PackForcing: Short Video Training Suffices for Long Video Sampling and Long Context Inference

Xiaofeng Mao[1,2†], Shaohao Rui[1,3,4†], Kaining Ying[1], Bo Zheng[1], Chuanhao Li[1], Mingmin Chi[1‡], and Kaipeng Zhang[1,3‡]

[1] Alaya Lab
[2] Fudan University
[3] Shanghai Innovation Institute
[4] Shanghai Jiao Tong University
kaipeng.zhang@shanda.com

Github: https://github.com/ShandaAI/PackForcing

Abstract. Autoregressive video diffusion models have demonstrated remarkable progress, yet they remain bottlenecked by intractable linear KV-cache growth, temporal repetition, and compounding errors during long-video generation. To address these challenges, we present **PackForcing**, a unified framework that efficiently manages the generation history through a novel three-partition KV-cache strategy. Specifically, we categorize the historical context into three distinct types: (1) **Sink tokens**, which preserve early anchor frames at full resolution to maintain global semantics; (2) **Mid tokens**, which achieve a massive spatiotemporal compression (∼32× token reduction) via a dual-branch network fusing progressive 3D convolutions with low-resolution VAE re-encoding; and (3) **Recent tokens**, kept at full resolution to ensure local temporal coherence. To strictly bound the memory footprint without sacrificing quality, we introduce a dynamic top-k context selection mechanism for the mid tokens, coupled with a continuous Temporal RoPE Adjustment that seamlessly re-aligns position gaps caused by dropped tokens with negligible overhead. Empowered by this principled hierarchical context compression, PackForcing can generate coherent 2-minute, 832×480 videos at 14–16 FPS on a single H200 GPU. It achieves a bounded KV cache enables a remarkable **24**× temporal extrapolation (5 s → 120 s), operating effectively trained on merely 5-second clips. Extensive results on VBench demonstrate state-of-the-art temporal consistency (26.07) and dynamic degree (56.25), proving that short-video supervision is sufficient for high-quality, long-video synthesis.

Keywords: Autoregressive Video Generation, Long Video Generation, KV Cache Management, Video Diffusion Model

† Equal contribution.
‡ Corresponding authors.

1 Introduction

Recent video diffusion models [1, 3, 5, 6, 9–11, 14, 26, 32–34] have demonstrated significant progress in high-fidelity and complex motion synthesis for short clips (5–15 s). However, their bidirectional architectures typically require the simultaneous processing of all frames within a spatiotemporal volume. This computationally intensive paradigm hinders the development of streaming or real-time generation.

Autoregressive video generation [4, 16, 40] addresses this limitation by employing a block-by-block generation strategy. Instead of computing the entire sequence jointly, these methods sequentially cache key-value (KV) pairs from previously generated blocks to provide continuous contextual conditioning. While this approach theoretically mitigates the memory bottlenecks of joint processing and enables unbounded-length video generation, its practical application for minute-scale generation is limited by two primary challenges:
(1) Error accumulation. Small prediction errors compound iteratively during the autoregressive denoising process, leading to progressive quality degradation and semantic drift. Although Self-Forcing [16] attempts to mitigate this by training on self-generated historical frames, it still suffers from severe error accumulation beyond its training horizon. Consequently, it exhibits a significant decline in text-video alignment: within 60 s, the model gradually loses the prompt's semantics, with its CLIP score dropping from 33.89 to 27.12 (Table 2).
(2) Unbounded memory growth. The KV cache scales linearly with the length of the generated video. For a 2-minute, 832×480 video at 16 FPS, the full attention context grows to ∼749K tokens, requiring ∼138 GB of KV storage across 30 transformer layers, well beyond the memory budget of a single commodity GPU. Standard workarounds, such as history truncation [40] or sliding windows [20], severely compromise long-range coherence. Even recent advanced baselines struggle with this bottleneck. For instance, DeepForcing introduces attention sinks and participative compression to retain informative tokens based on query importance. However, to prevent unbounded KV cache expansion, it ultimately relies on aggressive buffer truncation, leading to the irreversible loss of intermediate historical memory.

A fundamental dilemma thus emerges in autoregressive video generation: mitigating error accumulation requires an extensive contextual history, yet unbounded KV cache growth inevitably forces the discarding of critical memory under hardware constraints. Maintaining a large effective context window while strictly bounding the KV cache size remains a critical open problem.

Building upon DeepForcing's insights, we recognize the effectiveness of its deep sink and participative compression mechanisms in identifying and retaining crucial historical context. However, rather than irreversibly dropping unselected intermediate tokens to save memory, we propose efficiently compressing them. To this end, we introduce **PackForcing**, a unified framework comprehensively addressing both challenges via a principled three-partition KV cache design.

To this end, we introduce **PackForcing**, a unified framework that comprehensively addresses both error accumulation and memory bottlenecks via a prin-

cipled three-partition KV cache design. Specifically, our framework categorizes the historical context into: **(1) Sink tokens**, which preserve early anchor frames at full resolution to maintain global semantics and prevent drift; **(2) Compressed mid tokens**, which undergo a 128× spatiotemporal volume compression (via a dual-branch network) to efficiently retain the bulk of the historical memory; and **(3) Recent and current tokens**, which are kept at full resolution to ensure fine-grained local coherence.

This hierarchical design successfully bounds memory requirements while preserving critical information. To strictly limit the capacity of the compressed mid-buffer, we adapt dynamic context selection as an advanced top-k selection strategy, retrieving only the most informative mid tokens during generation. To resolve the ensuing positional discontinuities caused by managing unselected tokens, we introduce a novel incremental RoPE rotation that gracefully corrects temporal positions without requiring a full cache recomputation.

In a nutshell, our primary contributions are summarized as follows:

- **Three-partition KV cache.** We propose **PackForcing**, which partitions generation history into sink, compressed, and recent tokens, bounding per-layer attention to ∼27,872 tokens for any video length.
- **Dual-branch compression.** We design a hybrid compression layer fusing progressive 3D convolutions with low-resolution re-encoding. This achieve a 128× spatiotemporal compression (∼32× token reduction) for intermediate history, increasing effective memory capacity by over 27×.
- **Incremental RoPE rotation & Dynamic Context Selection.** We introduce a temporal-only RoPE adjustment to seamlessly correct position gaps during memory management. Alongside an importance-scored top-k token selection strategy, this ensures highly stable generation over extended horizons.
- **24× temporal extrapolation.** Trained exclusively on 5-second clips, PackForcing successfully generates coherent 2-minute videos. It achieves state-of-the-art VBench scores and demonstrates the most stable CLIP trajectory among all compared methods.

2 Related Work

Video Diffusion Models. Early video models inflated 2D U-Nets with pseudo-3D modules [1,13,14,29,30]. Recently, Diffusion Transformers (DiTs) [2,24] have emerged as the dominant architecture, treating videos as spatiotemporal patches to enable scalable 3D attention in state-of-the-art models (e.g., CogVideoX [37], Movie Gen [26], Wan [33], Open-Sora [42]). Concurrently, Flow Matching [19,21] has largely replaced standard diffusion to offer faster convergence. Despite these advances, current models primarily generate short clips (5–10 s). Joint spatiotemporal modeling for minute-level videos remains computationally intractable, as full 3D attention incurs a quadratic $\mathcal{O}((THW)^2)$ memory cost. This bottleneck directly motivates the need for memory-efficient, autoregressive long-video generation strategies.

Autoregressive Video Generation. Autoregressive video generation overcomes the fixed-length limitations of joint spatiotemporal modeling by synthesizing frames block-by-block and maintaining historical context via key-value (KV) caching. Recent methods have rapidly evolved this paradigm, exploring ODE-based initialization [40], self-generated frame conditioning [16], rolling temporal windows [20], long-short context guidance [36], and enlarged attention sinks [39]. Despite these innovations, existing approaches universally lack a mechanism to explicitly compress the KV cache. Consequently, they face a rigid trade-off: retaining the full history inevitably causes out-of-memory failures for videos exceeding roughly 80 seconds, whereas truncating the context buffer results in an irreversible loss of long-range coherence. PackForcing explicitly breaks this memory-coherence trade-off by introducing learned spatiotemporal token compression tailored for causal video generation.

KV Cache Management. KV cache management has been extensively studied in Large Language Models (LLMs) to enable long-context understanding. Representative techniques include retaining initial attention sinks [35], selecting heavy-hitter keys based on attention scores [41], and extending context via RoPE interpolation [25]. However, these methods primarily focus on token selection or eviction rather than explicit compression, as text representations are already highly compact. Video tokens, conversely, encode dense spatiotemporal grids characterized by massive inter-frame redundancy. Exploiting this unique structural redundancy motivates our learned 128× volume compression, achieving a memory reduction far beyond what token selection alone can provide.

Long Video Generation. Beyond purely autoregressive caching, traditional long video generation strategies often rely on modifying the inference noise scheduling [8, 27], designing hierarchical planning frameworks [15], or utilizing complex multi-stage extensions [12]. Infinity-RoPE [38] enables training-free, controllable infinite-length video generation by modifying 3D-RoPE and the KV cache, while Reward-Forcing [23] improves long-term consistency and motion quality in streaming video generation through EMA-Sink and reward-driven distillation. **Compared with these methods**, PackForcing operates within a unified, single-stage causal framework. By managing the historical context through hierarchical compression and position-corrected eviction, our method achieves the generation of arbitrarily long videos with strictly bounded memory footprint and constant-time attention cost.

3 Method

We first introduce the background on flow matching and causal KV caching (Sec. 3.1), then present the core components of PackForcing: the three-partition KV cache (Sec. 3.2), dual-branch compression (Sec. 3.3), Dual-Resolution Shifting with incremental RoPE adjustment (Sec. 3.4), and Dynamic Context Selection (Sec. 3.5).

3.1 Preliminaries

Flow Matching. Our base model builds upon the flow matching framework [19]. Given a clean video latent $\mathbf{x}_0$ and standard Gaussian noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, the noisy

latent at noise level $\sigma \in [0, 1]$ is constructed as:

$$\mathbf{x}_\sigma = (1 - \sigma)\,\mathbf{x}_0 + \sigma\,\boldsymbol{\epsilon}\,. \tag{1}$$

A neural network f_θ is trained to predict the velocity field $\mathbf{v}_\theta(\mathbf{x}_\sigma, \sigma) \approx \boldsymbol{\epsilon} - \mathbf{x}_0$.

KV Caching. A video sequence of T latent frames is partitioned into non-overlapping blocks, each containing B_f frames. Each block i, denoted as $\mathbf{z}_i \in \mathbb{R}^{B_f \times C \times H \times W}$ (where C, H, and W represent the channel, height, and width dimensions, respectively), is generated autoregressively. After spatial patchification, each block yields $n{=}B_f hw$ tokens, where h and w represent the spatial height and width after patchification.

During the generation of block i, each transformer layer l attends to the Key-Value (KV) pairs cached from all previously generated blocks:

$$\mathcal{C}^l = \big\{(\mathbf{K}_j^l,\, \mathbf{V}_j^l)\big\}_{j=1}^{i-1}, \tag{2}$$

where $\mathbf{K}_j^l, \mathbf{V}_j^l \in \mathbb{R}^{n \times N_h \times d_h}$, with N_h representing the number of attention heads and d_h denoting the head dimension.

The attention operation for the current block i concatenates these historical keys and values with its own:

$$\mathrm{Attn}(\mathbf{Q}_i,\, \mathcal{C}^l) = \mathrm{softmax}\Big(\frac{\mathbf{Q}_i\,\mathbf{K}_{1:i}^\top}{\sqrt{d_h}}\Big)\mathbf{V}_{1:i}\,, \tag{3}$$

where $\mathbf{Q}_i$ is the query matrix for block i, while $\mathbf{K}_{1:i}$ and $\mathbf{V}_{1:i}$ represent the concatenated keys and values from block 1 to i.

As generation proceeds, the KV cache grows linearly. For a 2-minute, 832×480 video at 16 FPS (h=30, w=52, B_f=4), the context size at the final block swells to ∼749K tokens—consuming an intractable amount of GPU memory. This fundamental scaling bottleneck directly motivates our three-partition design.

3.2 Three-Partition KV Cache

The core idea of PackForcing is to decouple the monotonically growing generation history into three distinct *functional* partitions. Rather than applying a one-size-fits-all eviction or compression strategy, we apply a tailored policy to each partition based on its temporal role and information density (Fig. 1).

Sink Tokens (Full resolution, never evicted). Inspired by the attention-sink phenomenon in StreamingLLM [35], we hypothesize that the earliest generated frames serve as critical semantic anchors. Let N_{sink} denote the number of these initial frames, corresponding to the first N_{sink}/B_f generation blocks. For a given transformer layer l, the sink cache is defined as:

$$\mathcal{C}_{\text{sink}}^l = \big\{(\mathbf{K}_j^l, \mathbf{V}_j^l)\big\}_{j=1}^{N_{\text{sink}}/B_f}\,, \quad |\mathcal{C}_{\text{sink}}^l| = \tfrac{N_{\text{sink}}}{B_f}\,n\,, \tag{4}$$

where j is the block index, and $\mathbf{K}_j^l, \mathbf{V}_j^l$ are the original, uncompressed key and value. These tokens lock in the scene layout, subject identity, and global style.

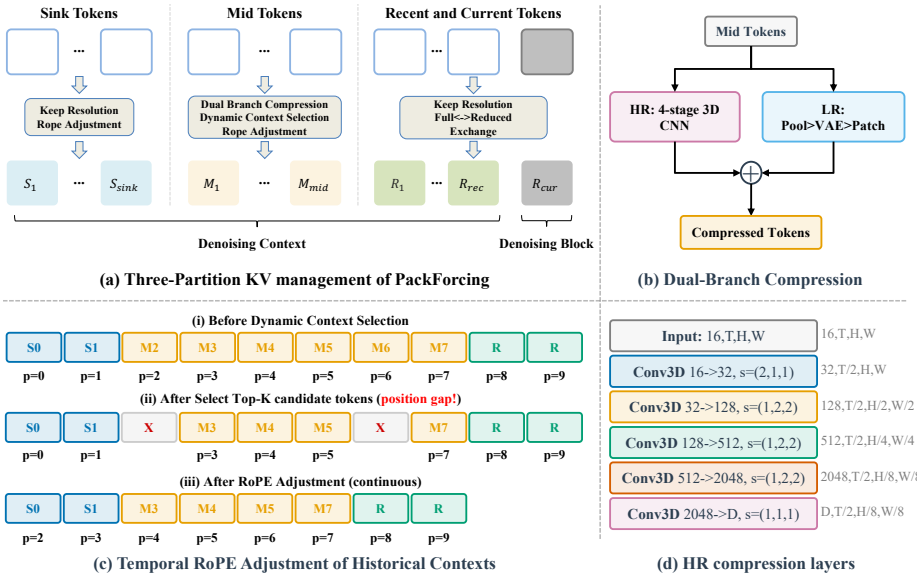

Fig. 1: Overview of PackForcing. (a) The three-partition KV management organizes the denoising context into sink tokens (full resolution), mid tokens (compressed and dynamically selected), and recent/current tokens (full resolution with full-to-reduced exchange). **(b) & (d)** The dual-branch compression module fuses a High-Resolution (HR) branch (a progressive 4-stage 3D CNN) with a Low-Resolution (LR) branch (pooling followed by VAE re-encoding) via element-wise addition to minimize token count. **(c)** To efficiently bound the memory footprint, dynamic context selection is applied to retain only the top-K informative mid tokens. To resolve the ensuing position gaps caused by dropped tokens, a Temporal RoPE Adjustment is utilized to re-align and enforce continuous positional indices across the historical contexts.

Because they are vital for preventing semantic drift, they are *never compressed or evicted*. We set N_{sink}=8 (two blocks), which consumes <2% of the total token budget for a 2-minute video, yet provides a robust and stable global reference throughout the entire generation process.

Compressed Mid Tokens (∼32× token reduction & dynamically routed). The vast majority of the video history falls between the initial sink frames and the most recent window. We define this region as the *mid* partition. Retaining this partition at full resolution is computationally prohibitive and highly redundant. Instead, tokens populating this region are represented by highly compressed KV pairs $(\tilde{\mathbf{K}}_j^l, \tilde{\mathbf{V}}_j^l)$ produced via our dual-branch module (Sec. 3.3). Furthermore, as this compressed buffer accumulates over time, we do not attend to the entire pool indiscriminately. We employ *Dynamic Context Selection* (Sec. 3.5) to dynamically evaluate query-key affinities, actively routing only the N_{mid} most informative blocks to form the active set $\mathcal{S}_{\text{mid}}$ for the current computation:

$$\mathcal{C}_{\text{mid}}^l = \left\{(\tilde{\mathbf{K}}_j^l, \tilde{\mathbf{V}}_j^l)\right\}_{j\in\mathcal{S}_{\text{mid}}}, \quad |\mathcal{C}_{\text{mid}}^l| \le N_{\text{mid}} \cdot N_c\,, \tag{5}$$

where the tilde ($\sim$) denotes compressed part and N_{mid} limits the active computational budget. N_c is the token count per compressed block, calculated as $N_c = \lfloor B_f/2 \rfloor \times \lfloor h/4 \rfloor \times \lfloor w/4 \rfloor$. Here, the factors 1/2 and 1/4 correspond to the downsampling strides of the compression module along the temporal and spatial dimensions. With default settings (B_f=4, h=30, w=52), each block is compressed to $N_c = 182$ tokens—a dramatic $\sim$32$\times$ reduction from the original n=6,240 tokens.

Recent & Current Tokens (Dual-resolution shifting). To maintain high-fidelity local temporal dynamics when generating new video frames, the most recently generated frames must be kept pristine. Let i denote the index of the block currently being generated and N_{recent} be the number of preceding recent frames. The context for this partition comprises the intact KV pairs from these recent blocks, alongside the current block i itself:

$$\mathcal{C}^l_{\text{rc}} = \left\{(\mathbf{K}^l_j, \mathbf{V}^l_j)\right\}^{i}_{j=i-N_{\text{recent}}/B_f}, \quad |\mathcal{C}^l_{\text{rc}}| = \left(\tfrac{N_{\text{recent}}}{B_f} + 1\right) n\,. \tag{6}$$

Preserving these recent tokens at uncompressed resolution guarantees smooth temporal transitions. Crucially, to bridge this partition with the mid-buffer without incurring sequential latency, we concurrently compute a low-resolution backup for these tokens. As detailed in Sec. 3.4, this dual-resolution shifting pipeline perfectly hides the compression overhead and ensures a seamless transition of aging recent tokens into long-term mid memory.

Bounded Attention Context. During the generation of block i, the transformer layer l concatenates the three partitions to form the active attention context:

$$\mathcal{C}^l = \mathcal{C}^l_{\text{sink}} \parallel \mathcal{C}^l_{\text{mid}} \parallel \mathcal{C}^l_{\text{rc}}\,, \tag{7}$$

which enforces a *constant* token count for the attention computation: $|\mathcal{C}^l| = \frac{N_{\text{sink}}}{B_f}\, n + N_{\text{mid}} \cdot N_c + \left(\frac{N_{\text{recent}}}{B_f} + 1\right) n$. Crucially, while the entire generation history is persistently maintained within the memory buffers (either at full resolution or in a highly compressed state), the actual context input for generating the block i is strictly bounded and independent of the total video length T. Rather than attending to the full growing sequence, this fixed-size input context is dynamically retrieved from the comprehensive historical partitions, ensuring $O(1)$ attention complexity without discarding any past memory.

3.3 Dual-Branch HR Compression

The mid partition requires a massive token reduction ($\sim$32$\times$) while retaining sufficient structural and semantic information for coherent attention patterns (see Fig. 2). A single-pathway compressor faces a steep trade-off: aggressive spatial downsampling preserves layout but loses texture, whereas semantic pooling preserves meaning but destroys spatial structure. To resolve this, we propose a *dual-branch* compression module (Fig. 1(b)) that aggregates fine-grained structure (HR branch) and coarse semantics (LR branch).

HR Branch: Progressive 3D Convolution. The HR branch operates directly on the VAE latent $\mathbf{z} \in \mathbb{R}^{B\times C\times T\times H\times W}$ to preserve local, fine-grained details. It

applies a cascade of strided 3D convolutions with SiLU activations. Specifically, it first performs a 2× temporal compression, followed by three stages of 2× spatial compression, and a final 1×1×1 projection to the model's hidden dimension d=1536. This yields a structurally rich representation $\mathbf{h}_{\text{HR}}$ with a total 128× volume reduction (2×8×8) in the latent space.

LR Branch: Pixel-Space Re-encoding. To capture complementary global context, the LR branch operates via a distinct pixel-space pathway. We decode the latent $\mathbf{z}$ back into pixel frames, apply a 3D average pooling (2× temporally, 4× spatially), and then re-encode the pooled frames back into the latent space using the frozen VAE encoder, followed by standard patch embedding to obtain $\mathbf{h}_{\text{LR}}$. This decoding-pooling-encoding pipeline preserves the perceptual layout far better than direct pooling in the latent space.

Feature Fusion. The outputs from both branches share the same dimensional space and are fused via element-wise addition:

$$\tilde{\mathbf{h}} = \mathbf{h}_{\text{HR}} + \mathbf{h}_{\text{LR}} \in \mathbb{R}^{B \times N_c \times d}, \tag{8}$$

where the compressed token count is $N_c = \lfloor T/2 \rfloor \times \lfloor H/8 \rfloor \times \lfloor W/8 \rfloor$. Given that the original patch embedding already performs a 1×2×2 spatial reduction, our dual-branch module effectively achieves a net token reduction of ∼32× per block (e.g., from 6,240 to 182 tokens). This simple yet effective fusion ensures comprehensive information retention under extreme compression.

3.4 Dual-Resolution Shifting and Incremental RoPE adjustment

Dual-Resolution Shifting Mechanism. Unlike FIFO methods that permanently discard tokens, we preserve long-term memory via a seamless dual-resolution pipeline. During chunk generation, we concurrently compute a full-resolution KV cache for immediate prediction and a reduced-resolution backup. Once the next chunk is generated, new full-resolution tokens replace the old ones, while the pre-computed compressed tokens smoothly slide into the *mid* partition. This retains comprehensive history while hiding compression latency.

The Position Misalignment Problem. Although this dual-track pipeline efficiently populates the compressed mid-buffer, strictly bounding the total memory capacity (N_{mid}) eventually necessitates evicting the absolute oldest compressed blocks from the mid partition. Our backbone uses 3D Rotary Position Embeddings (RoPE) [31] with separate temporal, height, and width frequencies $\boldsymbol{\theta} = [\boldsymbol{\theta}_t, \boldsymbol{\theta}_h, \boldsymbol{\theta}_w]$, allocated as $44 + 42 + 42 = 128$ dimensions per head. When a key is cached, it already carries the rotation for its absolute position p:

$$\mathbf{k}^{(p)}_{\text{cached}} = \mathbf{k}_{\text{raw}} \odot e^{i\,\boldsymbol{\theta}_p}. \tag{9}$$

After evicting Δ blocks ($\delta = \Delta B_f$ frames) to maintain the capacity budget, the *sink* keys still encode positions $0, \ldots, N_{\text{sink}}-1$, yet the earliest surviving *mid* key now encodes position $N_{\text{sink}} + \delta$. The resulting position gap breaks the positional continuity that the transformer attention relies on.

Incremental RoPE adjustment. We exploit two properties to resolve this without a full recomputation: (i) RoPE adjustments are *multiplicative*: $e^{i\boldsymbol{\theta}_p} \cdot$

$e^{i\boldsymbol{\theta}_\delta} = e^{i\boldsymbol{\theta}_{p+\delta}}$; and (ii) eviction shifts only the *temporal* axis. We therefore apply a highly efficient, temporal-only RoPE adjustment to the sink keys:

$$\mathbf{k}'_{\text{sink}} = \mathbf{k}_{\text{sink}} \odot e^{i\,\boldsymbol{\theta}_t(\delta)},\ \mathbf{1}_h,\ \mathbf{1}_w\ , \tag{10}$$

where $\mathbf{1}_h, \mathbf{1}_w$ are identity (unit-magnitude) rotations that leave spatial positions unchanged. After adjustment, the sink positions become $\delta, \delta{+}1, \ldots$, seamlessly preceding the mid positions. This operation is applied once per eviction event across all layers, costing $<0.1\%$ of total inference time.

3.5 Dynamic Context Selection

Treating all compressed blocks equally ignores their varying semantic importance. To prioritize visually critical keyframes, we introduce a dynamic context selection mechanism based on query-key affinity. Unlike DeepForcing [39], which permanently prunes unselected tokens, we employ a non-destructive soft-selection. We retrieve only the top-K most relevant mid-blocks for attention, keeping unselected tokens safely archived for potential future reactivation as the scene evolves. To ensure negligible overhead ($<1\%$), affinity scoring occurs only at the first denoising step of each block, caching the indices for subsequent steps. We further accelerate this by subsampling query tokens and using half the attention heads. This soft-routing improves subject consistency by $+0.8$ and the CLIP score by $+0.12$ over strict FIFO eviction (Table 5).

3.6 Empirical Analysis of Temporal Attention Patterns

To motivate our KV cache design, we empirically investigate the attention distribution of the denoising network during a 480-frame generation (Fig. 2). Our analysis reveals two critical insights (detailed in the Appendix): (1) Attention demand persists across the entire video history, invalidating naive FIFO eviction strategies; and (2) Highly attended tokens are sparsely and dynamically distributed, exhibiting a high Jaccard distance (0.75) between consecutive selection steps. These observations fundamentally justify our three-partition cache architecture (sink, mid, and recent). By aggressively compressing the sporadically queried yet globally essential mid-range tokens, we successfully preserve comprehensive spatiotemporal context within a strictly bounded memory footprint.

4 · Experiments

4.1 Experimental Settings

Implementation Details. PackForcing employs the Wan2.1-T2V-1.3B [33] backbone to generate 832×480 videos at 16 FPS. Text conditioning relies on the UMT5-XXL encoder, where text key-value pairs are computed once and cached globally to reduce overhead. Consistent with recent causal generative frameworks [16, 20, 39, 40], the causal student is initialized via ODE trajectory alignment. Training prompts are sourced from VidProM and augmented via large

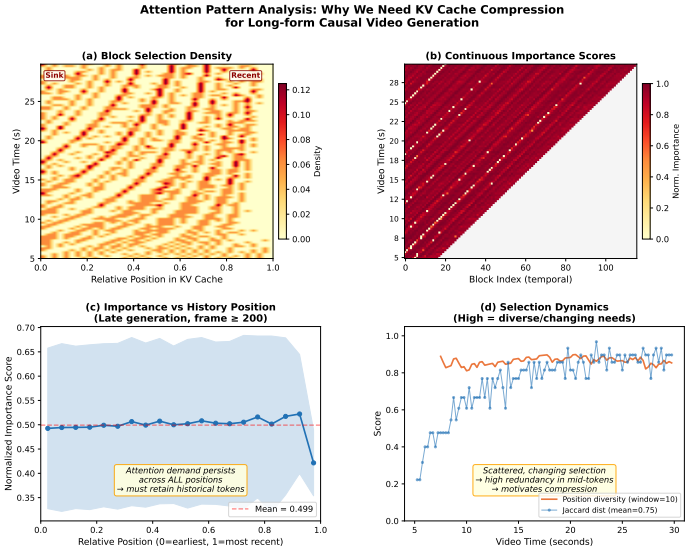

Fig. 2: Attention patterns in causal video generation. (a) Density map shows attention distributed across the entire history. **(b)** Importance scores $\phi_j = \sum_i \mathbf{q}_i^\top \mathbf{k}_j / \sqrt{d}$ reveal sparse information. **(c)** Near-flat late-stage importance (mean = 0.499) rules out FIFO eviction. **(d)** High Jaccard distance (0.75) and position diversity (> 0.85) show rapid, diverse block retrieval, motivating compression.

language models following the Self-Forcing paradigm. For the cache partitions, we set N_{sink}=8, N_{recent}=4, and N_{top}=16. Chunk-wise generation operates with B_f=4 latent frames per block and S=4 distilled denoising steps. The model is trained for 3,000 iterations with a batch size of 8 on a 20-latent-frame temporal window ($\tilde{5}$ seconds). We use AdamW [22] (β_1=0, β_2=0.999), setting learning rates to 2.0×10^{-6} for the generator G_θ and 1.0×10^{-6} for the fake score estimator s_{fake}, with a 1:5 update ratio. During inference, we utilize a classifier-free guidance scale of 3.0 and a timestep shift of 5.0.

Evaluation Protocols. To rigorously assess long video generation performance, we adopt the evaluation setting of VBench-Long [18]. Specifically, we utilize 128 text prompts sourced from MovieGen [26], adhering strictly to the prompt selection protocol established in Self-Forcing++ [7]. Consistent with the standard Self-Forcing [7] paradigm, each prompt is refined and expanded using Qwen2.5-7B-Instruct [28] prior to generation. Under this standardized setup, we generate videos at both 60 s and 120 s durations. We quantitatively evaluate the results using seven core metrics from the VBench framework [17]: Dynamic Degree (Dyn. Deg.), Motion Smoothness (Mot. Smth.), Overall Consistency (Over. Cons.), Imaging Quality (Img. Qual.), Aesthetic Quality (Aest. Qual.), Subject Consistency (Subj. Cons.), and Background Consistency (Back. Cons.). Furthermore, to measure the temporal stability of text-video alignment over ex-

tended durations, we compute and report CLIP scores at 10-second intervals throughout the generated sequences.

Table 1: Quantitative comparison on 60 s and 120 s benchmarks (7 VBench [17] metrics). Best results are highlighted in **bold**.

Method	Dyn. Deg.↑	Mot. Smth.↑	Over. Cons.↑	Img. Qual.↑	Aest. Qual.↑	Subj. Cons.↑	Back. Cons.↑
60 s generation							
CausVid [40]	48.43	98.04	23.36	65.69	60.63	84.53	89.84
LongLive [36]	44.53	98.70	25.73	69.06	**63.30**	92.00	92.97
Self-Forcing [16]	35.93	98.26	24.92	66.62	57.15	80.41	86.95
Rolling Forcing [20]	33.59	**98.70**	25.73	**71.06**	61.43	91.62	93.00
Deep Forcing [39]	53.67	98.56	21.75	67.75	58.88	**92.55**	**93.80**
PackForcing (ours)	**56.25**	98.29	**26.07**	69.36	62.56	90.49	93.46
120 s generation							
CausVid [40]	50.00	98.11	23.13	65.41	60.11	83.24	87.83
LongLive [36]	44.53	**98.72**	25.95	69.59	**63.00**	91.54	**93.73**
Self-Forcing [16]	30.46	98.12	23.42	62.49	51.68	74.40	83.57
Rolling Forcing [20]	35.15	98.65	25.45	**70.58**	60.62	90.14	92.40
Deep Forcing [39]	52.84	98.22	21.38	68.21	57.96	91.95	92.55
PackForcing (ours)	**54.12**	98.35	**26.05**	69.67	61.98	**92.84**	91.88

4.2 Main Results

Quantitative Comparison. Table 1 evaluates 60 s and 120 s video generation across seven VBench metrics. PackForcing excels in motion synthesis, achieving the highest Dynamic Degree at both durations (56.25 and 54.12) and outperforming the baseline, CausVid, by +7.82 and +4.12. This confirms that our persistent memory mechanism provides reliable temporal grounding, enabling the model to confidently synthesize extensive motion. Furthermore, PackForcing demonstrates superior stability over extended horizons. While baselines like Self-Forcing degrade significantly from 60 s to 120 s (e.g., Subject Consistency drops by 6.01, Aesthetic Quality by 5.47), our performance declines are marginal (-1.01 and -0.49, respectively). This robustness highlights the effectiveness of sink tokens in anchoring global semantics and the dynamically routed mid-buffer in preserving intermediate context. Finally, despite an aggressive $\sim$32$\times$ token reduction, PackForcing maintains highly competitive Image and Aesthetic Quality, proving that our dual-branch architecture successfully retains perceptually critical spatiotemporal details.

Long-Range Consistency. Table 2 evaluates the temporal stability of text-video alignment via CLIP scores at 10-second intervals. PackForcing maintains the highest alignment throughout the 60-second generation, with a marginal decline of only 1.14 points (from 34.04 to 32.90). Conversely, baselines suffer from compounding errors: Self-Forcing exhibits a severe 6.77-point drop, while CausVid declines by 1.86 points. This sustained temporal coherence directly

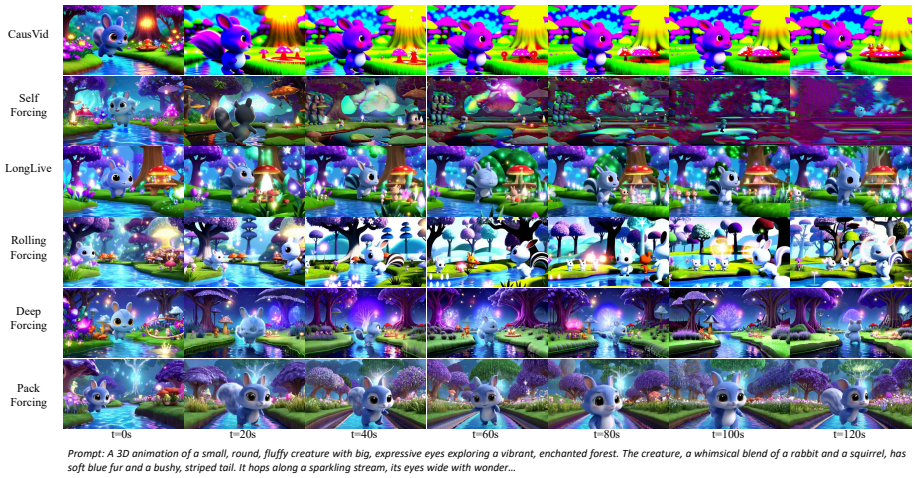

Fig. 3: Qualitative comparison of 120 s generation. Sampled frames across seven timestamps under the same prompt. PackForcing consistently maintains strict subject identity and high visual fidelity throughout the entire sequence. In contrast, baseline methods suffer from progressive degradation over time: Self-Forcing exhibits color shifts by 60 s and eventual collapse, CausVid loses background details, and LongLive generates severely restricted motion. Furthermore, Rolling-Forcing struggles with significant subject inconsistencies, while DeepForcing loses the main subject at 100s.

validates our sink token mechanism's ability to anchor global semantics across extended horizons.

Table 2: CLIP Score comparison on long video generation (60s).

Method	0–10 s	10–20 s	20–30 s	30–40 s	40–50 s	50–60 s	Overall
CausVid [40]	32.65	31.78	31.47	31.13	30.81	30.79	31.44
LongLive [36]	33.95	33.38	33.14	**33.51**	**33.45**	**33.36**	33.46
Self-Forcing [16]	33.89	33.23	31.66	29.99	28.37	27.12	30.71
Rolling Forcing [20]	33.85	33.39	32.94	32.78	32.49	32.25	32.95
Deep Forcing [39]	33.47	33.29	32.38	32.28	32.26	32.27	32.33
PackForcing (ours)	**34.04**	**33.99**	**33.70**	33.37	33.24	32.90	**33.54**

Qualitative Comparison. Fig. 3 presents sampled frames from a 120-second generation for all methods under the same text prompt. At 0 s all methods produce comparable quality. By 60 s, Self-Forcing exhibits visible color shift and object duplication; CausVid loses fine details in the background. At 120 s, only PackForcing and LongLive maintain coherent subjects, but LongLive shows noticeably less camera and subject motion. PackForcing preserves both subject identity and dynamic motion throughout the full two minutes, thanks to the persistent sink tokens and compressed history.

Fig. 4: Qualitative ablation results on 60 s generation. Removing sink tokens leads to progressive semantic drift, whereas disabling either the RoPE adjustment or dynamic context selection introduces severe frame reset artifacts.

4.3 Ablation Studies

We perform systematic ablations on the 60 s benchmark to evaluate each critical component of PackForcing. Qualitative comparisons are shown in Fig. 4.

Table 3: Quantitative ablation results of sink size. We consolidate overall CLIP scores and VBench metrics across varying sink sizes. Setting N_{sink}=8 achieves the optimal balance between dynamic motion and semantic consistency.

Sink Size	Overall CLIP ↑	Dyn. Deg. ↑	Mot. Smth. ↑	Over. Cons. ↑	Img. Qual. ↑	Aest. Qual. ↑	Subj. Cons. ↑	Back. Cons. ↑
0	31.24	20.31	**98.89**	23.11	72.87	59.37	74.72	86.37
2	34.85	49.69	98.57	26.78	72.59	65.85	91.37	93.68
4	34.85	**50.16**	98.57	**26.99**	72.59	65.84	91.37	93.73
8	35.09	49.84	98.68	26.29	**73.18**	**66.46**	93.11	94.53
16	**35.39**	35.16	98.82	26.71	73.05	66.34	**93.84**	**94.92**

Effect of Sink Tokens. To evaluate the impact of sink size (N_{sink}) on long-term stability, we conduct an ablation study on 60 randomly selected samples (Table 3). Removing attention sinks entirely (N_{sink}=0) causes severe semantic drift, evidenced by sharp drops in the Overall CLIP score (35.09 to 31.24) and Subject Consistency (93.11 to 74.72), confirming their role as critical global anchors (Fig. 4). Conversely, an excessively large sink (N_{sink}=16) maximizes consistency but stifles motion, plummeting the Dynamic Degree to 35.16 as the model over-conditions on static early frames. Setting N_{sink}=8 achieves the optimal balance. It preserves dynamic richness (49.84) and yields the best Image (73.18) and Aesthetic (66.46) Quality, with Subject Consistency within 1% of the N_{sink}=16 setting—all while consuming strictly bounded memory (<2% of the total token budget). Temporal CLIP details can be found in the Appendix.
Effect of Compression Branches. Table 4 isolates each compression branch, evaluated at 60 s where all variants fit in memory. The HR branch alone provides

Table 4: Ablation on compress. branches.

Branch	Img. Qual. ↑	Over. Cons. ↑	CLIP ↑	60 s
HR only	68.12	25.41	32.97	✓
LR only	67.45	25.18	33.11	✓
HR + LR	**69.36**	**26.07**	**33.54**	✓

Table 5: Ablation on eviction strategy.

Strategy	Subj. Cons. ↑	Over. Cons. ↑	CLIP ↑
Random	86.31	25.42	33.01
FIFO	87.82	25.91	33.42
Dynamic Select	**88.62**	**26.07**	**33.54**

strong spatial compression but misses coarse semantic cues; the LR branch alone preserves semantics but lacks spatial precision.

FIFO vs. Dynamic Context Selection. Table 5 compares memory management strategies within the compressed mid-buffer. Dynamic context selection outperforms standard FIFO, yielding notable improvements in Subject Consistency (+0.8) and the overall CLIP score (+0.12). This advantage stems from its affinity-driven approach, which prioritizes the retention of highly attended historical blocks rather than relying on a rigid chronological eviction.

4.4 Discussion

Generalization from Short-Video Supervision. The remarkable 24× temporal extrapolation capability of PackForcing (from 5 s training clips to 120 s generated videos) can be attributed to two primary mechanisms. First, the framework enforces context size invariance. By systematically compressing and managing the KV cache, the attention context remains strictly bounded (e.g., ∼27,872 tokens) during both training and inference. This effectively prevents out-of-distribution sequence length shifts that typically trigger error accumulation in standard autoregressive models. Second, the architecture ensures representational compatibility. Jointly training the dual-branch compression layer aligns the compressed tokens with the full-resolution tokens within the same latent subspace, enabling the transformer to seamlessly attend to the highly compressed historical features.

Motion Richness and Dynamic Degree. As indicated by the VBench evaluations, PackForcing consistently achieves superior dynamic degree scores. Existing autoregressive baselines, which often lack persistent long-range memory, tend to degenerate into producing low-variance, near-static frames as a conservative strategy to avoid compounding temporal errors. In contrast, by preserving the global layout through sink tokens and the intermediate motion trajectory through the compressed mid-buffer, our framework provides a reliable, uninterrupted spatiotemporal reference. This comprehensive contextual grounding encourages the model to confidently synthesize complex, continuous motion without collapsing into static artifacts.

Limitations and Future Work. While PackForcing excels in generating dynamic content, we observe a nuanced trade-off. Baselines such as LongLive achieve marginally higher Subject Consistency (92.00 versus our 90.49) at the severe expense of motion richness, yielding a Dynamic Degree of only 44.53 compared to our 56.25. Closing this consistency gap by further enhancing strict subject preservation without compromising the high dynamic diversity enabled by our compressed history remains a primary direction for future work.

5 Conclusion

In this paper, we introduce **PackForcing**, a unified framework that fundamentally resolves the dual bottlenecks of error accumulation and unbounded memory growth in autoregressive video generation. By strategically partitioning the KV cache into **sink**, **compressed mid**, and **recent** tokens, our approach strictly bounds the memory footprint and ensures constant-time attention complexity without discarding essential historical context. Empowered by a 128× dual-branch compression module (∼32× token reduction), incremental RoPE adjustments, and dynamic context selection, PackForcing achieves a remarkable 24× temporal extrapolation (5 s → 120 s). This demonstrates that short-video supervision is entirely sufficient for high-quality, long-video synthesis. Ultimately, it generates highly coherent 2-minute videos, establishing state-of-the-art VBench scores and the most robust text-video alignment among existing baselines, paving the way for efficient, unbounded video generation on standard hardware.

References

1. Blattmann, A., Rombach, R., Ling, H., Dockhorn, T., Kim, S.W., Fidler, S., Kreis, K.: Align your latents: High-resolution video synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 22563–22575 (2023) 2, 3
2. Brooks, T., Peebles, B., Holmes, C., DePue, W., Guo, Y., Jing, L., Schnurr, D., Taylor, J., Luhman, T., Luhman, E., Ng, C., Wang, R., Ramesh, A.: Video generation models as world simulators (2024), `https://openai.com/research/video-generation-models-as-world-simulators` 3
3. Ceylan, D., Huang, C.H.P., Mitra, N.J.: Pix2video: Video editing using image diffusion. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 23206–23217 (2023) 2
4. Chen, B., Martí Monsó, D., Du, Y., Simchowitz, M., Tedrake, R., Sitzmann, V.: Diffusion forcing: Next-token prediction meets full-sequence diffusion. Advances in Neural Information Processing Systems **37**, 24081–24125 (2024) 2
5. Chen, H., Zhang, Y., Cun, X., Xia, M., Wang, X., Weng, C., Shan, Y.: Videocrafter2: Overcoming data limitations for high-quality video diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 7310–7320 (2024) 2
6. Chen, J., Zhao, Y., Yu, J., Chu, R., Chen, J., Yang, S., Wang, X., Pan, Y., Zhou, D., Ling, H., et al.: Sana-video: Efficient video generation with block linear diffusion transformer. arXiv preprint arXiv:2509.24695 (2025) 2
7. Cui, J., Wu, J., Li, M., Yang, T., Li, X., Wang, R., Bai, A., Ban, Y., Hsieh, C.J.: Self-forcing++: Towards minute-scale high-quality video generation. arXiv preprint arXiv:2510.02283 (2025) 10
8. Ge, S., Nah, S., Liu, G., Poon, T., Tao, A., Catanzaro, B., Jacobs, D., Huang, J.B., Liu, M.Y., Balaji, Y.: Preserve your own correlation: A noise prior for video diffusion models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 22930–22941 (2023) 4
9. Harvey, W., Naderiparizi, S., Masrani, V., Weilbach, C., Wood, F.: Flexible diffusion modeling of long videos. Advances in neural information processing systems **35**, 27953–27965 (2022) 2

10. He, H., Xu, Y., Guo, Y., Wetzstein, G., Dai, B., Li, H., Yang, C.: CameraCtrl: Enabling camera control for text-to-video generation (2024) 2
11. He, H., Yang, C., Lin, S., Xu, Y., Wei, M., Gui, L., Zhao, Q., Wetzstein, G., Jiang, L., Li, H.: Cameractrl ii: Dynamic scene exploration via camera-controlled video diffusion models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 13416–13426 (2025) 2
12. Henschel, R., Khachatryan, L., Poghosyan, H., Hayrapetyan, D., Tadevosyan, V., Wang, Z., Navasardyan, S., Shi, H.: Streamingt2v: Consistent, dynamic, and extendable long video generation from text. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 2568–2577 (2025) 4
13. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Advances in neural information processing systems **33**, 6840–6851 (2020) 3
14. Ho, J., Salimans, T., Gritsenko, A., Chan, W., Norouzi, M., Fleet, D.J.: Video diffusion models. Advances in neural information processing systems **35**, 8633–8646 (2022) 2, 3
15. Hong, S., Seo, J., Shin, H., Hong, S., Kim, S.: Large language models are frame-level directors for zero-shot text-to-video generation. In: First Workshop on Controllable Video Generation@ ICML24 (2023) 4
16. Huang, X., Li, Z., He, G., Zhou, M., Shechtman, E.: Self forcing: Bridging the train-test gap in autoregressive video diffusion. arXiv preprint arXiv:2506.08009 (2025) 2, 4, 9, 11, 12, 18
17. Huang, Z., He, Y., Yu, J., Zhang, F., Si, C., Jiang, Y., Zhang, Y., Wu, T., Jin, Q., Chanpaisit, N., 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) 10, 11
18. Huang, Z., Zhang, F., Xu, X., He, Y., Yu, J., Dong, Z., Ma, Q., Chanpaisit, N., Si, C., Jiang, Y., et al.: Vbench++: Comprehensive and versatile benchmark suite for video generative models. IEEE Transactions on Pattern Analysis and Machine Intelligence (2025) 10
19. Lipman, Y., Chen, R.T., Ben-Hamu, H., Nickel, M., Le, M.: Flow matching for generative modeling. arXiv preprint arXiv:2210.02747 (2022) 3, 4
20. Liu, K., Hu, W., Xu, J., Shan, Y., Lu, S.: Rolling forcing: Autoregressive long video diffusion in real time. arXiv preprint arXiv:2509.25161 (2025) 2, 4, 9, 11, 12, 18
21. Liu, X., Gong, C., Liu, Q.: Flow straight and fast: Learning to generate and transfer data with rectified flow. arXiv preprint arXiv:2209.03003 (2022) 3
22. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017) 10
23. Lu, Y., Zeng, Y., Li, H., Ouyang, H., Wang, Q., Cheng, K.L., Zhu, J., Cao, H., Zhang, Z., Zhu, X., et al.: Reward forcing: Efficient streaming video generation with rewarded distribution matching distillation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 34385–34397 (2026) 4
24. Peebles, W., Xie, S.: Scalable diffusion models with transformers. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 4195–4205 (2023) 3
25. Peng, B., Quesnelle, J., Fan, H., Shippole, E.: Yarn: Efficient context window extension of large language models. arXiv preprint arXiv:2309.00071 (2023) 4
26. Polyak, A., Zohar, A., Brown, A., Tjandra, A., Sinha, A., Lee, A., Vyas, A., Shi, B., Ma, C.Y., Chuang, C.Y., et al.: Movie gen: A cast of media foundation models. arXiv preprint arXiv:2410.13720 (2024) 2, 3, 10

27. Qiu, H., Xia, M., Zhang, Y., He, Y., Wang, X., Shan, Y., Liu, Z.: Freenoise: Tuning-free longer video diffusion via noise rescheduling. arXiv preprint arXiv:2310.15169 (2023) 4
28. Qwen, :, Yang, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Li, C., Liu, D., Huang, F., Wei, H., Lin, H., Yang, J., Tu, J., Zhang, J., Yang, J., Yang, J., Zhou, J., Lin, J., Dang, K., Lu, K., Bao, K., Yang, K., Yu, L., Li, M., Xue, M., Zhang, P., Zhu, Q., Men, R., Lin, R., Li, T., Tang, T., Xia, T., Ren, X., Ren, X., Fan, Y., Su, Y., Zhang, Y., Wan, Y., Liu, Y., Cui, Z., Zhang, Z., Qiu, Z.: Qwen2.5 technical report (2025) 10
29. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10684–10695 (2022) 3
30. Singer, U., Polyak, A., Hayes, T., Yin, X., An, J., Zhang, S., Hu, Q., Yang, H., Ashual, O., Gafni, O., et al.: Make-a-video: Text-to-video generation without text-video data. arXiv preprint arXiv:2209.14792 (2022) 3
31. Su, J., Ahmed, M., Lu, Y., Pan, S., Bo, W., Liu, Y.: Roformer: Enhanced transformer with rotary position embedding. Neurocomputing **568**, 127063 (2024) 8
32. Valevski, D., Leviathan, Y., Arar, M., Fruchter, S.: Diffusion models are real-time game engines. arXiv preprint arXiv:2408.14837 (2024) 2
33. Wan, T., Wang, A., Ai, B., Wen, B., Mao, C., Xie, C.W., Chen, D., Yu, F., Zhao, H., Yang, J., et al.: Wan: Open and advanced large-scale video generative models. arXiv preprint arXiv:2503.20314 (2025) 2, 3, 9, 18
34. Wang, Z., Yuan, Z., Wang, X., Li, Y., Chen, T., Xia, M., Luo, P., Shan, Y.: Motionctrl: A unified and flexible motion controller for video generation. In: ACM SIGGRAPH 2024 Conference Papers. pp. 1–11 (2024) 2
35. Xiao, G., Tian, Y., Chen, B., Han, S., Lewis, M.: Efficient streaming language models with attention sinks. arXiv preprint arXiv:2309.17453 (2023) 4, 5
36. Yang, S., Huang, W., Chu, R., Xiao, Y., Zhao, Y., Wang, X., Li, M., Xie, E., Chen, Y., Lu, Y., et al.: Longlive: Real-time interactive long video generation. arXiv preprint arXiv:2509.22622 (2025) 4, 11, 12, 18
37. Yang, Z., Teng, J., Zheng, W., Ding, M., Huang, S., Xu, J., Yang, Y., Hong, W., Zhang, X., Feng, G., et al.: Cogvideox: Text-to-video diffusion models with an expert transformer. arXiv preprint arXiv:2408.06072 (2024) 3
38. Yesiltepe, H., Meral, T., Akan, A.K., Oktay, K., Yanardag, P.: Infinity-rope: Action-controllable infinite video generation emerges from autoregressive self-rollout. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 40256–40265 (2026) 4
39. Yi, J., Jang, W., Cho, P.H., Nam, J., Yoon, H., Kim, S.: Deep forcing: Training-free long video generation with deep sink and participative compression. arXiv preprint arXiv:2512.05081 (2025) 4, 9, 11, 12, 18
40. Yin, T., Zhang, Q., Zhang, R., Freeman, W.T., Durand, F., Shechtman, E., Huang, X.: From slow bidirectional to fast autoregressive video diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 22963–22974 (2025) 2, 4, 9, 11, 12, 18
41. Zhang, Z., Sheng, Y., Zhou, T., Chen, T., Zheng, L., Cai, R., Song, Z., Tian, Y., Ré, C., Barrett, C., et al.: H2o: Heavy-hitter oracle for efficient generative inference of large language models. Advances in Neural Information Processing Systems **36**, 34661–34710 (2023) 4
42. Zheng, Z., Peng, X., Yang, T., Shen, C., Li, S., Liu, H., Zhou, Y., Li, T., You, Y.: Open-sora: Democratizing efficient video production for all. arXiv preprint arXiv:2412.20404 (2024) 3