PAST- AND FUTURE-INFORMED KV CACHE POLICY WITH SALIENCE ESTIMATION IN AUTOREGRESSIVE VIDEO DIFFUSION

Anonymous authors
Paper under double-blind review

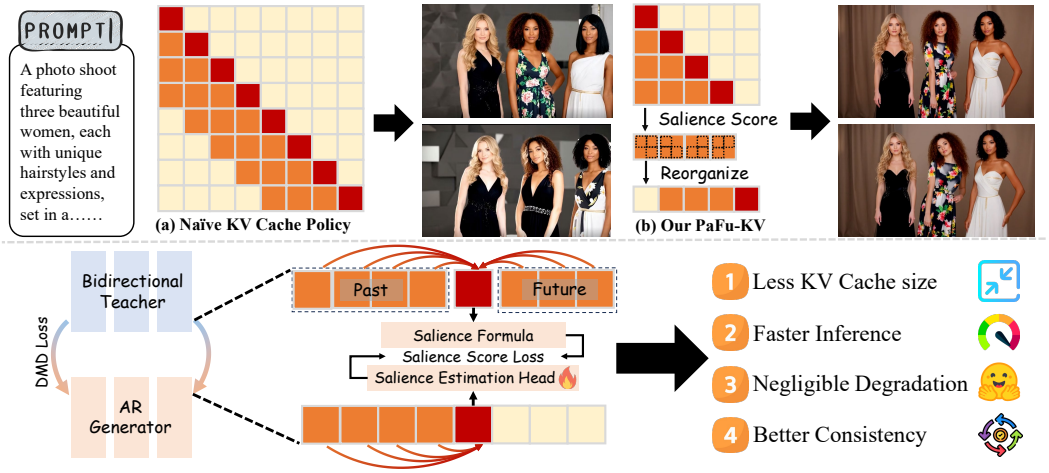


Figure 1: By distilling both past and future contextual information from a bidirectional teacher model, our PaFu-KV retaining KV Cache token with high salience score, achieving less KV Cache size and faster inference with negligible degradation on generation quality.

ABSTRACT

Video generation is pivotal to digital media creation, and recent advances in autoregressive video generation have markedly enhanced the efficiency of real-time video synthesis. However, existing approaches generally rely on heuristic KV Cache policies, which ignore differences in token importance in long-term video generation. This leads to the loss of critical spatiotemporal information and the accumulation of redundant, invalid cache, thereby degrading video generation quality and efficiency. To address this limitation, we first observe that token contributions to video generation are highly time-heterogeneous and accordingly propose a novel Past- and Future-Informed KV Cache Policy (PaFu-KV). Specifically, PaFu-KV introduces a lightweight Salience Estimation Head distilled from a bidirectional teacher to estimate salience scores, allowing the KV cache to retain informative tokens while discarding less relevant ones. This policy yields a better quality-efficiency trade-off by shrinking KV cache capacity and reducing memory footprint at inference time. Extensive experiments on benchmarks demonstrate that our method preserves high-fidelity video generation quality while enables accelerated inference, thereby enabling more efficient long-horizon video generation. Our code will be released upon paper acceptance.

1 INTRODUCTION

Video generation plays a crucial role in digital media creation (Xue et al., 2025; Ma et al., 2025), driven by recent Diffusion Transformer (DiT)-based models, such as Wan (Wan et al., 2025), Hunyuan (Kong et al., 2024), and CogVideo (Yang et al., 2024), which have demonstrated remarkable

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

progress in producing impressive videos (Yang et al., 2025b). However, their reliance on bidirectional attention incurs high memory overhead and long generation latency (Liang et al., 2024), which in turn degrades quality in long video generation and limits support for real-time interaction. To address this limitation, recent work (Huang et al., 2025a; Yang et al., 2025a) integrates the autoregressive (AR) paradigm with DiT (Peebles & Xie, 2023) for sequential video generation using KV caching, enabling real-time interactive video generation (Yang et al., 2025a; Shin et al., 2025).

Traditional AR (Ren et al., 2025; Tian et al., 2024; Yu et al., 2025a) video generation models adopt indiscriminate KV retention and eviction strategies, such as first-in-first-out (FIFO), ignoring the importance of different tokens in long-horizon video generation. These heuristic key-value (KV) caching strategies lead to the loss of critical contextual information, exacerbated error accumulation, and ultimately a dual degradation in generation quality and efficiency. Prior works (Xiang et al., 2025; Yi et al., 2025) have explored complementary strategies to mitigate such limitations, including sparse KV Cache retrieval at the semantic or prompt level, yet these methods introduce extra computational overhead during inference. More importantly, such strategies suffer from an inherent myopia: they assess token importance solely based on the KV Cache of existing local neighbors, without accounting for a token's potential influence on future generations and long-horizon historical KV Cache states, causing the performance gap that is to be bridged in this paper.

Our work addresses an important challenge in AR video diffusion: causal frame generation relies exclusively on past tokens, overlooking future frame correlations. This asymmetric temporal dependency limits the model's ability to reason over long horizons and often degrades long-range identity consistency, particularly in scenarios involving object occlusion or reappearance. To tackle this, we propose a token salience scoring criterion grounded in both past and future relevance, enabling reliable KV Cache management. Specifically, high-salience tokens are strongly backed by past or future context and exhibit high self-consistency, making them robust for long-term cache retention. In contrast, low-salience tokens lack stable global anchoring, are prone to representational drift, and pose a high-risk source of error accumulation. Building on this criterion, we leverage a large-scale DiT-based teacher model to capture long-range dependencies that causal AR models cannot access, thanks to its inherent bidirectional self-attention mechanism that simultaneously models past and future token relationships. We further introduce a spatialtemporal-balanced salience scoring strategy that explicitly disentangles attention contributions from past, present, and future interactions, thus enabling more accurate estimation of the global utility of tokens in long video generation.

Based on the aforementioned salience scoring strategy, we propose Past- and Future-informed KV Cache (PaFu-KV), a novel caching policy for AR video generation that significantly reduces KV Cache footprint while preserving generation fidelity. Specifically, as causal AR video diffusion models inherently lack access to future information during inference, we integrate a lightweight learnable Salience Estimation Head (SEH) into the DiT backbone to predict token-level salience scores, where the training is seamlessly incorporated into the teacher model with the data-free Distribution Matching Distillation paradigm (Yin et al., 2024b;a), thereby uncovering the historical tokens actually required for future generation under optimal information flow. During inference, PaFu-KV leverages SEH-predicted token salience scores to retain informative tokens in the KV Cache and evict those with marginal generative contributions. By selectively preserving salient tokens and discarding low-salience ones, the model learns to retain key information for fidelity preservation while discarding redundant tokens to mitigate error amplification, yielding more stable and coherent long-horizon video generation. Notably, based on the key observation that salient tokens exhibit heavy overlap across layers, we can efficiently capture global salience information by introducing a lightweight salience score prediction only at a single specific layer, while preserving the model's inference efficiency. Extensive experiments on public benchmark datasets validate the efficacy of our method, and our key contributions are summarized as follows:

- We propose PaFu-KV, a novel past- and future-informed KV Cache policy that explicitly targets KV Cache size reduction for efficient AR video generation.
- We design a lightweight Salience Estimation Head (SEH) and train it via Distribution Matching Distillation in a data-free paradigm, distilling omnidirectional knowledge from a bidirectional teacher to accurately estimate token importance under causal generation constraints with minimal consumption.
- Extensive experiments demonstrate that PaFu-KV significantly reduces KV Cache size and accelerates inference while preserving high-quality video generation.

2 Related Work

Autoregressive Video Diffusion. To enable efficient long video generation (Zhang & Agrawala, 2025; Jiang et al., 2024; Li et al., 2025), recent studies have increasingly focused on AR video diffusion. In the early stage, Ca2-VDM (Gao et al., 2024) adopts a teacher-forcing scheme, while Chen et al. (2024) introduced Diffusion Forcing. Building on this paradigm, CausVid (Yin et al., 2025) leverages DMD (Yin et al., 2024b) to distill a few-step AR video generator from a bidirectional DiT-based teacher model with Diffusion Forcing (Chen et al., 2024) framework. However, these approaches suffer from training-inference discrepancy, which leads to error accumulation during inference. Self Forcing (Huang et al., 2025a) mitigates this issue by conditioning each frame on generated context, but still exhibits semantic drift when generating long videos. Drawing inspiration from StreamingLLM (Xiao et al., 2023) in the Large Language Model (LLM) domain, LongLive (Yang et al., 2025a) introduces a frame sink mechanism and extends training to longer video sequences. Additionally, Reward Forcing proposed by Lu et al. (2025) augmented DMD with a reward function and applies frame sink with an Exponential Moving Average. Yu et al. (2025b) incorporated a state-space model (Gu & Dao, 2024) into AR video diffusion to jointly process global and local memory. Although effective at mitigating error accumulation, these methods largely overlook KV Cache optimization, which is essential for improving efficiency in AR video diffusion.

KV Cache in Large Language Models. Recent works in the LLM domain have explored improving the efficiency of AR generation by reducing the memory footprint and computational cost of the KV Cache. Early approaches employ attention-driven token selection to identify important tokens, such as H2O (Zhang et al., 2023) and SnapKV (Li et al., 2024). Beyond token-level, hierarchical compression strategies have been proposed to better preserve long-range context. PyramidalKV (Cai et al., 2024) progressively downsamples older KV entries into coarser representations, while CAKE (Qin et al., 2025) introduces adaptive cascading eviction policies. In contrast, AdaKV (Feng et al., 2024) explores head-wise KV Cache eviction by establishing a theoretical loss bound and allocating adaptive budgets across attention heads. In parallel, RazorAttention (Tang et al., 2024) leverages retrieval heads to guide KV Cache compression by preserving KV pairs likely to be accessed during decoding. Collectively, these methods aim to improve inference efficiency by approximating full attention through principled KV Cache compression. While these methods are effective in LLM, the multi-step denoising nature of diffusion prevents their direct adoption. Thus, we focus on designing a KV Cache policy specifically tailored for AR video diffusion in this work.

3 Method

3.1 Preliminary

Autoregressive Video Diffusion. In AR video diffusion models, the joint distribution over an N-frame video $x_{1:N}$ is factorized as

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

A key challenge in AR video diffusion is the training-inference discrepancy. Self-Forcing mitigates this issue by introducing a self-rollout mechanism, which aligns training with inference via a diffusion-based conditional formulation. Specifically, each frame is generated by denoising Gaussian noise with a few-step diffusion model G_θ, conditioned on previously generated frames via sliding-window attention with a fixed window size L over timesteps $\{t_0, t_1, \ldots, t_T\}$. A KV Cache stores representations of recent frames and is maintained with a fixed capacity using a FIFO eviction policy. At each timestep t_j, the denoised i-th frame is re-perturbed via a forward diffusion process Ψ, yielding the input $x^i_{t_{j-1}}$ for the subsequent denoising step. This process can be formulated as:

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

where KV is the KV Cache from previous frames.

Distribution Matching Distillation. Distribution Matching Distillation (DMD) formulates few-step diffusion distillation as a distribution matching problem based on the reverse Kullback–Leibler

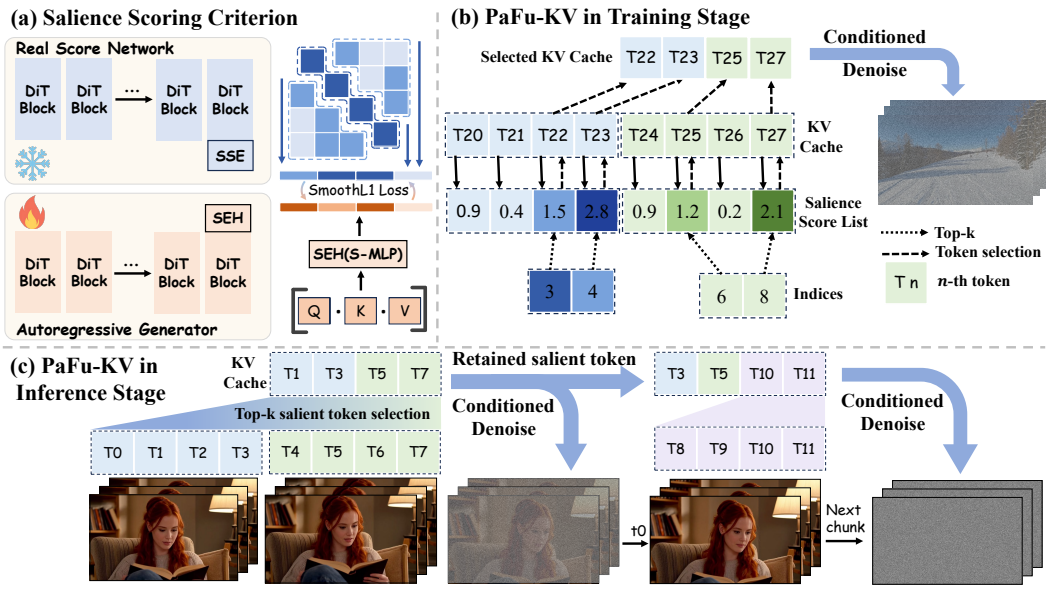


Figure 2: Overall framework of PaFu-KV. (a) Overview of the salience score criterion during training. (b) Detailed training procedure of PaFu-KV. (c) Inference pipeline of PaFu-KV, where the KV Cache is maintained at a compact size via salience-based eviction.

(KL) divergence. It employs a few-step generator G_θ, together with a frozen real score network and a continuously trained fake score network. Rather than explicitly computing the KL divergence, DMD optimizes the generator by directly using the gradient of the reverse KL objective with respect to θ. By defining $x = G_\theta(z)$ with $z \sim \mathcal{N}(0, \mathbf{I})$, the gradient of the reverse KL divergence with respect to θ can be written as:

$$\nabla_\theta D_{\mathrm{KL}} = \mathbb{E}_{z\sim\mathcal{N}(0,\mathbf{I})}\left[-\Big(s_{\mathrm{real}}(x) - s_{\mathrm{fake}}(x)\Big)\frac{\partial G_\theta(z)}{\partial \theta}\right], \tag{3}$$

where $s_*(x) = \nabla_x \log p_*(x)$ denotes the score function. This formulation reveals that the optimization of G_θ is governed by the discrepancy between the real and fake score, thereby providing a principled direction toward the target data distribution.

3.2 Spatialtemporal-balanced Salience Estimation

As illustrated in Fig. 2(a), our salience scoring criterion operates through two complementary components: (i) a Spatialtemporal-balanced Salience Estimation (SSE), detailed in this section, and (ii) the Salience Estimation Head (SEH), described in Sec. 3.3. Prior studies (Zhang et al., 2023; Singhania et al., 2024) demonstrate that attention weights effectively capture token importance in self-attention. Given input of $Q, K \in \mathbb{R}^{B\times N\times L\times D}$, where B, N, L, and D denote the batch size, number of attention heads, sequence length and head dimension, respectively, attention weights are computed by $P = \mathrm{softmax}(QK^\top/\sqrt{D})$, where $P \in \mathbb{R}^{B\times N\times L\times L}$. We estimate key importance by taking the maximum attention weights over the query dimension, capturing the strongest interaction between each key and any query, where higher scores indicate stronger relevance. Consistent with prior works, removing tokens with low scores incurs negligible impact on generation quality. Furthermore, we observe a pronounced diagonal bias in P. Specifically, we collect key indices corresponding to maximal query responses and construct a 7×7 histogram by uniformly partitioning the index and token ranges, as shown in Fig. 3(a). Our empirical analysis indicates that self-attention predominantly focuses on intra-frame or temporally local information. As a result, naïve global max-based aggregation over-selects local tokens while under-representing cross-frame interactions that are essential for temporal coherence in video generation.

Consequently, our SSE is explicitly designed to mitigate this bias and better capture long-range temporal dependencies. We set the length of each block as L_B, and further decompose P into three

components based on each block: the upper triangular, the diagonal, and the lower triangular. For a query and key at position i and j respectively, the lower and upper boundaries for each block are defined as $\ell(i) = b(i) \cdot L_B$ and $h(i) = \ell(i) + L_B$, where $b(i) = \lfloor \frac{i}{L_B} \rfloor$ and $\lfloor \cdot \rfloor$ denotes the floor operation. Accordingly, we define the index sets corresponding to each component as $\mathcal{I}_{\text{low}}(j) = \{\, i \mid j < \ell(i) \,\}$, $\mathcal{I}_{\text{diag}}(j) = \{\, i \mid \ell(i) \le j < h(i) \,\}$, and $\mathcal{I}_{\text{up}}(j) = \{\, i \mid j \ge h(i) \,\}$, respectively. We then take the maximum over the query dimension for each component and average across attention heads. The value for each components are computed as follows:

$$\begin{aligned} \text{low}_{b,j} &= \frac{1}{N} \sum_{n} \left(\max_{i \in \mathcal{I}_{\text{low}}(j)} P_{b,h,i,j} \right), \\ \text{diag}_{b,j} &= \frac{1}{N} \sum_{n} \left(\max_{i \in \mathcal{I}_{\text{diag}}(j)} P_{b,h,i,j} \right), \\ \text{up}_{b,j} &= \frac{1}{N} \sum_{n} \left(\max_{i \in \mathcal{I}_{\text{up}}(j)} P_{b,h,i,j} \right). \end{aligned} \tag{4}$$

Overall, we aggregate these values to define the salience score s, which captures a token's strongest influence across past, present, and future interactions, as formulated below:

$$s_{b,j} = \begin{cases} \frac{\text{diag}_{b,j} + \text{low}_{b,j}}{2}, & 0 \le j < L_B, \\ \frac{\text{up}_{b,j} + \text{diag}_{b,j} + \text{low}_{b,j}}{3}, & L_B \le j < L - L_B, \\ \frac{\text{diag}_{b,j} + \text{up}_{b,j}}{2}, & L - L_B \le j < L. \end{cases} \tag{5}$$

3.3 Salience Estimation Head

A fundamental challenge in the AR video diffusion is that future information is unobservable, while the salience score depends on both historical context and anticipated future utility. To this end, we introduce a learnable Salience Estimation Head (SEH) that infers salience score from existing representations while implicitly modeling future relevance.

To validate this design and analyze the effective placement of salience estimation, we compute salience scores at each Transformer layer of Wan2.1-14B and extract the top-k salient tokens, with $k = L/2$. Taking the final layer as a reference, we measure the overlap between its top-k tokens and those from earlier layers. Since AR video diffusion retains the KV Cache only at the final denoising step t_0 (Huang et al., 2025a), our analysis focuses on salience scores evaluated at t_0. As shown in Fig. 3(b), the top-k tokens across layers exhibit substantial overlap at t_0, indicating strong cross-layer consensus on token salience. Since the final layer operates on the most semantically stabilized representations, its top-k token selection reliably approximates that of earlier layers with negligible loss. Therefore, computing top-k token indices only at the final layer is sufficient, avoiding redundant per-layer selection and significantly improving efficiency in AR video generation. We incorporate our SEH into the final layer of AR video diffusion.

Under the attention mechanism (Vaswani et al., 2017), a token contributes meaningfully only when it is relevant, well-aligned, and informative, as encoded by the query, key, and value, respectively. Relying on only a subset of these components leads to incomplete salience estimation. Consequently, we concatenate QKV as the input to SEH to provide a joint and informative representation for salience prediction. Salience estimation can be formulated as a token-wise scalar regression task, where scores are used solely for ranking and cache retention. Moreover, QKV features are extracted after multiple Transformer layers and are already highly contextualized, obviating the need for additional spatial or temporal modeling. Accordingly, we adopt a lightweight MLP, termed Salience-MLP (S-MLP), as SEH, achieving a favorable trade-off between expressiveness and efficiency without introducing complex architectures such as attention or convolution. We further analyze the impact of the architectural choices through ablation studies.

3.4 Training and Inference of PaFu-KV

Based on our analysis in Sec. 3.3, we incorporate the S-MLP into the final layer of the AR video diffusion and activate it only at the t_0 timestep. At this stage, we concatenate the Q, K, and V

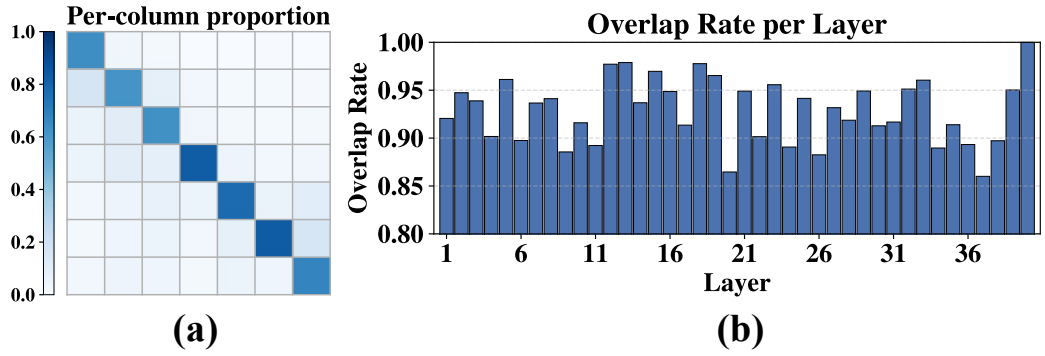


Figure 3: (a) A 7×7 count matrix constructed by uniformly partitioning key indices with maximal query responses and the token index range. (b) Overlap ratio of top-k salient token indices between each intermediate Transformer layer and the final layer of Wan2.1-14B at the final denoising step t_0.

representations along the feature dimension and pass the concatenated features to S-MLP to predict a scalar salience score for each token. With the estimated salience scores, PaFu-KV operates under two distinct regimes: training and inference, as indicated in Fig. 2(b) and (c).

Our training procedure is built on the DMD pipeline. We calculate salience score s_{real} from real score network as supervision based on Eq. 4 and 5. In parallel, we estimate salience scores from the generator G_θ using the SEH S_ϕ. Our salience score loss is defined as the SmoothL1 loss between the predicted and the target salience scores, formulated as:

$$\mathcal{L}_S = \text{SmoothL1}(S_\phi([Q, K, V]),\ s_{\text{real}})\,. \tag{6}$$

During training, to reduce memory footprint, we enable the gradient checkpointing technique in PyTorch (Paszke et al., 2019), which avoids storing intermediate activations during the forward pass and recomputes them during backpropagation. In-place modification of the KV Cache leads to incorrect gradients due to inconsistent cache states across recomputed process. To address this issue, we adopt an index-based cache management strategy. Specifically, instead of explicitly reorganizing the KV Cache, we maintain a token index cache I_s, which records the indices of the top-k salient tokens selected for each generated chunk based on both current and historical salience scores. The index cache is updated and rolled out synchronously with the KV Cache, ensuring consistent correspondence between token indices and KV entries throughout training. Consequently, Eq. 2 is modified as follows:

$$x^i_{t_{j-1}} = \Psi\Big(G_\theta\Big(x^i_{t_j}, t_j, \text{Select}(KV; I_s)\Big), t_{j-1}\Big)\,, \tag{7}$$

During inference, we initialize an empty KV Cache and perform AR video generation in a chunk-wise fashion. After each chunk is generated, we obtain its corresponding KV entries and predicted salience scores. Once KV Cache reaches its capacity, we concatenate the salience scores of the current chunk with those of the historical cache and retain the top-k salient tokens to update the KV Cache, evicting the remaining entries. The next chunk is generated conditioned on the updated cache, and this procedure is iteratively applied until the full video is generated.

4 Experiments

4.1 Experimental Settings

Experimental Implementation. Our PaFu-KV was trained under the Self Forcing (Huang et al., 2025a) paradigm. To enable long video generation, we implemented the streaming long tuning strategy proposed in Longlive (Yang et al., 2025a). Our PaFu-KV is built on Wan2.1-T2V-1.3B (Wan et al., 2025), a flow matching model based on the DiT architecture originally designed for generating 5-second clips, and distilled from Wan2.1-T2V-14B in DMD training pipeline. We train our model in

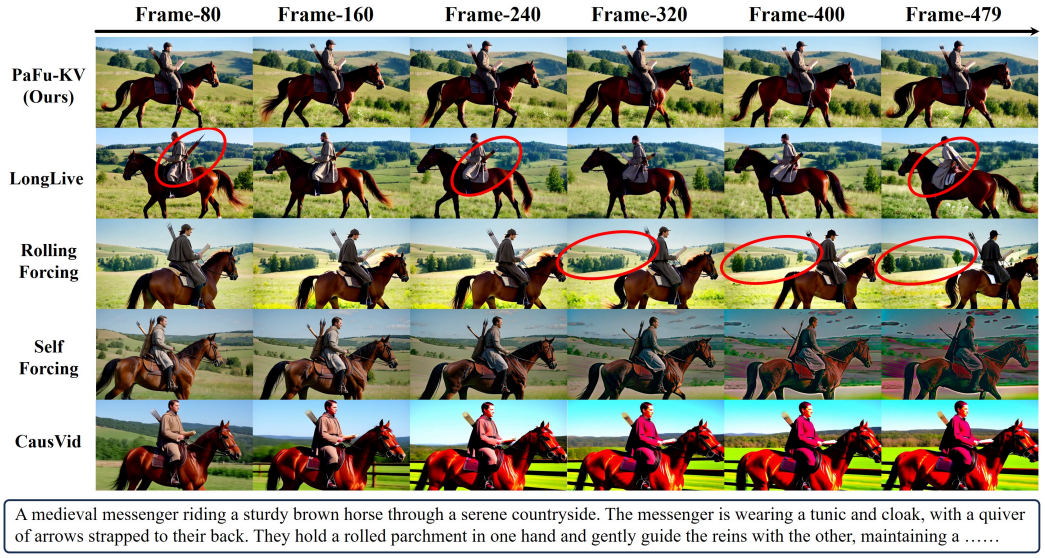


Figure 4: Qualitative experimental results on 30-second videos. We compare PaFu-KV with representative open-source autoregressive video generation models. We explicitly mark the inconsistent regions using a red circle in the figure for better visualization.

Table 1: Comparison results on short video generation. Experiments are conducted on a single H100 GPU, and evaluation scores are calculated on the official prompts of VBench (Huang et al., 2024).

Model Types	Methods	Real-time Performance			Evaluation Scores		
		Params	Resolution	FPS ↑	Total↑	Quality↑	Semantic↑
Bidirectional Diffusion	LTX-Video(HaCohen et al., 2024)	1.9B	768×512	8.92	80.00	82.30	70.79
	Wan(Wan et al., 2025)	1.3B	832×480	0.78	84.26	85.30	80.09
AR Diffusion	SkyReels-V2(Chen et al., 2025)	1.3B	960×540	0.49	82.67	84.70	74.53
	MAGI-1(Teng et al., 2025)	4.5B	832×480	0.19	79.18	82.04	67.74
	CausVid(Yin et al., 2025)	1.3B	832×480	17.0	81.20	84.05	69.80
	NOVA(Deng et al., 2024)	0.6B	768×480	0.88	80.12	80.39	<u>79.05</u>
	Pyramid Flow(Jin et al., 2024)	2.0B	640×384	6.7	81.72	84.74	69.62
	Self Forcing(Huang et al., 2025a)	1.3B	832×480	17.0	<u>84.31</u>	85.07	**81.28**
	LongLive(Yang et al., 2025a)	1.3B	832×480	20.7	84.87	<u>86.97</u>	76.47
	Rolling Forcing(Liu et al., 2025)	1.3B	832×480	15.79	81.22	84.08	69.78
	Ours	1.3B	832×480	**23.6**	**85.12**	**88.56**	71.36

two stages. In the first stage, we trained the SEH and AR video generator for 500 iterations without KV Cache eviction. In the second stage, we train both components for an additional 2,500 iterations with KV Cache eviction activated. Training runs for 3,000 iterations on 16 H200 GPUs with gradient accumulation of 4, spanning approximately two days. We optimize the AR generator and SEH using separate AdamW optimizers with the same learning rate of 1×10^{-5}. The critical models in the DMD pipeline are optimized using additional AdamW optimizers with a learning rate of 2×10^{-6}. All optimizers use the same momentum parameters, with $\beta_1 = 0.0$ and $\beta_2 = 0.999$. Our model is trained on extended VidProM prompts proposed in Longlive (Yang et al., 2025a). The KV Cache size is set to 4680, and ablation studies on the cache size are provided in Sec. 4.3. Additionally, we incorporate the frame-sink mechanism from LongLive (Yang et al., 2025a) and evaluate its effect via ablation. Additional training details are provided in the Supplementary Material.

Evaluation Protocol. We compare our method against a range of existing open-source video generation models under two evaluation protocols: short video generation and long video generation. For short video generation, we include LTXVideo (HaCohen et al., 2024), Wan2.1 (Wan et al., 2025), SkyReels-V2 (Chen et al., 2025), MAGI-1 (Teng et al., 2025), CausVid (Yin et al., 2025), NOVA (Deng et al., 2024), Pyramid Flow (Jin et al., 2024), Self Forcing (Huang et al., 2025a) LongLive (Yang et al., 2025a), and Rolling Forcing. For long video generation, we compare with

Table 2: Comparison results on long video generation. Evaluation scores are calculated on VBench-Long (Huang et al., 2025b). Best results are in bold, and second-best results are underlined.

Methods	Evaluation Scores					
	Subject Cons. ↑	Background Cons. ↑	Motion Smooth. ↑	Aesthetic Qual. ↑	Imaging Qual. ↑	Quality Drift ↑
SkyReels-V2(Chen et al., 2025)	89.23	93.45	98.76	61.55	62.90	5.59
MAGI-1(Teng et al., 2025)	90.86	93.25	**99.20**	59.91	59.87	2.15
CausVid(Yin et al., 2025)	87.99	89.99	98.09	60.95	66.38	2.18
Self Forcing(Huang et al., 2025a)	86.48	90.29	98.47	60.54	68.68	1.66
LongLive(Yang et al., 2025a)	90.12	91.42	<u>99.07</u>	**62.94**	69.46	0.59
Rolling Forcing(Liu et al., 2025)	<u>92.80</u>	<u>93.71</u>	98.70	62.39	<u>70.75</u>	**0.01**
Ours	**93.91**	**94.14**	98.14	<u>62.48</u>	**71.23**	<u>0.29</u>

SkyReels-V2 (Chen et al., 2025), CausVid (Yin et al., 2025), Self Forcing (Huang et al., 2025a), and LongLive (Yang et al., 2025a). For clarity and fairness, we categorize all baselines into two groups according to their generation paradigms: bidirectional diffusion models and AR models. Specifically, the bidirectional diffusion models include LTXVideo and Wan2.1-1.3B, while the remaining fall under the AR model category.

Evaluation Metrics. For short video generation, we generate 5-second videos using extended VBench (Huang et al., 2024) prompts following Self Forcing (Huang et al., 2025a) and evaluate the results with VBench metrics, where the total score is computed as a weighted average of a quality score and a semantic score. For long video generation, we generate 30-second videos at 16 fps with a resolution of 832×480 using the official prompts from VBench-Long (Huang et al., 2025b). We evaluate the generated videos using subject consistency, background consistency, motion smoothness, aesthetic quality, and imaging quality. Additionally, we compute the absolute difference in imaging quality to assess quality drift as proposed by Zhang & Agrawala (2025), and evaluate real-time performance in terms of generation throughput.

4.2 Experimental Results

Quantitative Results. Comprehensive evaluations are conducted on short and long video generation benchmarks, namely VBench (Huang et al., 2024) and VBench-Long (Huang et al., 2025b). As shown in Table 1, for short video generation, our PaFu-KV achieves a higher total score than Self Forcing and LongLive, as well as a higher semantic score compared to Rolling Forcing. This improvement can be attributed to our selection of more salient tokens during generation. In addition, by reducing the KV Cache size, PaFu-KV outperforms existing methods in real-time performance, achieving higher throughput. Furthermore, as presented in Table 2, PaFu-KV achieves the best performance in subject and background consistency, while exhibiting lower quality drift with only marginal degradation in motion smoothness. Overall, quantitative results on both short and long video generation indicate that PaFu-KV effectively accelerates AR diffusion-based generation while improving temporal consistency with negligible quality degradation.

Qualitative Results. Qualitative analysis comparing our method with existing open-source approaches demonstrates the effectiveness of PaFu-KV. In particular, regions with noticeable inconsistencies are highlighted for clearer comparison. As shown in Fig. 4, CausVid (Yin et al., 2025) and Self Forcing (Huang et al., 2025a) exhibit severe temporal drift caused by error accumulation when generating long videos. Rolling Forcing (Liu et al., 2025) suffers from background inconsistency across frames. In contrast, LongLive (Yang et al., 2025a) fails to preserve subject consistency over long temporal horizons. Compared with these methods, our approach achieves substantially improved temporal consistency in long video generation and demonstrates comparable robustness to error accumulation, particularly when contrasted with CausVid and Self Forcing. Additional qualitative results are provided in the Supplementary Material. For qualitative results in video format, additional materials are available at: `https://anonymous.4open.science/r/PaFu-KV-B7EE/README.md`.

4.3 Ablation Study

To assess the contribution of key components in PaFu-KV, we conduct ablation experiments focusing on four critical factors: (1) the architecture of SEH, (2) the computation of the salience score, (3) the size of the reduced KV Cache, and (4) the implementation of frame sink. All ablation experiments are conducted on VBench-Long (Huang et al., 2025b) using the same evaluation protocol as long video generation. As summarized in Table 3, both the S-MLP design in SEH and the proposed salience score computation have a significant impact on long video generation performance. For clarity, *Subj.*, *Back.*, *Mot.*, *Aes.*, *Imag.*, and *Drift* denote subject consistency, background consistency, motion smoothness, aesthetic quality, imaging quality, and quality drift, respectively. Visualization results of the ablation study are provided in the Supplementary Material.

The impact of SEH architecture. We evaluate the impact of the SEH by implementing SEH with a multi-head attention mechanism and a 1D convolution configured with a kernel size of 3. As indicated in Table 3, despite incorporating a larger number of parameters, the generation quality degrades, suggesting that increased architectural complexity does not improve long video generation.

The impact of salience score computation. To examine the effectiveness of our proposed salience score computation, we replace it with simpler alternatives, including global maximum pooling and global average pooling. As indicated in Table 3, global maximum pooling yields poorer subject and background consistency and higher quality drift, while global average results in severe degradation.

The impact of reduced KV Cache size. We further investigate the effect of the Cache size, where the cache size of the full model is 4680. We evaluate our PaFu-KV under four cache size configurations: 9360, 6240, 3120, and 1560 tokens. Each configuration is a multiple of 1560, which corresponds to the latent tokens number of a single video frame. Experimental results show that cache sizes of 9360 and 6240 tokens achieve performance comparable to the full model. However, when the cache size is reduced below 4680 tokens, a degradation in generation quality is observed.

The impact of frame sink. We conduct an ablation study on the frame-sink mechanism. The results indicate that removing frame sink leads to marginal degradation in long video generation performance, suggesting that it is not a primary contributor to the performance gains of PaFu-KV.

Table 3: Ablation experiments are conducted following the long video generation protocol.

Factors	**Settings**	Evaluation scores					
		Subj.↑	Back.↑	Mot.↑	Aes.↑	Imag.↑	Drift↑
SEH architecture	Conv	85.47	85.13	94.38	58.04	67.04	1.68
	MHA	88.65	88.71	95.74	60.24	69.47	1.18
Salience score computation	Max	87.65	88.71	94.34	60.18	69.33	1.45
	Avg	78.87	79.78	76.94	52.94	58.49	2.54
KV Cache size	9360	94.12	93.89	98.04	62.06	71.89	0.25
	6240	93.45	94.29	98.54	62.57	70.76	0.31
	3120	85.26	86.48	82.11	53.48	64.58	1.12
	1560	74.48	75.18	79.44	49.74	58.69	2.36
Frame sink	w/o frame sink	93.12	93.09	97.84	61.89	70.45	0.44
Full Model	-	**93.91**	**94.14**	**98.14**	**62.48**	**71.23**	**0.29**

5 Conclusion

We introduce PaFu-KV, a past- and future-informed KV Cache policy guided by token salience. By analyzing bidirectional diffusion models, we identify salience as a key factor for long-horizon generation stability and propose a score that disentangles past, present, and future attention to mitigate diagonal bias. A lightweight SEH then distills this global salience information from a bidirectional teacher into a causal AR video diffusion model. During inference, PaFu-KV leverages the predicted salience scores to selectively retain informative tokens while evicting those with marginal generative utility, thereby maintaining a compact KV Cache and reducing interference from drifting representations. Experiments on short- and long-video benchmarks show that PaFu-KV accelerates inference with a compact KV Cache while preserving generation quality and improving temporal coherence with limited error accumulation.

AI USE STATEMENT

We used ChatGPT to assist with the implementation and debugging of parts of the proposed method, the interpretation and presentation of experimental results, and the writing, editing, and translation of the manuscript. ChatGPT was not used to formulate the core research ideas, hypotheses, methodology, or experimental design. All AI-assisted code was reviewed, executed, and tested by the authors, and all experiments were conducted by the authors. The authors independently verified all experimental results, analyses, manuscript content, references, and citations. AI tools were not used to generate experimental results or fabricate data. The authors take full responsibility for the content of this work.

REFERENCES

Zefan Cai, Yichi Zhang, Bofei Gao, Yuliang Liu, Yucheng Li, Tianyu Liu, Keming Lu, Wayne Xiong, Yue Dong, Junjie Hu, et al. Pyramidkv: Dynamic kv cache compression based on pyramidal information funneling. *arXiv preprint arXiv:2406.02069*, 2024.

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.

Guibin Chen, Dixuan Lin, Jiangping Yang, Chunze Lin, Junchen Zhu, Mingyuan Fan, Hao Zhang, Sheng Chen, Zheng Chen, Chengcheng Ma, et al. Skyreels-v2: Infinite-length film generative model. *arXiv preprint arXiv:2504.13074*, 2025.

Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in neural information processing systems*, 35:16344–16359, 2022.

Haoge Deng, Ting Pan, Haiwen Diao, Zhengxiong Luo, Yufeng Cui, Huchuan Lu, Shiguang Shan, Yonggang Qi, and Xinlong Wang. Autoregressive video generation without vector quantization. *arXiv preprint arXiv:2412.14169*, 2024.

Yuan Feng, Junlin Lv, Yukun Cao, Xike Xie, and S Kevin Zhou. Ada-kv: Optimizing kv cache eviction by adaptive budget allocation for efficient llm inference. *arXiv preprint arXiv:2407.11550*, 2024.

Kaifeng Gao, Jiaxin Shi, Hanwang Zhang, Chunping Wang, Jun Xiao, and Long Chen. Ca2-vdm: Efficient autoregressive video diffusion model with causal generation and cache sharing. *arXiv preprint arXiv:2411.16375*, 2024.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *First conference on language modeling*, 2024.

Yoav HaCohen, Nisan Chiprut, Benny Brazowski, Daniel Shalem, Dudu Moshe, Eitan Richardson, Eran Levin, Guy Shiran, Nir Zabari, Ori Gordon, et al. Ltx-video: Realtime video latent diffusion. *arXiv preprint arXiv:2501.00103*, 2024.

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*, 2025a.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. Vbench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21807–21818, 2024.

Ziqi Huang, Fan Zhang, Xiaojie Xu, Yinan He, Jiashuo Yu, Ziyue Dong, Qianli Ma, Nattapol Chanpaisit, Chenyang Si, Yuming Jiang, et al. Vbench++: Comprehensive and versatile benchmark suite for video generative models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025b.

Yuming Jiang, Tianxing Wu, Shuai Yang, Chenyang Si, Dahua Lin, Yu Qiao, Chen Change Loy, and Ziwei Liu. Videobooth: Diffusion-based video generation with image prompts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6689–6700, 2024.

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.

Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. Hunyuanvideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024.

Wuyang Li, Wentao Pan, Po-Chien Luan, Yang Gao, and Alexandre Alahi. Stable video infinity: Infinite-length video generation with error recycling. *arXiv preprint arXiv:2510.09212*, 2025.

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. *Advances in Neural Information Processing Systems*, 37:22947–22970, 2024.

Feng Liang, Akio Kodaira, Chenfeng Xu, Masayoshi Tomizuka, Kurt Keutzer, and Diana Marculescu. Looking backward: Streaming video-to-video translation with feature banks. *arXiv preprint arXiv:2405.15757*, 2024.

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.

Yunhong Lu, Yanhong Zeng, Haobo Li, Hao Ouyang, Qiuyu Wang, Ka Leong Cheng, Jiapeng Zhu, Hengyuan Cao, Zhipeng Zhang, Xing Zhu, et al. Reward forcing: Efficient streaming video generation with rewarded distribution matching distillation. *arXiv preprint arXiv:2512.04678*, 2025.

Yue Ma, Kunyu Feng, Zhongyuan Hu, Xinyu Wang, Yucheng Wang, Mingzhe Zheng, Xuanhua He, Chenyang Zhu, Hongyu Liu, Yingqing He, et al. Controllable video generation: A survey. *arXiv preprint arXiv:2507.16869*, 2025.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

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

Ziran Qin, Yuchen Cao, Mingbao Lin, Wen Hu, Shixuan Fan, Ke Cheng, Weiyao Lin, and Jianguo Li. Cake: Cascading and adaptive kv cache eviction with layer preferences. *arXiv preprint arXiv:2503.12491*, 2025.

Sucheng Ren, Qihang Yu, Ju He, Xiaohui Shen, Alan Yuille, and Liang-Chieh Chen. Beyond next-token: Next-x prediction for autoregressive visual generation. *arXiv preprint arXiv:2502.20388*, 2025.

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

Prajwal Singhania, Siddharth Singh, Shwai He, Soheil Feizi, and Abhinav Bhatele. Loki: Low-rank keys for efficient sparse attention. *Advances in Neural Information Processing Systems*, 37: 16692–16723, 2024.

Hanlin Tang, Yang Lin, Jing Lin, Qingsen Han, Shikuan Hong, Yiwu Yao, and Gongyi Wang. Razorattention: Efficient kv cache compression through retrieval heads. *arXiv preprint arXiv:2407.15891*, 2024.

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.

Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. *Advances in neural information processing systems*, 37:84839–84865, 2024.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

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

Xunzhi Xiang, Yabo Chen, Guiyu Zhang, Zhongyu Wang, Zhe Gao, Quanming Xiang, Gonghu Shang, Junqi Liu, Haibin Huang, Yang Gao, et al. Macro-from-micro planning for high-quality and parallelized autoregressive long video generation. *arXiv preprint arXiv:2508.03334*, 2025.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. *arXiv preprint arXiv:2309.17453*, 2023.

Haiwei Xue, Xiangyang Luo, Zhanghao Hu, Xin Zhang, Xunzhi Xiang, Yuqin Dai, Jianzhuang Liu, Zhensong Zhang, Minglei Li, Jian Yang, et al. Human motion video generation: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

Shuai Yang, Wei Huang, Ruihang Chu, Yicheng Xiao, Yuyang Zhao, Xianbang Wang, Muyang Li, Enze Xie, Yingcong Chen, Yao Lu, et al. Longlive: Real-time interactive long video generation. *arXiv preprint arXiv:2509.22622*, 2025a.

Xiangpeng Yang, Linchao Zhu, Hehe Fan, and Yi Yang. Videograin: Modulating space-time attention for multi-grained video editing. In *The Thirteenth International Conference on Learning Representations*, 2025b.

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.

Jung Yi, Wooseok Jang, Paul Hyunbin Cho, Jisu Nam, Heeji Yoon, and Seungryong Kim. Deep forcing: Training-free long video generation with deep sink and participative compression. *arXiv preprint arXiv:2512.05081*, 2025.

Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and Bill Freeman. Improved distribution matching distillation for fast image synthesis. *Advances in neural information processing systems*, 37:47455–47487, 2024a.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6613–6623, 2024b.

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

Hu Yu, Biao Gong, Hangjie Yuan, DanDan Zheng, Weilong Chai, Jingdong Chen, Kecheng Zheng, and Feng Zhao. Videomar: Autoregressive video generatio with continuous tokens. *arXiv preprint arXiv:2506.14168*, 2025a.

Yifei Yu, Xiaoshan Wu, Xinting Hu, Tao Hu, Yangtian Sun, Xiaoyang Lyu, Bo Wang, Lin Ma, Yuewen Ma, Zhongrui Wang, et al. Videossm: Autoregressive long video generation with hybrid state-space memory. *arXiv preprint arXiv:2512.04519*, 2025b.

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

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.

A Appendix

B Additional Implementation Details

KV Cache and Salience Score List. In the full PaFu-KV model, we set the KV Cache size to 4680 tokens, corresponding to three times the number of latent tokens per video frame, and use this configuration consistently during training and inference. During training, we enable gradient checkpointing in PyTorch to reduce GPU memory footprint by avoiding the storage of intermediate activations and recomputing them during backpropagation. Each training iteration operates on a 5-second video segment, corresponding to 21 latent frames. As a result, the effective KV Cache size during training is required to be substantially larger than that used at inference. Specifically, the cache size is first expanded to $21 \times 1560 = 32{,}760$ tokens, corresponding to 21 latent frames. When adopting long video tuning following LongLive Yang et al. (2025a), the KV Cache is further enlarged to $33 \times 1560 = 51{,}480$ tokens to support extended AR rollouts. We set the salience score list to have the same size as the KV Cache, storing 51,480 scalar values, while the KV Cache stores 51,480 tokens each with 12 attention heads and a feature dimension of 128. Consequently, the memory overhead of the salience score list is negligible compared to that of the KV Cache. As described in Sec. 3.4, we maintain an index structure to select the top-k salient tokens for each denoising chunk. When a latent segment of 21 frames is denoised, both the DMD loss and the salience score loss are computed to optimize the AR video diffusion model. Although the KV Cache is configured with a large capacity during training, the effective KV Cache used for attention computation within each denoising chunk remains fixed at 4680 tokens, consistent with the inference-time setting. Since each timestep denoises a video chunk consisting of 4680 tokens, we set $L_B = 4680$ as the block length in the Spatialtemporal-balanced Salience Estimation used by the real score network. During inference, the KV Cache size is accordingly set to 4680 tokens.

S-MLP. We adopt a lightweight MLP (S-MLP) as the architecture for the Salience Estimation Head (SEH). The S-MLP consists of two linear layers with a SiLU activation in between, and is incorporated into the final layer of the AR video diffusion model. The query, key, and value tensors, each of shape $\mathbb{R}^{B\times N\times L\times D}$ with $B = 1$, $N = 12$, $L = 4680$, and $D = 128$, are first concatenated. We then permute the tensor to exchange the N and L dimensions and reshape it by merging the attention-head dimension N with the feature dimension D, resulting in a merged feature dimension $D_M = N \times D = 1536$. The resulting query, key, and value representations are further concatenated along the D_M dimension and fed into the S-MLP. The input, hidden, and output dimensions of the S-MLP are 4608, 1024, and 12, respectively. Finally, an averaging operation is applied over the output dimension of size 12 to produce the salience score list, which is subsequently used for salience-based token selection and KV Cache management.

C Streaming Long Tuning and KV Re-caching in LongLive.

Streaming Long Tuning. Our PaFu-KV is trained using Streaming Long Tuning proposed by Yang et al. (2025a). Streaming Long Tuning is designed to address the error accumulation that arises when AR video models trained on short clips are deployed for long-horizon generation. Instead of adopting a train-short–test-long regime, this procedure explicitly aligns training with inference by simulating long AR video diffusion rollouts. Concretely, the training proceeds in a streaming manner: the model first generates a short video clip from an empty context and receives supervision from a teacher model (Wan2.1-14B) on the generated clip. The generated frames are then stored in the KV Cache and treated as fixed historical context for the next iteration, in which the model generates the subsequent clip conditioned on its own past predictions. This rolling process continues until a predefined maximum video length is reached, after which training restarts from a fresh

context. To ensure scalability, gradients are computed only for the newly generated clip at each iteration, while earlier frames are detached and used solely as causal context, thereby bounding memory usage by the clip length rather than the total video duration. By repeatedly exposing the model to long, self-generated, and progressively imperfect contexts during training, streaming long tuning mitigates error accumulation and temporal drift, and effectively aligns the training distribution with inference-time conditions.

KV Re-caching. KV re-caching is introduced to facilitate smooth and semantically consistent prompt switching during interactive long video generation. In causal AR models, prompt information is repeatedly injected through cross-attention layers and propagated forward by self-attention, causing prompt semantics to be deeply embedded in the KV Cache. As a result, simply retaining the cache across prompt switches often leads to delayed or incorrect adherence to new prompts, whereas clearing the cache entirely introduces abrupt visual discontinuities. KV re-caching resolves this trade-off by selectively refreshing the cache at prompt boundaries. Specifically, at the first generation step following a prompt switch, the model recomputes the KV Cache by jointly encoding the already generated video frames as visual context together with the new prompt, thereby removing residual semantics from the previous prompt while preserving motion and appearance continuity. Subsequent frames are then generated autoregressively using this refreshed cache without further intervention. This re-caching operation is performed only once per prompt switch and incurs minimal additional overhead. By integrating KV re-caching into both training and inference, the model learns to rapidly adapt to new prompts while maintaining smooth temporal transitions, ensuring consistent visual quality and prompt adherence in interactive long video generation.

D ALGORITHM

To facilitate reproducibility and provide a precise procedural description of our method, we present the complete algorithmic workflow of the training and inference of PaFu-KV in Alg. 1 and 2, respectively. Additionally, computing the ground-truth salience scores in the teacher model requires explicit calculation of attention weights, which can incur substantial GPU memory overhead during training, as the teacher employs full bidirectional self-attention and involves all tokens in the attention computation. To mitigate this issue, we adopt a block-wise attention computation strategy to reduce memory footprint, as detailed in Algorithm 3. The pseudocode summarizes the training and inference procedures described in Sec. 3.4, including KV Cache management, salience score computation, and token selection. All operations adhere to the notations and settings defined in the main text.

E ADDITIONAL QUALITATIVE RESULTS.

In this section, we provide additional qualitative visual comparisons in Fig. 5, Fig. 6, and Fig. 7. The prompt used for each example is shown below the corresponding figure, and regions exhibiting noticeable inconsistencies are highlighted using red circles. The results indicate that our method achieves strong long-horizon consistency. Qualitative results in video format are provided at `https://anonymous.4open.science/r/PaFu-KV-B7EE/README.md`. Additionally, we provide visualizations for the ablation study in Fig 8. As illustrated in the figure, the averaging strategy results in substantial error accumulation, whereas attention-based and naive max strategies markedly reduce temporal consistency.

F DISCUSSION AND FUTURE WORK.

In this section, we provide further discussion of PaFu-KV and outline directions for future work. From a practical deployment perspective, in addition to generation speed, GPU memory footprint is a critical factor when deploying video generation models. Although PaFu-KV reduces the KV Cache size by selecting top-k salient tokens and thereby accelerates inference, the resulting reduction in memory usage is relatively limited. This phenomenon is common among KV-cache optimization approaches in the AR video diffusion domain. The primary reason is the widespread adoption of FlashAttention Dao et al. (2022), which significantly reduces the memory footprint of the attention mechanism, the dominant source of memory consumption in Transformer architectures.



Figure 5: Qualitative experimental results on 30-second videos. We compare PaFu-KV with representative open-source autoregressive video generation models. We explicitly mark the inconsistent regions using a red circle in the figure for better visualization.

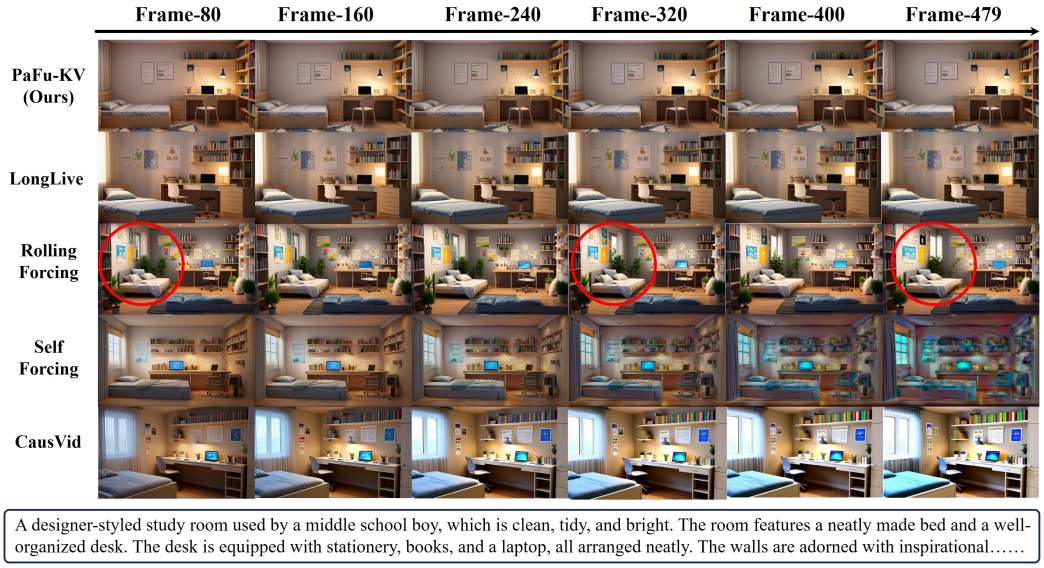


Figure 6: Qualitative experimental results on 30-second videos. We compare PaFu-KV with representative open-source autoregressive video generation models. We explicitly mark the inconsistent regions using a red circle in the figure for better visualization.

Specifically, FlashAttention computes attention in a block-wise manner using a streaming softmax, discarding intermediate $QK^\top$ scores and softmax probabilities on the fly instead of materializing the full attention matrix. As a result, it trades recomputation for memory efficiency and reduces the memory complexity from $O(N^2)$ to approximately $O(N)$. Under this regime, further reducing the KV Cache size yields only marginal memory savings. Consequently, reducing GPU memory footprint most effectively requires reducing the model parameter. In future work, we plan to further investigate AR video diffusion models with the goal of improving deployment efficiency while



Figure 7: Qualitative experimental results on 30-second videos. We compare PaFu-KV with representative open-source autoregressive video generation models. We explicitly mark the inconsistent regions using a red circle in the figure for better visualization.

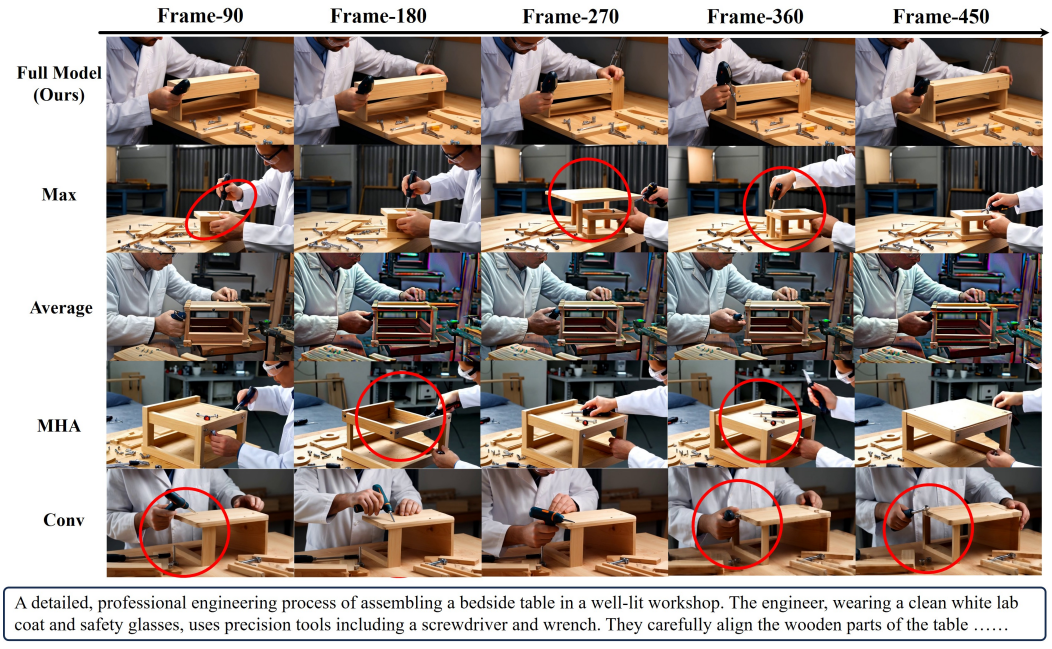


Figure 8: Visualization of ablation study.

maintaining high generation quality, thereby enhancing the practical applicability of long video generation systems.

Algorithm 1 PaFu-KV Training

Require: Denoise timesteps $\{t_0, t_1, \ldots, t_T\}$
Require: Number of video frames N
Require: AR diffusion model G_θ (returns KV embeddings via G_θ^{KV})
Require: Salience MLP $S_{\hat{\theta}}$ (takes intermediate features and outputs salience score)
Require: Token number of top salience score k
loop
Initialize KV Cache $\mathbf{KV} \leftarrow [\,]$
Initialize generator salience score list $\mathbf{SL_g} \leftarrow [\,]$
Initialize list of indices for selected top-k token in KV Cache $\mathbf{IL} \leftarrow [\,]$
Sample $s \sim \mathrm{Uniform}(0, 2, \ldots, T)$
for $i = 1, \ldots, N$ **do**
  Initialize $x^i_{t_T} \sim \mathcal{N}(0, I)$
  for $j = T, \ldots, s$ **do**
    if $j = s$ **then**
      Enable gradient computation
      Set $\hat{x}^i_0 \leftarrow G_\theta(x^i_{t_j};\, t_j,\, \mathrm{Select}(\mathbf{KV}; \mathbf{IL}))$
      Enable salience computation
      Set $\{q,k,v\}, \mathbf{kv}^i \leftarrow G_\theta(\hat{x}^i_0; 0, \mathrm{Select}(\mathbf{KV}; \mathbf{IL}))$
      Set $s^i_g \leftarrow S_{\hat{\theta}}(\{q, k, v\})$
      $\mathbf{KV}.\mathrm{append}(\mathbf{kv}^i)$
      $\mathbf{SL_g}.\mathrm{append}(s^i_g)$
      Indices $ids^i \leftarrow \text{Top-}k(\mathbf{SL_g})$
      $\mathbf{IL}.\mathrm{append}(ids^i)$
      Disable salience computation
      Disable gradient computation
    else
      Disable gradient computation
      Set $\hat{x}^i_0 \leftarrow G_\theta(x^i_{t_j};\, t_j,\, \mathrm{Select}(\mathbf{KV}; \mathbf{IL}))$
      Sample $\epsilon \sim \mathcal{N}(0, I)$
      Set $x^i_{t_{j-1}} \leftarrow \Psi(\hat{x}^i_0, \epsilon, t_{j-1})$
    end if
  end for
end for
Compute distribution matching loss $\mathcal{L}_{\mathrm{dm}}$
Compute salience score list in real score network $\mathbf{SL}_r$
Set $\mathcal{L}_{\mathrm{s}} \leftarrow \mathrm{SmoothL1}(\mathbf{SL}_r, \mathbf{SL_g})$
Update θ by minimizing $\mathcal{L}_{\mathrm{dm}}$ and $\mathcal{L}_{\mathrm{s}}$
end loop

Algorithm 2 PaFu-KV Inference

Require: KV Cache size of L tokens
Require: Denoise timesteps $\{t_0, \ldots, t_T\}$
Require: Number of generated video frames M
Require: AR diffusion model G_θ, salience MLP $S_{\hat{\theta}}$
Require: Token number of top salience score k
Initialize KV Cache $\mathbf{KV} \leftarrow [\,]$
Initialize salience score list $\mathbf{SL} \leftarrow [\,]$
Sample $s \sim \mathrm{Uniform}(0, 1, \ldots, T)$
for $i = 1, \ldots, M$ **do**
  Initialize $x^i_{t_T} \sim \mathcal{N}(0, I)$
  for $j = T, \ldots, 0$ **do**
    Set $\hat{x}^i_0 \leftarrow G_\theta(x^i_{t_j};\, t_j,\, \mathrm{Select}(\mathbf{KV}; \mathbf{IL}))$
    if $j = 0$ **then**
      Set $\mathbf{kv}^i, s^i \leftarrow G_\theta(\hat{x}^i_0; 0, \mathrm{Select}(\mathbf{KV}; \mathbf{IL}))$
      if $\|\mathbf{KV}\| = L$ **then**
        Indices $ids^i \leftarrow \text{Top-}k(\mathbf{SL_g})$
        $\mathbf{KV} \leftarrow \mathrm{Select}(\mathbf{KV};\, ids^i)$
        $\mathbf{SL} \leftarrow \mathrm{Select}(\mathbf{SL};\, ids^i)$
      end if
      $\mathbf{KV}.\mathrm{append}(\mathbf{kv}^i)$
      $\mathbf{SL}.\mathrm{append}(\mathbf{s}^i)$
    else
      Set $x^i_{t_{j-1}}$ via G_θ with $\mathrm{Select}(\mathbf{KV}; \mathbf{IL})$
    end if
  end for
end for

Algorithm 3 Block-wise Salience Score Computation

Require: Query, Key $\mathbf{Q}$, $\mathbf{K}$
Require: Chunk size c, block size b
Ensure: Salience score $\mathbf{s}$
Initialize $\mathbf{out}_{\text{upper}}, \mathbf{out}_{\text{diag}}, \mathbf{out}_{\text{down}} \leftarrow \mathbf{0}$
for $s = 0$ **to** $N - 1$ **step** c **do**
  $e \leftarrow s + c$
  $i \leftarrow \lfloor s/b \rfloor$ ▷ Block index
  $l \leftarrow i \cdot b, \quad h \leftarrow l + b$
  Extract query chunk $\mathbf{Q}_{s:e}$
  Compute attention weights $\mathbf{A} \leftarrow \mathbf{Q}_{s:e}\mathbf{K}^\top/\sqrt{D}$
  Aggregate salience $\mathbf{m} \leftarrow \max \mathbb{E}_{\text{head}}[\mathbf{A}]$
  $\mathbf{s}_{\text{down}}[:, :l] \leftarrow \max(\mathbf{out}_{\text{down}}[:, :l], \mathbf{m}[:, :l])$
  $\mathbf{s}_{\text{diag}}[:, l:h] \leftarrow \max(\mathbf{out}_{\text{diag}}[:, l:h], \mathbf{m}[:, l:h])$
  $\mathbf{s}_{\text{upper}}[:, h:] \leftarrow \max(\mathbf{out}_{\text{upper}}[:, h:], \mathbf{m}[:, h:])$
end for
Initialize $\mathbf{out} \in \mathbb{R}^{B \times N}$
Fuse salience scores:
return $\mathbf{out}$