STEP FORCING: TRAINING-FREE STEP-MATCHED KV CACHING FOR LONG VIDEO GENERATION

Anonymous authors
Paper under double-blind review

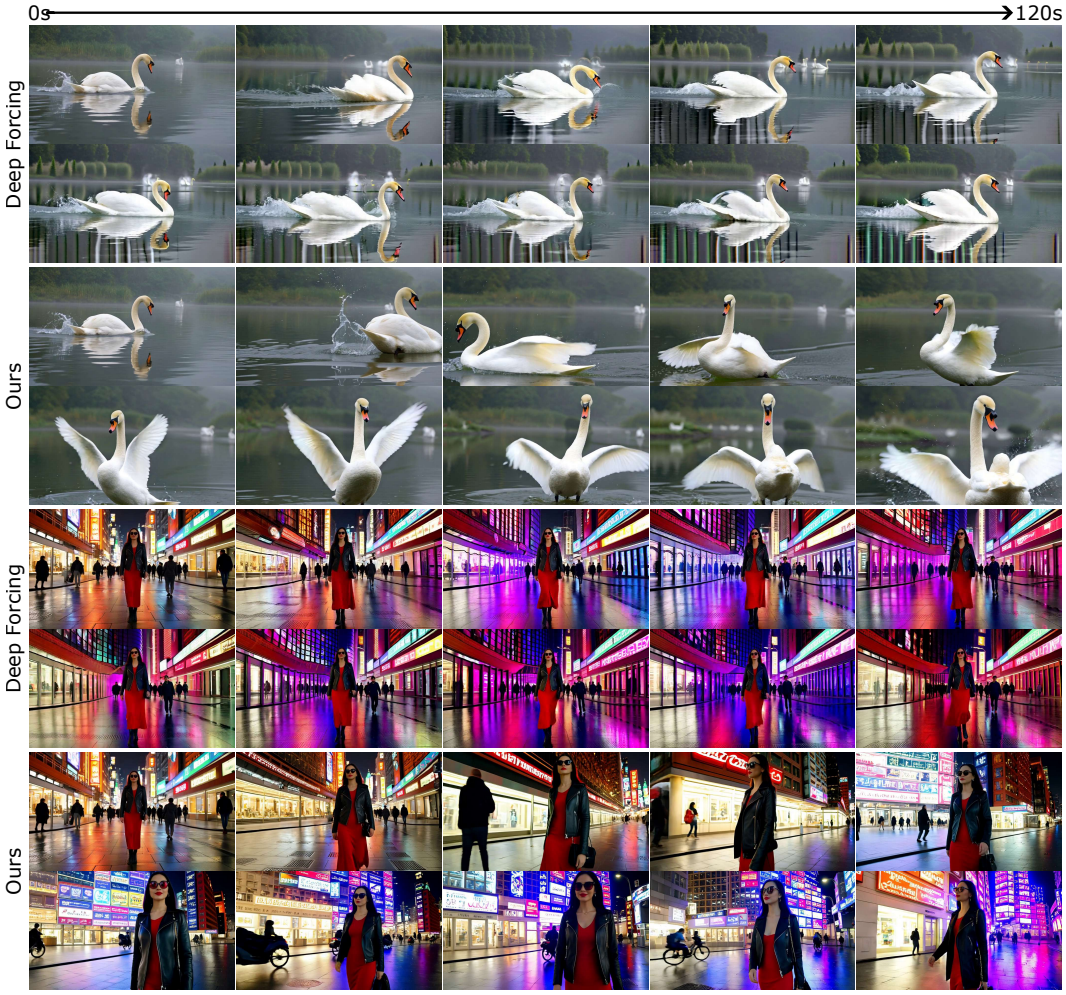


Figure 1: We propose **Step Forcing**, a plug-and-play strategy that can be integrated into existing autoregressive video diffusion baselines. Compared to Deep Forcing, **Step Forcing** mitigates model drift and preserves high generation fidelity, even for long-horizon generation up to 120 seconds.

ABSTRACT

Autoregressive video diffusion models adapt bidirectional video diffusion models into causal architectures, enabling long video generation through autoregressive rollout. However, these models often suffer from model drift over time, leading to progressive degradation in visual quality. Existing studies generally attribute such model drift to error accumulation during autoregressive rollout. Although existing methods alleviate error accumulation, they overlook a fundamental mismatch in the chunk-wise diffusion process, where the noisy state of the current chunk attends to the same KV caches extracted from previously generated clean chunks. To address this mismatch, we present **Step Forcing**, a training-free approach de-

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.

signed to mitigate model drift during autoregressive long video generation. We find that reusing fixed clean KV caches throughout the denoising trajectory introduces a step-wise representational mismatch and that noisy and clean KV caches serve complementary roles in preserving visual quality and temporal consistency. Building on these insights, **Step Forcing** reuses noisy KV caches from corresponding denoising steps while retaining a subset of clean anchor tokens. Our proposed **Step Forcing** effectively mitigates model drift by using step-matched KV tokens, enabling baseline models to generate long videos with high visual quality and temporal consistency. Results on benchmark quantitatively demonstrate that **Step Forcing** improves the performance of baseline models.

1 INTRODUCTION

Recent advances in diffusion transformer architectures (Peebles & Xie, 2023) have driven remarkable progress in video generation, enabling modern video diffusion models (Yang et al., 2024; Wang et al., 2025) to synthesize high-quality videos lasting several seconds with increasingly realistic visual appearance, coherent motion, and accurate semantic alignment. These video diffusion models typically rely on bidirectional attention.

However, bidirectional video diffusion models are generally trained to produce fixed-length video sequences and cannot readily generate videos beyond the sequence length used during training (e.g., 5s). To address this limitation, recent studies (Yin et al., 2025; Huang et al., 2026; Zhu et al., 2026) have developed autoregressive video diffusion models based on causal attention. These models generate videos sequentially in chunks through autoregressive rollout, using a KV caching mechanism to reuse contextual information from previously generated clean chunks and thereby enabling the generation length to be flexibly extended at inference time (Fig. 4a). Nevertheless, as autoregressive generation proceeds, these models often suffer from *model drift*, manifested as progressive oversaturation, over-sharpening, and semantic drift. Existing studies (Huang et al., 2026; Liu et al., 2025; Yi et al., 2025; Shaulov et al., 2026) generally attribute such model drift to error accumulation during autoregressive rollout. While these methods alleviate error accumulation, they overlook a fundamental mismatch in the chunk-wise diffusion process: *at each denoising step in current chunk, each noisy state attends to the same clean KV caches extracted from previously clean chunk*. Consequently, clean KV caches are misaligned with the evolving step-specific representations of the current noisy state.

Our key finding is that such mismatched clean KV caches contribute an important source of model drift (Sec. 3.2 and Fig. 2). This finding leads to our central insight: KV caches extracted at different denoising steps from previously generated chunks serve distinct roles (Fig. 3). First, KV caches extracted from the noisy states of previous chunks at the corresponding intermediate denoising timestep are better aligned with the step-specific representations of the current chunk, thereby preserving visual quality. However, these caches may lack sufficiently reliable content information to maintain temporal consistency during autoregressive rollouts. Second, KV caches extracted from the clean states of previous chunks after the final denoising step contain sufficient content information to maintain temporal consistency across chunks.

Motivated by the above observations, we introduce **Step Forcing**, a training-free approach for mitigating model drift during autoregressive long video generation. Specifically, **Step Forcing** modifies the reverse diffusion process such that, at each denoising step of the current chunk, the model reuses KV caches extracted from the noisy states of previous chunks at the corresponding timestep while retaining a subset of anchor tokens from their clean KV caches. In contrast to existing methods that rely solely on KV caches extracted from clean states to reduce error accumulation, **Step Forcing** provides the current chunk with step-matched KV caches, thereby mitigating model drift while preserving visual quality and temporal consistency over extended generation horizons.

We show that the proposed **Step Forcing** significantly mitigates model drift, enabling baseline models to generate long videos while maintaining high visual quality. Our main contributions are summarized as follows:

- Our systematic analysis reveals a mismatch between clean KV caches and the step-specific representations of the current chunk. This mismatch arises from reusing the same caches across denoising steps and contributes to model drift (Sec. 3.2).

- We propose **Step Forcing**, which provides the current chunk with step-matched KV caches, thereby mitigating model drift and enabling high-quality long video generation.
- We evaluate **Step Forcing** on four autoregressive long video generation methods, including both training-free and training-based approaches. Extensive experiments demonstrate that **Step Forcing** consistently mitigates model drift across different baseline methods and substantially improves the quality of generated long videos.

2 Related work

Bidirectional Video Models. Recent bidirectional video diffusion models, such as CogVideoX (Yang et al., 2024) and WAN (Wang et al., 2025), have demonstrated strong capabilities in generating high-quality videos by leveraging bidirectional attention across video frames. Despite their impressive generation quality, these models are generally designed for short video clips, typically around 5 seconds, and struggle to generate substantially longer videos directly.

Autoregressive Video Models. CausVid (Yin et al., 2025) adapts a pretrained bidirectional video diffusion model into an autoregressive causal model, enabling long video generation via autoregressive rollouts while requiring training only on short chunks. However, its autoregressive generation process relies on previously generated frames, which can lead to model drift over time (Yin et al., 2025). To mitigate this issue, recent works have explored both training-based and training-free solutions. Training-based methods (Huang et al., 2026; Liu et al., 2025; Yang et al., 2026; Xu et al., 2026; Mao et al., 2026; Xue et al., 2026; Yuan et al., 2026; Cao et al., 2026; Chen et al., 2026b) mitigate this issue by aligning training with autoregressive inference. Self-Forcing (Huang et al., 2026) aligns training with inference by performing autoregressive self-rollouts with KV caching during training, where each frame is generated conditioned on previously self-generated outputs. Rolling Forcing (Liu et al., 2025) mitigates long-horizon error accumulation by training the model to jointly denoise multiple frames within a rolling window while maintaining the initial-frame KV states as a global context anchor. However, training-based methods often require costly and time-consuming optimization, motivating the exploration of training-free solutions. While the methods discussed above focus on generating long videos from a single prompt, LongLive (Yang et al., 2026) targets interactive long video generation with streaming prompts. It finetunes a short-clip model and uses KV-recache and a frame-level attention sink to improve consistency. In this work, we focus on long video generation from a single prompt. Training-free methods (Yi et al., 2025; Shaulov et al., 2026; Xiang et al., 2026; Wu et al., 2026; Li et al., 2026; Meng et al., 2026; Luo et al., 2026; Zhao et al., 2026; Chen et al., 2026a) instead aim to suppress error propagation at inference time without updating model parameters. Deep Forcing (Yi et al., 2025) mitigates long-horizon error accumulation through KV-cache management, combining Deep Sink for persistent context preservation and Participative Compression for pruning redundant or degraded historical tokens. TokenTrim (Shaulov et al., 2026) mitigates temporal drift by identifying unstable latent conditioning tokens and pruning them from the autoregressive context at inference time, thereby preventing corrupted latent information from propagating to subsequent chunks. Pathwise TTC (Xiang et al., 2026) also operates purely at test time and alleviates temporal drift by anchoring the autoregressive sampling trajectory to the initial frame, without requiring additional training.

Recent studies have also explored methods for accelerating autoregressive video models (Lv et al., 2026; Samuel et al., 2026; Zou et al., 2026; Liu et al., 2026). Light Forcing (Lv et al., 2026) combines sparsity allocation across chunks with hierarchical sparse attention to reduce computation while retaining informative historical context. FAST-AR (Samuel et al., 2026) compresses KV caches using temporal correspondence and sparsifies self-attention and cross-attention through approximate nearest neighbor retrieval. Inspired by bidirectional diffusion models, HiAR (Zou et al., 2026) argues that context at corresponding noise levels provides sufficient signal for temporal consistency in autoregressive video generation. It introduces hierarchical denoising to accelerate generation through pipelined parallel inference across chunks. In contrast, we directly analyze autoregressive video models and find that KV caches at corresponding noise levels preserve visual quality but lack sufficiently reliable content information to maintain temporal consistency. In the inference phase, HiAR alters the standard generation sequence to perform hierarchical denoising, while our proposed method maintains the chunk-wise autoregressive pipeline as a plug-and-play module.

In this work, we identify the reuse of the same clean KV caches throughout the entire denoising trajectory of the current chunk as an important source of model drift. Building on this observation,

we propose a training-free technique that provides the current chunk with step-aligned KV caches, thereby mitigating model drift during long video generation.

3 METHOD

We first briefly review multi-step video diffusion models and few-step autoregressive video diffusion models (Sec. 3.1). We then examine how interventions on clean KV caches contribute to the observed drift. (Sec. 3.2). Based on this analysis, we propose a matched KV caching strategy to mitigate model drift during autoregressive rollouts (Sec. 3.3). Finally, we provide an overview of our proposed method, **Step Forcing** (Sec. 3.4).

3.1 PRELIMINARIES: AUTOREGRESSIVE VIDEO DIFFUSION MODELS

Video Diffusion Models. Video diffusion models (Yang et al., 2024; Wang et al., 2025) generate videos through a multi-step denoising process (Ho et al., 2020; Song et al., 2020) that progressively reverses a predefined forward noising process. Given a clean sample $\mathbf{x}_0$, the corresponding noisy sample at timestep t can be expressed as

$$\mathbf{x}_t = \alpha_t \mathbf{x}_0 + \sigma_t \boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{1}$$

where α_t and σ_t denote the signal and noise scales, respectively. They are specified by a predefined noise schedule and satisfy $\alpha_t = 1 - \sigma_t$. During inference, the model progressively transforms a Gaussian noise sample into a clean sample along a denoising trajectory, with the noise scale σ_t gradually decreasing over the denoising steps.

Autoregressive Video Diffusion Models. Autoregressive video diffusion models (Yin et al., 2025; Huang et al., 2026) combine autoregressive decomposition with denoising diffusion models, generating video frames or chunks sequentially while modeling each conditional distribution with a diffusion process. These models are typically obtained by distilling a multi-step video diffusion model into a few-step counterpart. The resulting few-step model follows a discrete denoising trajectory derived from that of the original multi-step model, along which the noise scale likewise decreases over denoising steps. Specifically, let a video consist of N frames, which are partitioned into L chunks $\{\mathbf{x}_0^i\}_{i=1}^L$, where $i \in \{1, \dots, L\}$ denotes the chunk index. In the special case where each chunk contains a single frame, we have $L = N$. For a given chunk i ($i > 1$), its denoising trajectory is denoted by $\{\mathbf{x}_{t_j}^i\}_{j=1}^J$, where J is the total number of denoising steps, and the noise scale satisfy

$$\sigma_{t_1} \geq \sigma_{t_2} \geq \cdots \geq \sigma_{t_J}. \tag{2}$$

At each denoising timestep t_j, the diffusion denoising process transforms an intermediate noisy chunk $\mathbf{x}_{t_j}^i$ toward its clean counterpart conditioned on the previously generated clean chunks $\mathbf{x}_0^{<i}$. It then adds noise to move the denoised result to the next timestep t_{j-1}, yielding $\mathbf{x}_{t_{j-1}}^i$. Following consistency-based few-step diffusion models (Song et al., 2023; Yin et al., 2024a), the sampling process alternates between denoising and noise addition steps to improve sample fidelity. An autoregressive video diffusion model factorizes the conditional video distribution as $p(\mathbf{x}_0^{1:L} \mid c) = \prod_{i=1}^L p(\mathbf{x}_0^i \mid \mathbf{x}_0^{<i}, c)$, where c denotes the text prompt and $\mathbf{x}_0^{<i}$ denotes previously generated chunks. Each chunk $\mathbf{x}_0^i$ is then generated by a diffusion denoising process conditioned on both the prompt and the preceding clean chunks.

Since each chunk is conditioned on the preceding clean chunks, naively recomputing their keys and values at every denoising step would introduce substantial redundant computation (Brown et al., 2020). Autoregressive video diffusion models leverage KV caching mechanism to improve efficiency, reusing the keys and values of previous clean chunks and avoiding redundant recomputation (see Fig. 4a). Specifically, for a given denoising timestep t_j in chunk i ($i > 1$), the keys and values used for attention are formed by concatenating the clean KV caches $\mathbf{K}_0^{<i}, \mathbf{V}_0^{<i} \in \mathbb{R}^{S \times d}$ from previously chunks $\mathbf{x}_0^{<i}$ and current KV states $\mathbf{K}_{t_j}^i, \mathbf{V}_{t_j}^i \in \mathbb{R}^{s \times d}$ from chunk $\mathbf{x}_{t_j}^i$:

$$\mathbf{K} = \texttt{Concat}\left(\mathbf{K}_0^{<i}, \mathbf{K}_{t_j}^i\right), \quad \mathbf{V} = \texttt{Concat}\left(\mathbf{V}_0^{<i}, \mathbf{V}_{t_j}^i\right), \tag{3}$$

where $\mathbf{K}, \mathbf{V} \in \mathbb{R}^{(S+s) \times d}$, s denotes the sequence length of the current chunk, corresponding to the number of tokens in its current KV states, and $S = (i-1) \times s$ denotes the total number of cached tokens from the preceding $i-1$ clean chunks. We omit the attention-head dimension for brevity.



Figure 2: **Effect of historical KV cache interventions on CausVid.** (a) Outputs at the first and final steps for selected chunks in the vanilla model. (b) Historical KV reuse is disabled at the first step when generating chunk 2. (c) Historical KV reuse is disabled at an intermediate step of each chunk, with the results shown for chunk 7. The interventions in (b) and (c) mitigate model drift.

3.2 Diagnosing Drift through Historical KV Conditioning

Recent studies have shown that autoregressive video diffusion models are prone to model drift, as exemplified by the CausVid (Yin et al., 2025) outputs shown in Fig. 2a (2nd row). Autoregressive video diffusion models typically combine DMD distillation (Yin et al., 2024b;a) with KV caching (Brown et al., 2020) to enable long video generations. Since few-step diffusion models distilled with DMD generally do not exhibit noticeable model drift (Yin et al., 2024b;a; Ge et al., 2025; Shao et al., 2025; Nie et al., 2026), the drift (e.g., over-saturation, semantic drift, color drift) observed in autoregressive video generation is unlikely to originate from the few-step model itself. Intuitively, it mainly stems from the repeated reuse of mismatched KV caches across chunks.

To validate this intuition, we investigate model drift at the denoising step, since clean KV caches is applied throughout the denoising process of each chunk. Specifically, we compare each subsequent chunk against the first chunk at the corresponding denoising step. As shown in Fig. 2a, each row corresponds to a denoising step and exhibits progressive drift in the generated chunks over time. Moreover, in the first denoising step (Fig. 2a (1st row)), the second chunk already exhibits noticeable quality drift compared with the first (1st-2nd columns), and the discrepancy becomes more severe in later chunks relative to the last step outputs (3rd column). Also, we use FVD (Unterthiner et al., 2018) and MS-SWD (He et al., 2024) to measure distributional degradation and color discrepancy, respectively. The quantitative evaluations exhibit a consistent trend. Specifically, for the second chunk (Fig. 2a (2nd column)), the first denoising step yields substantially higher FVD and MS-SWD values than the last denoising step (2002.96 vs. 620.65 and 4.45 vs. 2.61, respectively). Meanwhile, at both denoising steps, FVD and MS-SWD consistently increase for later chunks. This analysis indicates that model drift emerges as early as the first denoising step, is more pronounced at that step than at subsequent steps, and progressively accumulates as more chunks are generated.

The aforementioned analysis shows that model drift is already present at the first denoising step. Therefore, we examine how KV caching affects model drift at this step. Specifically, when generating the second chunk, we do not reuse the KV cache from the first chunk during the first denoising step. Instead, we use only the keys and values computed for the current chunk at this step, i.e., $\mathbf{K} = \mathbf{K}^2_{j_1}$ and $\mathbf{V} = \mathbf{V}^2_{j_1}$. The results are shown in Fig. 2b. Compared with the vanilla second chunk (Fig. 2a (1st row, 2nd column)), the resulting second chunk (Fig. 2b (1st row)) is more closely aligned with the first chunk (Fig. 2a (1st row, 1st column)), leading to reduced model drift and better FVD and MS-SWD scores. Model drift at the final denoising step is also alleviated (Fig. 2b (2nd row)). Furthermore, disabling clean KV cache reuse at an intermediate denoising step in each chunk reduces model drift in the subsequent chunk (Fig. 2c).

In conclusion, clean KV caches contribute to model drift, stemming from their mismatch with the representations required at each denoising step of current chunk.

3.3 Matched KV Caches

Diffusion Step KV Caches. To avoid the model drift associated with using the mismatched clean KV caches $\mathbf{K}_0^{<i}, \mathbf{V}_0^{<i}$ at each diffusion denoising step (Eq. (3)), we explore the use of noisy KV



Figure 3: Videos generated by the original autoregressive diffusion models (1st and 3rd rows) and by their counterparts using Step KV (2nd and 4th rows).

caches $\mathbf{K}_{t_j}^{<i}, \mathbf{V}_{t_j}^{<i}$ from the corresponding denoising step instead. Under this setting, the keys and values used for attention in the i-th ($i > 1$) chunk are reformulated as:

$$\hat{\mathbf{K}} = \texttt{Concat}\left(\mathbf{K}_{t_j}^{<i}, \mathbf{K}_{t_j}^{i}\right), \quad \hat{\mathbf{V}} = \texttt{Concat}\left(\mathbf{V}_{t_j}^{<i}, \mathbf{V}_{t_j}^{i}\right). \tag{4}$$

We refer to this cache formulation as *Step KV* (see Fig. 4b). We observe that *Step KV* effectively alleviates model drift when generating long videos. For example, given the prompt "Aerial tracking camera shot. A white semi-truck drives on a highway", the generated video exhibits progressive drift as the autoregressive rollouts (Fig. 3 (1st and 3rd rows)). In contrast, with *Step KV*, the generated video better preserves visual quality during long-horizon rollouts (Fig. 3 (2nd and 4th rows)).

Table 1: Effect of *Step KV* on visual quality and temporal consistency.

Methods	Color	Object Class	Subject Cons.	Motion Smooth.
Self Forcing	84.78	93.16	**97.40**	**98.39**
Step KV	**85.45**	**93.48**	97.01	97.01

The results demonstrate that Step KV better aligns cached representations with the requirements of each denoising step, improving visual quality while introducing a trade-off in temporal consistency. As shown in Tab. 1, replacing clean KV caches with noisy ones improves Color and Object Class scores but reduces Subject Consistency and Motion Smoothness. This suggests that noisy KV caches better support visual quality but provide less reliable historical content for maintaining consistency across chunks. This effect is also visible in Fig. 3: with Step KV, the truck's appearance gradually changes in CausVid (Yin et al., 2025) (Fig. 3 (2nd row)), while its structure progressively splits into two connected sections in Self Forcing (Huang et al., 2026) (Fig. 3 (4th row)). These observations motivate retaining clean KV information as stable content anchors during autoregressive rollout.

Anchor Tokens Selection. Based on the above analysis, we assume that the noisy KV caches used in Step KV provide representations that are better aligned with the generation visual quality required by the diffusion denoising steps, but lack sufficient reliable content information to maintain temporal consistency during autoregressive rollout. Meanwhile, clean KV caches play a crucial role in maintaining temporal consistency, although they can cause shifts in color and saturation (Fig. 3, 1st and 3rd rows). Therefore, clean KV caches should be retained to preserve the original model's temporal consistency.

However, retaining the complete clean KV caches for all historical chunks would introduce substantial memory and computational overhead (see Tab. 5 for detail). Inspired by acceleration strategies that use a small set of tokens to approximate the full token set (Wang et al., 2024; Ji et al., 2026; Guo et al., 2025), we retain only a compact subset of clean KV tokens, termed *anchor tokens*, to provide stable references to previously generated content. Specifically, given the clean $\mathbf{K}_0^m$ of chunk m, where $0 \le m \le i-1$, we sample s_{t_j} most important tokens in step t_j following a sampling strategy inspired by in (Frieze et al., 2004) under the low-rank assumption (Saxena et al., 2024). The ℓ-th token is sampled with probability P_ℓ satisfying

$$P_\ell \ge c \, \|(\mathbf{K}_0^m)^{(\ell)}\|_2^2 \Big/ \|\mathbf{K}_0^m\|_F^2, \tag{5}$$

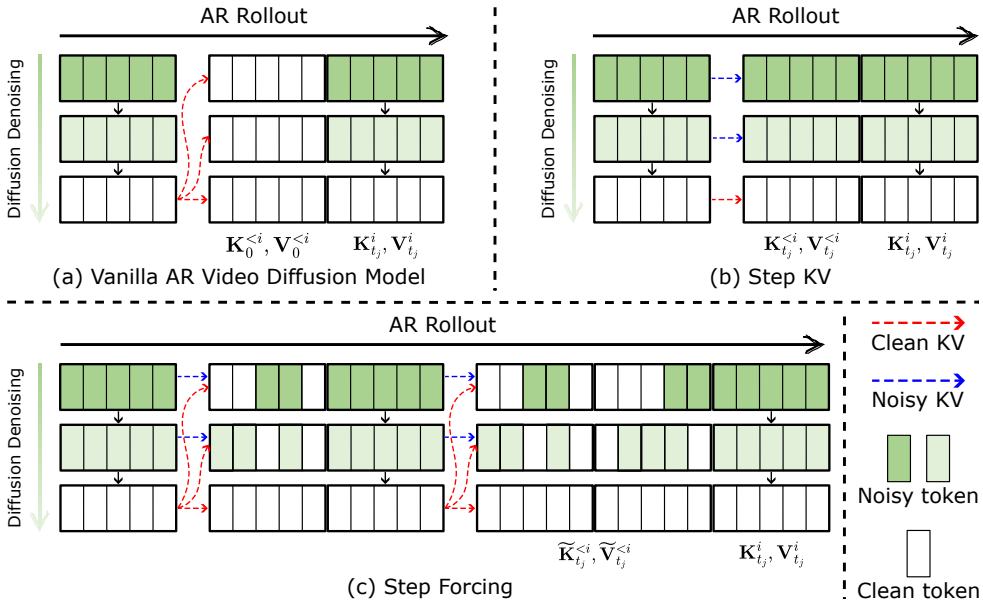


Figure 4: (a) Vanilla Autoregressive video diffusion models. (b) *Step KV* reuses noisy KV caches $\mathbf{K}_{t_j}^{<i}$ and $\mathbf{V}_{t_j}^{<i}$ extracted from previous chunks at the corresponding denoising timestep. (c) Step Forcing uses step-matched KV tokens.

where $0 < c \leq 1$. The indices of the sampled tokens are collected in $\mathcal{I}_{t_j}^m$, with $|\mathcal{I}_{t_j}^m| = s_{t_j}$, where $|\cdot|$ denotes the number of elements in a set. We then construct a modified key cache by replacing the key tokens at the selected anchor positions with their clean counterparts:

$$\left(\widetilde{\mathbf{K}}_{t_j}^m\right)^{(\ell)} := \begin{cases} (\mathbf{K}_0^m)^{(\ell)}, & \ell \in \mathcal{I}_{t_j}^m, \\ \left(\mathbf{K}_{t_j}^m\right)^{(\ell)}, & \ell \notin \mathcal{I}_{t_j}^m. \end{cases} \tag{6}$$

Here, $(\cdot)^{(\ell)}$ denotes the ℓ-th token, or equivalently the ℓ-th row, of a key cache. Thus, in $\widetilde{\mathbf{K}}_{t_j}^m$, the tokens indexed by $\mathcal{I}_{t_j}^m$ are taken from the clean key cache $\mathbf{K}_0^m$, while the remaining tokens are left unchanged in the noisy key cache $\mathbf{K}_{t_j}^m$. By concatenating the modified key caches of all preceding chunks, we obtain $\widetilde{\mathbf{K}}_{t_j}^{<i} = \texttt{Concat}\left(\widetilde{\mathbf{K}}_{t_j}^0, \ldots, \widetilde{\mathbf{K}}_{t_j}^{i-1}\right)$. Likewise, the modified value cache $\widetilde{\mathbf{V}}_{t_j}^{<i}$ is constructed by applying the same replacement rule using $\mathcal{I}_{t_j}^m$. The complete KV cache used for chunk i are then constructed as:

$$\widetilde{\mathbf{K}} = \texttt{Concat}\left(\widetilde{\mathbf{K}}_{t_j}^{<i}, \mathbf{K}_{t_j}^i\right), \quad \widetilde{\mathbf{V}} = \texttt{Concat}\left(\widetilde{\mathbf{V}}_{t_j}^{<i}, \mathbf{V}_{t_j}^i\right). \tag{7}$$

In this way, the noisy KV caches facilitates step-adaptive denoising, whereas the clean anchor tokens preserve cross-chunk appearance, structure, and motion consistency throughout autoregressive rollout.

More Clean KV Tokens at Early Denoising Steps. The explicit visual information of the noisy KV caches reused by Step KV varies substantially along the denoising trajectory. We characterize this variation using the signal-to-noise ratio (SNR) , which at denoising step t_j is defined as $\mathrm{SNR}(t_j) = \left(1 - \sigma_{t_j}\right)^2 \Big/ \sigma_{t_j}^2 \in [0, +\infty]$. As the noise scale decreases along the denoising trajectory (see Eq. (2)), the corresponding SNR increases monotonically: $\mathrm{SNR}(t_1) \leq \mathrm{SNR}(t_2) \leq \cdots \leq \mathrm{SNR}(t_J)$. Thus, at early denoising steps, the noisy KV caches used by Step KV contain less explicit visual information and therefore require more clean anchor tokens, whereas fewer clean anchor tokens are sufficient at later steps.

To ensure sufficient visual information at each step along the denoising trajectory, we dynamically determine the number of sampled clean tokens s_{t_j}. To this end, we define the noise ratio at timestep t_j as $\rho_{t_j} = 1 \Big/ [1 + \mathrm{SNR}(t_j)] = \sigma_{t_j}^2 \Big/ [\left(1 - \sigma_{t_j}\right)^2 + \sigma_{t_j}^2]$, The number of sampled clean tokens s_{t_j} is then determined by

Table 2: Quantitative comparison with relevant autoregressive diffusion models on the VBench (Huang et al., 2024) benchmark for 5s and VBench-Long (Huang et al., 2025) benchmark for 30s video generation.

Method	Training Free	Subject Cons.	Background Cons.	Aesthetic Quality	Imaging Quality	Overall Cons.	Motion Smooth.	Total↑	Quality↑	Semantic↑
5 seconds										
Wan2.1	✗	96.34	97.29	62.43	66.51	27.07	97.44	83.96	84.92	80.10
CausVid	✗	95.10	**96.56**	62.67	65.32	25.01	95.66	78.43	80.13	71.67
+Ours	✓	**96.30**	95.45	**65.93**	**68.74**	**26.14**	**97.81**	**83.63**	**84.59**	**79.81**
30 seconds										
Rolling Forcing	✗	98.25	97.30	63.92	70.82	26.49	98.65	82.83	83.57	79.86
CausVid	✗	95.10	96.58	62.67	65.32	25.01	95.66	79.05	80.34	73.88
+Ours	✓	**98.08**	**96.98**	**64.46**	**67.18**	**25.32**	**97.85**	**82.79**	**84.33**	**76.62**
Self Forcing	✗	**98.58**	**97.06**	61.80	**68.77**	**26.43**	98.56	80.84	81.94	76.42
+Ours	✓	97.42	96.29	**62.62**	68.26	25.82	**98.62**	**81.94**	**83.04**	**77.51**
Causal Forcing	✗	96.73	**96.12**	56.26	67.56	21.72	97.81	78.62	82.64	62.54
+Ours	✓	**97.21**	96.05	**59.60**	**68.61**	**23.24**	**97.41**	**80.66**	**84.32**	**66.02**
Deep Forcing	✓	96.87	**95.92**	**63.96**	68.25	26.38	**98.21**	83.03	84.31	77.90
+Ours	✓	**96.92**	95.88	63.88	**68.91**	**26.39**	98.19	**83.12**	**84.40**	78.03

$$s_{t_j} = \left[s_{\min} + (s_{\max} - s_{\min}) \left(\frac{\rho_{t_j} - \rho_{t_J}}{\rho_{t_1} - \rho_{t_J}} \right)^{\gamma} \right], \tag{8}$$

where $\gamma > 0$ controls the decay rate. Consequently, the number of sampled clean tokens satisfies $s_{t_1} = s_{\max} \geq s_{t_2} \geq \cdots \geq s_{t_J} = s_{\min}$. The corresponding maximum and minimum clean token ratios are defined as $r_{\max} = s_{\max} \big/ s$ and $r_{\min} = s_{\min} \big/ s$, respectively.

3.4 Step Forcing

The overall framework of the proposed **Step Forcing** is illustrated in Fig. 4c. **Step Forcing** departs from the vanilla autoregressive video diffusion model, which relies on clean KV caches from previous chunks for rollout (Fig. 4a). At each denoising step of the current chunk, it reuses noisy KV caches extracted at the corresponding step from previous chunks while retaining anchor tokens from their clean KV caches. Therefore, while vanilla models store only clean KV caches, **Step Forcing** stores KV caches for every denoising step, increasing memory usage.

To address this memory overhead, we compress the KV caches to maintain a memory footprint comparable to that of vanilla models. Specifically, following the sampling strategy in Eq. (5), we select $\lceil \beta s \rceil$ tokens from the cache $\mathbf{K}_{t_j}^m$ at each denoising step, where s denotes the number of tokens before compression and $\beta \in [1/J, 1]$ controls the fraction of retained tokens. For notational simplicity, we subsequently reuse s to denote the number of tokens after compression. Our full algorithm is presented in Algorithm 1.

4 Experiments

Experimental Setup. Our proposed method, **Step Forcing**, can be integrated into autoregressive video diffusion models to generate videos of varying lengths with reduced model drift. We evaluate it on CausVid (Yin et al., 2025) for 5-second video generation. We also apply it to three training-based models (CausVid (Yin et al., 2025), Self Forcing (Huang et al., 2026), and Causal Forcing (Zhu et al., 2026)) and the training-free method Deep Forcing (Yi et al., 2025) for 30-second video generation. Following the evaluation settings of the baselines, we evaluate our method on VBench (Huang et al., 2024) for 5-second video generation and on VBench-Long (Huang et al., 2025) for 20- and 30-second video generation.

We set the maximum and minimum clean token ratios, $r_{\max}$ and $r_{\min}$, to 0.8 and 0.4 for CausVid, to 1.0 and 0.2 Causal Forcing, and to 1.0 and 0.8, respectively, for Self Forcing and Deep Forcing. We

Table 3: Ablation study of r_{max} and r_{min} with γ=4.

r_{max}	r_{min}	Total↑	Quality↑	Semantic↑
0.9	0.3	82.65	84.26	76.23
0.8	0.4	82.93	84.43	76.69
0.8	0.2	82.91	84.41	76.90
0.6	0.2	82.54	83.94	76.91
0.0	0.0	82.25	83.32	77.97

Table 4: Ablation study of γ.

γ	Total↑	Quality↑	Semantic↑
2	82.91	84.41	76.90
4	82.93	84.43	76.69
8	82.97	84.40	77.25
16	82.54	83.94	76.91
32	82.47	83.85	76.96

Table 5: Ablation study of β. ΔMem.: Memory increase (%).

β	ΔMem.↓	Total↑	Quality↑	Semantic↑
1	57.8%	82.97	84.40	77.25
0.8	40.47%	82.91	84.30	77.34
0.5	14.45%	82.79	84.33	76.62
0.4	5.78%	82.67	84.22	76.48
1/3	0%	82.71	83.50	79.54

set the decay rate γ to 8 for all models. For the fraction β, we use 0.5 for CausVid, 0.6 for Causal Forcing and Self Forcing, and 0.7 for Deep Forcing.

Main Results. Tab. 2 presents the evaluation results on the VBench (Huang et al., 2024) and VBench-Long (Huang et al., 2025) benchmarks. Since CausVid exhibits color drift and oversaturation when generating 5-second videos, we first evaluate its performance with **Step Forcing** (Ours) in this setting. As shown in Tab. 2 (*5 seconds*), **Step Forcing** improves CausVid's total, quality, and semantic scores, bringing them close to those of Wan2.1. Fig. 5 shows the same trend. To evaluate the effectiveness of **Step Forcing** in long video generation, we integrate it into CausVid to generate 30-second videos. As shown in Tab. 2 (*30 seconds*), **Step Forcing** substantially improves the score of vanilla CausVid to 82.79, bringing it close to that of the state-of-the-art method Rolling Forcing (82.83). Integrating **Step Forcing** with the other two training-based models, Self Forcing and Causal Forcing, improves their total scores in VBench-Long scores by 1.10 and 2.04 points, respectively. We also observe performance gains when integrating it with the training-free method Deep Forcing. The qualitative results in Fig. 6 further support the improvements, showing consistent gains in visual quality and semantic over the vanilla models (See Sec. B.). For long-horizon generation (e.g., 120s), **Step Forcing** further improves the baseline and reduces drift (Fig. 1).



Figure 5: Qualitative comparison with Wan2.1 and CausVid for 5-second video generation. Ours mitigates model drift in CausVid and achieves visual quality comparable to that of Wan2.1.

Ablation Study. We conduct an ablation study on $r_{\max}$, $r_{\min}$, γ, and β in the proposed method using CausVid for 30-second video generation and report the quantitative results in Tabs. 3 to 5. As shown in Tab. 3, we compare different configurations of $r_{\max}$ and $r_{\min}$ with γ fixed at 4 and find that setting $r_{\max} = 0.8$ and $r_{\min} = 0.4$ achieves a better balance between quality and semantic. For completeness, we also evaluate the setting without replacing noisy KV caches with clean ones and observe a drop in the quality score (Tab. 3 (last row)).

We then fixed $r_{\max}$ and $r_{\min}$ and conducted an ablation study over different values of γ. As shown in Tab. 4, $\gamma = 8$ achieves a favorable trade-off between quality and semantic performance, resulting in a better total score. Finally, since reusing noisy KV caches at each step increases memory usage, we compress the noisy KV caches using the parameter β. Tab. 5 presents the impact of different β values on memory usage and performance. We set $\beta = 0.5$, which significantly reduces memory usage while maintaining an acceptable level of performance.

5 Conclusion

In this work, we propose a training-free approach to mitigate model drift in autoregressive video diffusion models during long video generation. We identify a denoising-step mismatch, particularly in the KV caches of most existing chunk-wise autoregressive diffusion methods. Leveraging this insight, we propose **Step Forcing**, a plug-and-play, training-free method that provides step-matched context to pre-trained autoregressive video diffusion models without modifying their training or generation schedules. **Limitation:** Our training-free approach retains a subset of clean KV tokens to maintain temporal consistency. However, reusing these tokens may still contribute to gradual model drift when generating longer videos. Future work could explore fine-tuning with step-matched context to further mitigate drift over extended generation horizons.

AI USE STATEMENT

In this work, generative AI tools were used solely to polish sentences in the manuscript to improve readability. We manually reviewed all AI-assisted text to ensure accuracy and preserve the original scientific meaning. We take full responsibility for the final content of this work.

REFERENCES

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Tri Cao, Hung Nguyen, Phong Nguyen, and Khoi Nguyen. Recency forcing: Bridging the long-horizon gap in autoregressive video generation. *arXiv preprint arXiv:2609.19729*, 2026.

Jiayu Chen, Junbei Tang, Wenbiao Zhao, Maoliang Li, Jiayi Luo, Zihao Zheng, Jiawei Yang, Guojie Luo, and Xiang Chen. Pyramid forcing: Head-aware pyramid kv cache policy for high-quality long video generation. *arXiv preprint arXiv:2605.13111*, 2026a.

Shuo Chen, Cong Wei, Sun Sun, Ping Nie, Kai Zhou, Ge Zhang, Ming-Hsuan Yang, and Wenhu Chen. Context forcing: Consistent autoregressive video generation with long context, 2026b. URL https://arxiv.org/abs/2602.06028.

Alan Frieze, Ravi Kannan, and Santosh Vempala. Fast monte-carlo algorithms for finding low-rank approximations. *Journal of the ACM (JACM)*, 51(6):1025–1041, 2004.

Xingtong Ge, Xin Zhang, Tongda Xu, Yi Zhang, Xinjie Zhang, Yan Wang, and Jun Zhang. Senseflow: Scaling distribution matching for flow-based text-to-image distillation. *arXiv preprint arXiv:2506.00523*, 2025.

Hang Guo, Yawei Li, Taolin Zhang, Jiangshan Wang, Tao Dai, Shu-Tao Xia, and Luca Benini. Fastvar: Linear visual autoregressive modeling via cached token pruning. *arXiv preprint arXiv:2503.23367*, 2025.

Jiaqi He, Zhihua Wang, Leon Wang, Tsein-I Liu, Yuming Fang, Qilin Sun, and Kede Ma. Multiscale sliced wasserstein distances as perceptual color difference measures. In *European Conference on Computer Vision*, pp. 425–442. Springer, 2024.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. *Advances in Neural Information Processing Systems*, 38:167283–167308, 2026.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. VBench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Ziqi Huang, Fan Zhang, Xiaojie Xu, Yinan He, Jiashuo Yu, Ziyue Dong, Qianli Ma, Nattapol Chanpaisit, Chenyang Si, Yuming Jiang, Yaohui Wang, Xinyuan Chen, Ying-Cong Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. VBench++: Comprehensive and versatile benchmark suite for video generative models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025. doi: 10.1109/TPAMI.2025.3633890.

Yicheng Ji, Zhizhou Zhong, Jun Zhang, Qin Yang, XiTai Jin, Ying Qin, Wenhan Luo, Shuiyang Mao, Wei Liu, and Huan Li. Forcing-kv: Hybrid kv cache compression for efficient autoregressive video diffusion models. *arXiv preprint arXiv:2605.09681*, 2026.

Jia Li, Xiaomeng Fu, Xurui Peng, Weifeng Chen, Youwei Zheng, Tianyu Zhao, Jiexi Wang, Fangmin Chen, Xing Wang, and Hayden Kwok-Hay So. Train short, inference long: Training-free horizon extension for autoregressive video generation. *arXiv preprint arXiv:2602.14027*, 2026.

Jinxiu Liu, Xuanming Liu, Kangfu Mei, Yandong Wen, Ming-Hsuan Yang, and Weiyang Liu. Streaming autoregressive video generation via diagonal distillation. In *ICLR*, 2026.

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.

Jiayi Luo, Qiyan Liu, Tengyang Wang, JunHao Liu, Jiayu Chen, Cong Wang, Hanxin Zhu, Chen Gao, Xiaobin Hu, Qingyun Sun, et al. Future forcing: Future-aware training-free kv cache policy for autoregressive video generation. *arXiv preprint arXiv:2605.30083*, 2026.

Chengtao Lv, Yumeng Shi, Yushi Huang, Ruihao Gong, Shen Ren, and Wenya Wang. Light forcing: Accelerating autoregressive video diffusion via sparse attention. *arXiv preprint arXiv:2602.04789*, 2026.

Xiaofeng Mao, Shaohao Rui, Kaining Ying, Bo Zheng, Chuanhao Li, Mingmin Chi, and Kaipeng Zhang. Packforcing: Short video training suffices for long video sampling and long context inference, 2026. URL https://arxiv.org/abs/2603.25730.

Yu Meng, Xiangyang Luo, Letian Li, Wenyuan Jiang, Chen Gao, Xinlei Chen, Yong Li, and Xiao-Ping Zhang. Tethercache: Stabilizing autoregressive long-form video generation with gated recall and trusted alignment. *arXiv preprint arXiv:2606.13035*, 2026.

Weili Nie, Julius Berner, Nanye Ma, Chao Liu, Saining Xie, and Arash Vahdat. Transition matching distillation for fast video generation. *arXiv preprint arXiv:2601.09881*, 2026.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4172–4182. IEEE, 2023.

Dvir Samuel, Issar Tzachor, Matan Levy, Michael Green, Gal Chechik, and Rami Ben-Ari. Fast autoregressive video diffusion and world models with temporal cache compression and sparse attention. *arXiv preprint arXiv:2602.01801*, 2026.

Utkarsh Saxena, Gobinda Saha, Sakshi Choudhary, and Kaushik Roy. Eigen attention: Attention in low-rank space for kv cache compression. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 15332–15344, 2024.

Shitong Shao, Hongwei Yi, Hanzhong Guo, Tian Ye, Daquan Zhou, Michael Lingelbach, Zhiqiang Xu, and Zeke Xie. Magicdistillation: Weak-to-strong video distillation for large-scale few-step synthesis. *arXiv preprint arXiv:2503.13319*, 2025.

Ariel Shaulov, Eitan Shaar, Amit Edenzon, and Lior Wolf. Tokentrim: Inference-time token pruning for autoregressive long video generation. *arXiv preprint arXiv:2602.00268*, 2026.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 32211–32252, 2023.

Thomas Unterthiner, Sjoerd Van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018.

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

Hongjie Wang, Difan Liu, Yan Kang, Yijun Li, Zhe Lin, Niraj K Jha, and Yuchen Liu. Attention-driven training-free efficiency enhancement of diffusion models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16080–16089. IEEE, 2024.

Mingqiang Wu, Weilun Feng, Zhefeng Zhang, Haotong Qin, Yuqi Li, Guoxin Fan, Xiaokun Liu, Zhulin An, Libo Huang, Yongjun Xu, et al. Echo-forcing: A scene memory framework for interactive long video generation. *arXiv preprint arXiv:2605.16003*, 2026.

Xunzhi Xiang, Zixuan Duan, Guiyu Zhang, Haiyu Zhang, Zhe Gao, Junta Wu, Shaofeng Zhang, Tengfei Wang, Qi Fan, and Chunchao Guo. Pathwise test-time correction for autoregressive long video generation. *arXiv preprint arXiv:2602.05871*, 2026.

Boxun Xu, Yuming Du, Zichang Liu, Siyu Yang, Ziyang Jiang, Siqi Yan, Rajasi Saha, Albert Pumarola, Wenchen Wang, and Peng Li. Sparse forcing: Native trainable sparse attention for real-time autoregressive diffusion video generation. *arXiv preprint arXiv:2604.21221*, 2026.

Bowen Xue, Brandon Y Feng, Chenguo Lin, Yuchen Lin, Yujia Zeng, Lvmin Zhang, Maneesh Agrawala, Honglei Yan, and Panwang Pan. Ring forcing: Towards precise long-term memory for autoregressive video diffusion. In *European Conference on Computer Vision*, pp. 225–242. Springer, 2026.

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. In *ICLR*, 2026.

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 William T Freeman. Improved distribution matching distillation for fast image synthesis. In *NeurIPS*, 2024a.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Frédo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *CVPR*, 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 *CVPR*, 2025.

Shenghai Yuan, Yuanyang Yin, Zongjian Li, Xinwei Huang, Xiao Yang, and Li Yuan. Helios: Real real-time long video generation model. *arXiv preprint arXiv:2603.04379*, 2026.

Zengqun Zhao, Yanzuo Lu, Ziquan Liu, Jifei Song, Jiankang Deng, and Ioannis Patras. Relax forcing: Relaxed KV-memory for consistent long video generation. In *British Machine Vision Conference (BMVC)*, 2026.

Hongzhou Zhu, Min Zhao, Guande He, Hang Su, Chongxuan Li, and Jun Zhu. Causal forcing: Autoregressive diffusion distillation done right for high-quality real-time interactive video generation. *arXiv preprint arXiv:2602.02214*, 2026.

Kai Zou, Dian Zheng, Hongbo Liu, Tiankai Hang, Bin Liu, and Nenghai Yu. Hiar: Efficient autoregressive long video generation via hierarchical denoising. *arXiv preprint arXiv:2603.08703*, 2026.

APPENDIX

A ALGORITHM DETAIL OF **STEP FORCING**

Algorithm 1: : **Step Forcing**

Input : The number of the chunk L. The denoising step J. The autoregressive video diffusion model ϕ[a].

Output: The generated videos $\{x_0^i\}_{i=1}^L$.

for $i = 1, \cdots, L$ **do**
  $x_{t_1}^i \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$;
  if $i = 1$ **then**
    $\mathbf{K}_0^{<i}, \mathbf{V}_0^{<i} \leftarrow [], []$
    for $j = 1, \cdots, J$ **do**
      $\widetilde{\mathbf{K}}_{t_j}^{<i}, \widetilde{\mathbf{V}}_{t_j}^{<i} \leftarrow [], []$
      $x_{t_{j+1}}^i, \mathbf{K}_{t_j}^i, \mathbf{V}_{t_j}^i \xleftarrow{\phi} x_{t_j}^i$
      $\mathbf{K}_{t_j}^{<i}.\texttt{append}(\mathbf{K}_{t_j}^i), \mathbf{V}_{t_j}^{<i}.\texttt{append}(\mathbf{V}_{t_j}^i)$
    end
    $x_0^i \leftarrow x_{t_{J+1}}^i$
    $\mathbf{K}_0^i, \mathbf{V}_0^i \xleftarrow{\phi} x_{t_0}^i$
    $\mathbf{K}_0^{<i}.\texttt{append}(\mathbf{K}_0^i), \mathbf{V}_0^{<i}.\texttt{append}(\mathbf{V}_0^i)$
  else
    for $j = 1, \cdots, J$ **do**
      $\mathbf{K}_{t_j}^i, \mathbf{V}_{t_j}^i \xleftarrow{\phi} x_{t_j}^i$
      $\widetilde{\mathbf{K}} = \texttt{Concat}\left(\widetilde{\mathbf{K}}_{t_j}^{<i}, \mathbf{K}_{t_j}^i\right), \widetilde{\mathbf{V}} = \texttt{Concat}\left(\widetilde{\mathbf{V}}_{t_j}^{<i}, \mathbf{V}_{t_j}^i\right), ;$ // Eq. (7)
      $x_{t_{j+1}}^i \leftarrow x_{t_j}^i, \widetilde{\mathbf{K}}, \widetilde{\mathbf{V}}$
      // Anchor Tokens Selection
      The chosen indices $\mathcal{I}_{t_j}^i$, based on $\mathbf{K}_0^i$; // Eq. (5)
      $\left(\widetilde{\mathbf{K}}_{t_j}^i\right)^{(\ell)} := \begin{cases} \left(\mathbf{K}_0^i\right)^{(\ell)}, & \ell \in \mathcal{I}_{t_j}^i, \\ \left(\mathbf{K}_{t_j}^i\right)^{(\ell)}, & \ell \notin \mathcal{I}_{t_j}^i. \end{cases}$; // Eq. (6)
      $\left(\widetilde{\mathbf{V}}_{t_j}^i\right)^{(\ell)} := \begin{cases} \left(\mathbf{V}_0^i\right)^{(\ell)}, & \ell \in \mathcal{I}_{t_j}^i, \\ \left(\mathbf{V}_{t_j}^i\right)^{(\ell)}, & \ell \notin \mathcal{I}_{t_j}^i. \end{cases}$; // Eq. (6)
      $\widetilde{\mathbf{K}}_{t_j}^{<i}.\texttt{append}(\widetilde{\mathbf{K}}_{t_j}^i), \widetilde{\mathbf{V}}_{t_j}^{<i}.\texttt{append}(\widetilde{\mathbf{V}}_{t_j}^i)$
    end
    $x_0^i \leftarrow x_{t_{J+1}}^i$
  end
end
Return The generated videos $\{x_0^i\}_{i=1}^L$.

[a] We omit the noise addition step during inference for brevity.

B ADDITIONAL EXPERIMENTALS

Fig. 6 shows qualitative results of integrating **Step Forcing** into existing autoregressive video diffusion baselines. **Step Forcing** improves long-video generation by mitigating model drift while preserving generation fidelity.

When the baseline natively supports long-horizon video generation, such as 120-second videos, incorporating **Step Forcing** further improves generation quality and reduces drift. Taking Deep

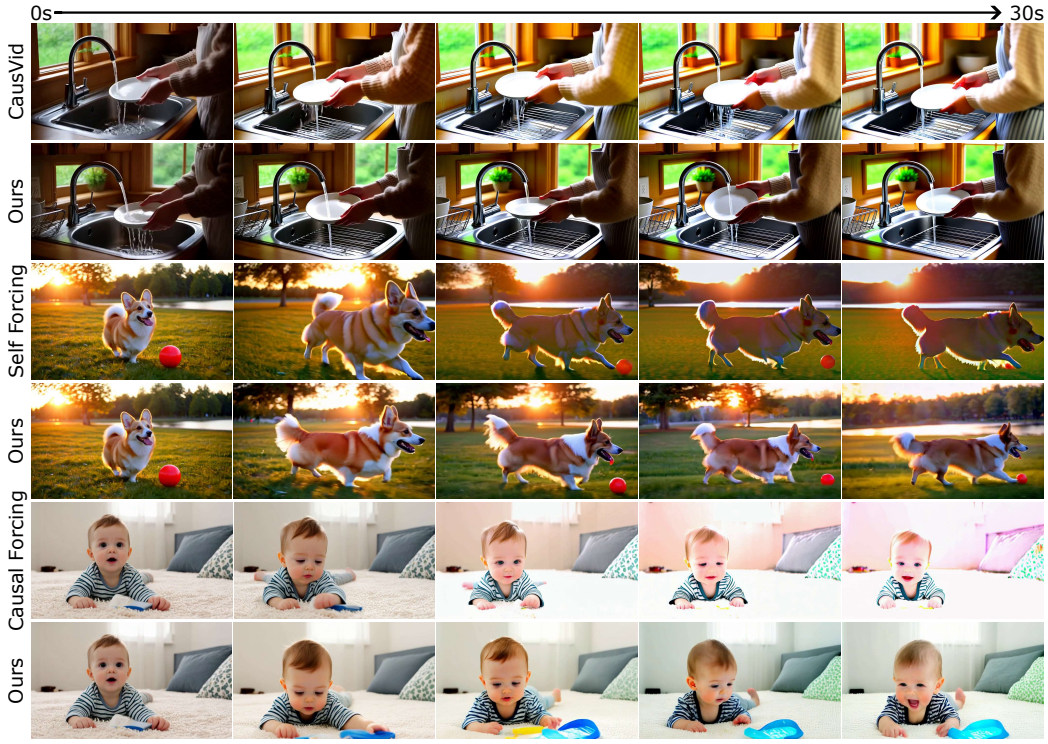


Figure 6: Qualitative results for 30-second video generation. Compared with the baselines, incorporating our proposed **Step Forcing** improves generation quality and mitigates generation drift.

Forcing as an example, Fig. 1 presents qualitative results. Although Deep Forcing can generate long-horizon videos, the results tend to exhibit slight generation drift and lack dynamics. Incorporating our method reduces drift while enhancing the dynamics of the generated videos.

C TEXT PROMPTS

The text prompts used to generate the videos shown in this paper are listed below.

Fig. 1: "A graceful white swan glides across a misty lake, dipping, splashing, and turning with serene elegance, its reflection mirroring every movement in the calm water.", and "A stylish woman walks down a Tokyo street filled with warm glowing neon and animated city signage. She wears a black leather jacket, a long red dress, and black boots, and carries a black purse. She wears sunglasses and red lipstick. She walks confidently and casually. The street is damp and reflective, creating a mirror effect of the colorful lights. Many pedestrians walk about.".

Fig. 2: "A brown bear walks in a grassy field.".

Fig. 3: "Aerial tracking camera shot. A white semi-truck drives on a highway".

Fig. 5: "A serene river winds through a lush, verdant forest, its crystal-clear waters reflecting the vibrant greens of the surrounding foliage. The scene begins with a close-up of the gentle current, revealing smooth pebbles and fish darting beneath the surface. As the camera pans out, the river's banks are lined with tall, ancient trees whose branches form a natural canopy overhead, dappling the water with sunlight. Birds flit between the trees, their songs harmonizing with the soft murmur of the river. Further downstream, a family of deer cautiously approaches the water's edge to drink, their reflections shimmering in the tranquil flow. The video concludes with a wide shot of the river meandering into the distance, disappearing into the heart of the forest, evoking a sense of peace and timeless beauty."

Fig. 6: “A person in a cozy kitchen setting is washing dishes at a sink filled with sudsy water. The individual, who appears to be middle-aged with casual attire, is scrubbing a plate with a sponge. Water droplets splash and cling to the dishes as they are washed. The kitchen is warmly lit, with sunlight streaming in from a window nearby. The background shows other kitchen appliances and cabinets. The person is standing with a focused expression, moving the sponge in circular motions across the dish. Medium close-up shot, static scene.”, “A cute happy Corgi playing in park, sunset”, and “A person is baby waking up”.