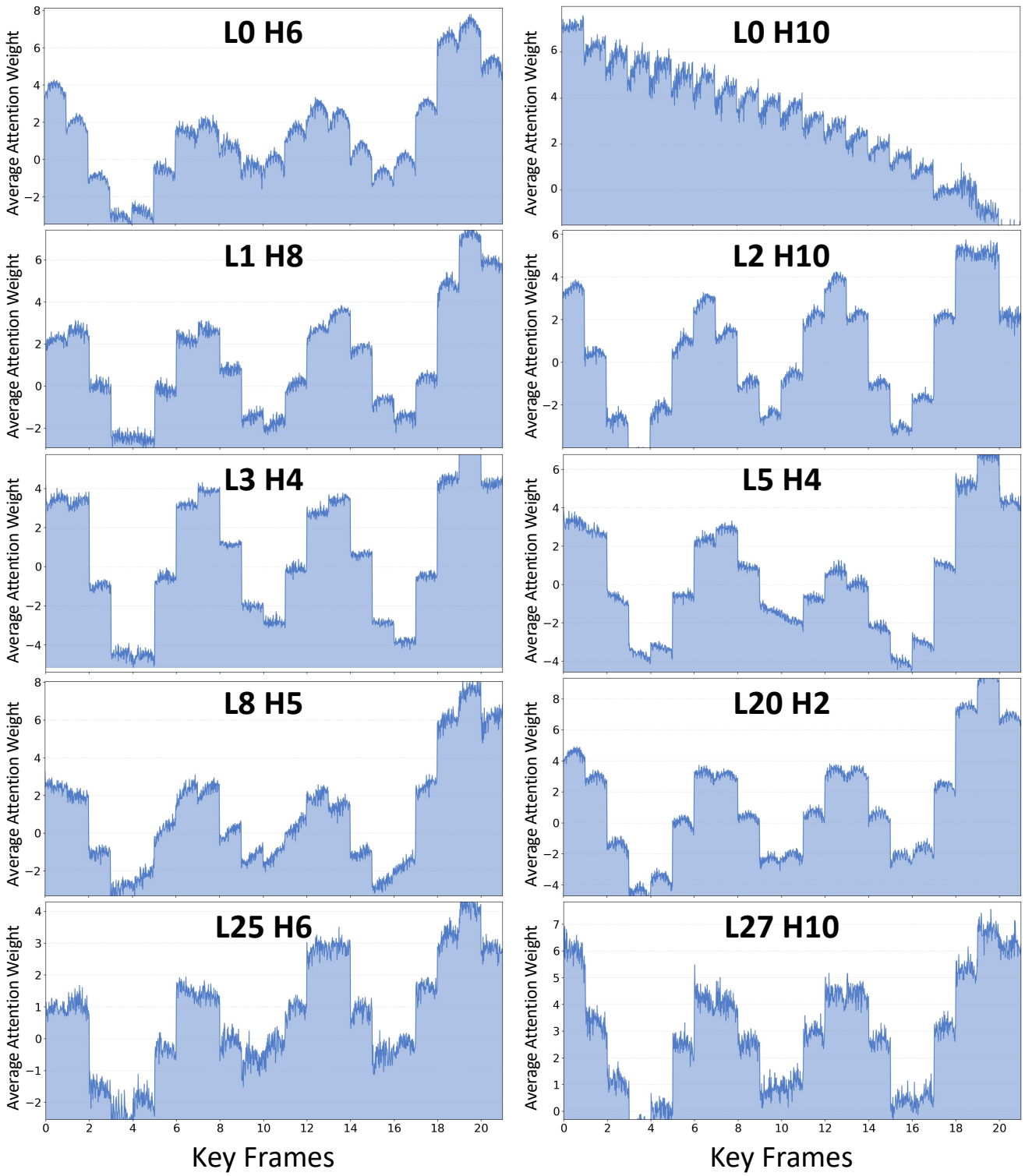

Figure 16. **Attention weight distribution across earlier frames.** Query-averaged attention showing how the last chunk (frames 19-21) attends to earlier KV cache entries (frames 0-18). Each frame consists of 1,560 tokens (spatially arranged latent patches). We visualize multiple attention heads from different layers, demonstrating that substantial attention to intermediate tokens is consistent across layers and heads.

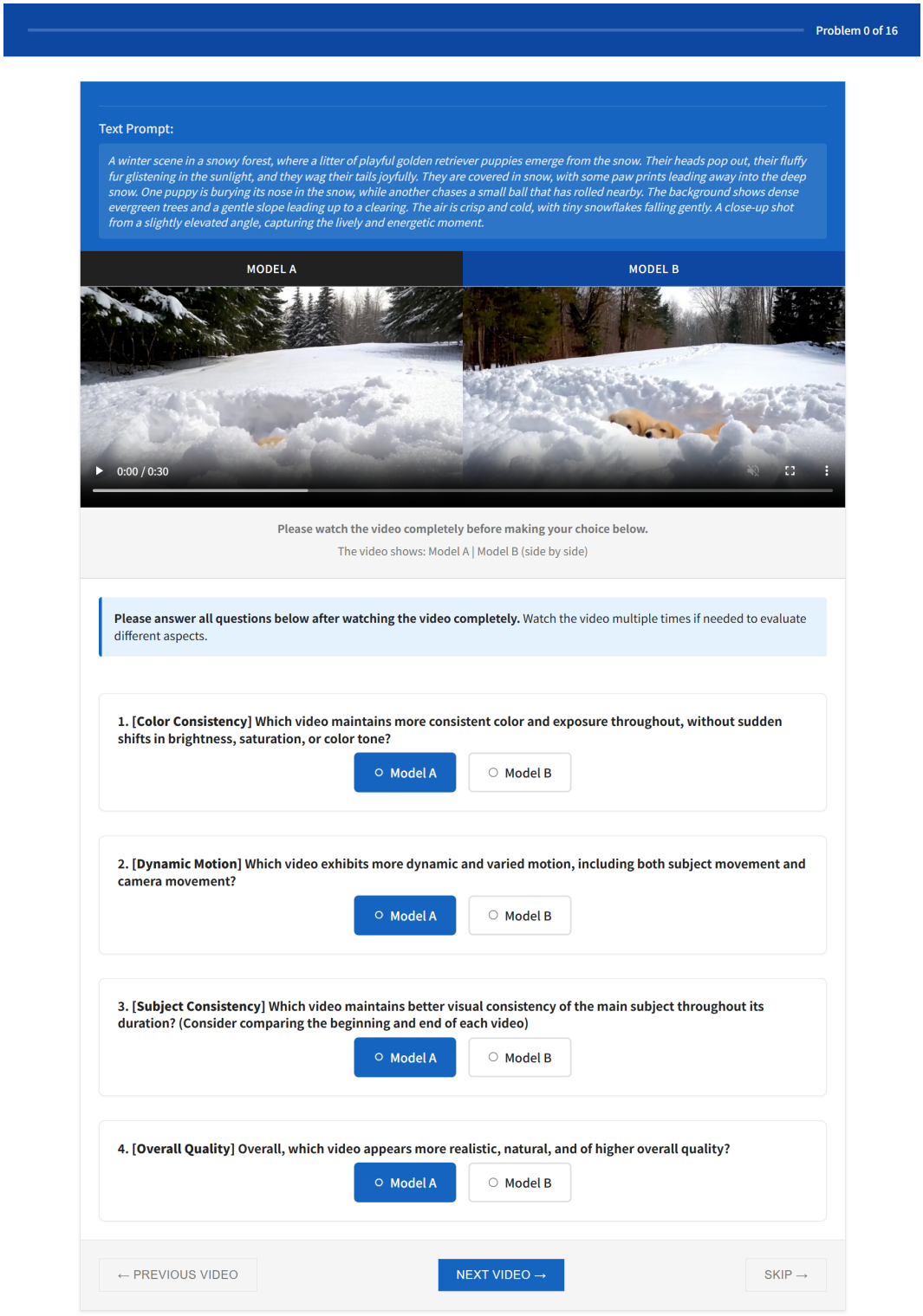

Figure 17. **Example of the user interface for the user study.**