UniTemp: Unlocking Video Generation in Any Temporal Order via Bidirectional Distillation

Lin Zhang[1*], Sicheng Mo[3], Zefan Cai[1], Jinhong Lin[1], Zihao Lin[4], Jiuxiang Gu[2], Krishna Kumar Singh[2], Yuheng Li[2†], and Yin Li[1†]

[1] University of Wisconsin Madison, USA
[2] Adobe Research, USA
[3] University of California Los Angeles, USA
[4] University of California Davis, USA

Abstract. Autoregressive video diffusion models have emerged as a promising approach for long video generation, achieving strong performance in streaming settings. However, existing methods are restricted to forward temporal generation, whereas practical video creation often requires flexible generation order, *e.g.*, conditioning on future context to extend backward, or on both past and future context for inbetween generation. We bridge this gap by training *a single autoregressive model* that supports generation in arbitrary temporal directions. A key technical challenge arises from the Causal 3D VAE widely used in video diffusion models, where latents are encoded strictly conditioned on past context. While suited for forward generation, this causal structure causes inter-block discontinuities when generation proceeds backward. To address this, we introduce blockwise *anchor latents*, a set of auxiliary latents that aim to restore the missing past context at block boundaries during backward generation. Built on this design, we propose *UniTemp*, a bidirectional distillation framework that trains an autoregressive student model for any-direction video generation. At inference time, UniTemp conditions on arbitrary past and/or future frames, improving controllability for both bidirectional and inbetween generation. Through extensive experiments, we demonstrate that UniTemp maintains competitive performance on short and long video generation in comparison to forward-only methods, while enabling diverse creation workflows such as bidirectional video extension, inbetween generation, looping video generation, scene transition, and visual story generation. Project website: `https://lzhangbj.github.io/projects/unitemp/`.

Keywords: Video Generation · Autoregressive Models · Video Editing

1 Introduction

Diffusion transformers [10,20,27] have dominated the video generation landscape, achieving high-quality synthesis across diverse visual domains [2,15,19,21,29,35,

*Work partially done during internship at Adobe Research.
†Equal advising.

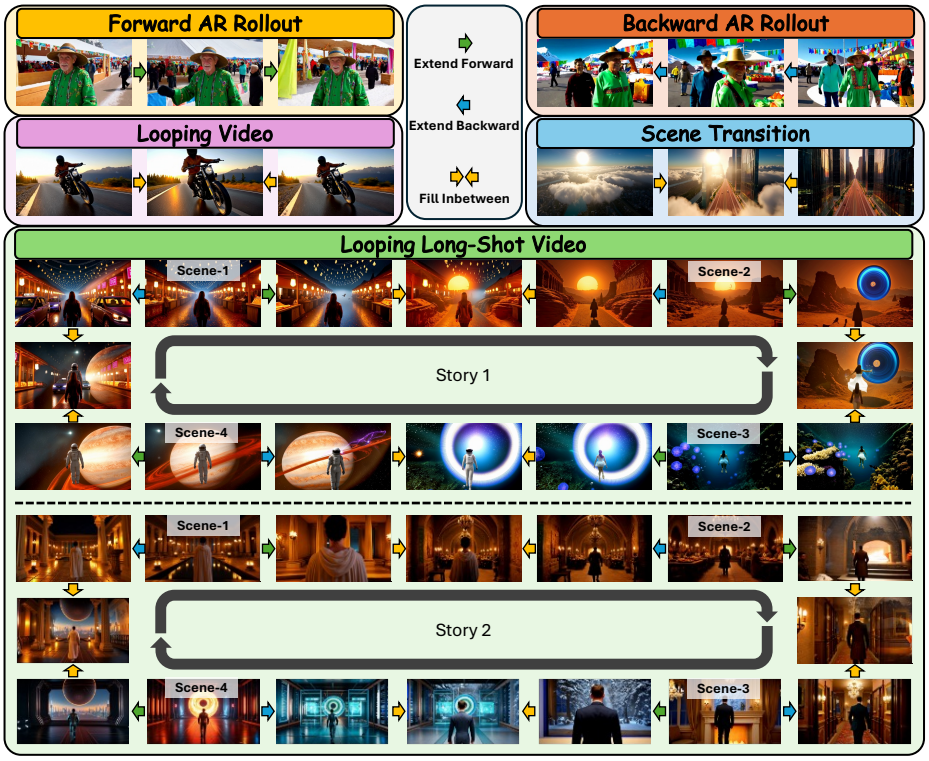

Fig. 1: We present **UniTemp**, a unified distillation framework that delivers a single model capable of flexibly generating video conditioned on past context, future context, or both, and supporting a wide range of generation tasks.

43]. Despite their impressive performance, these models require multiple denoising steps with full-sequence attention at inference, making them computationally expensive and difficult to scale to long video generation. Recently, asymmetric distillation [38, 39] has been introduced to address these limitations by distilling a large multi-step full-attention teacher model into a few-step autoregressive student. By generating videos block by block and caching prior key-value (KV) pairs, such models enable efficient streaming video generation [40].

Many existing approaches for autoregressive video generation only support *forward* (causal) video generation, where each block is conditioned exclusively on previously generated content. Real-world video creation workflows, however, often require flexible temporal ordering and iterative editing of generated content. For example, a user may wish to generate a prologue for an existing clip (backward extension), fill a temporal gap between two segments (inbetween generation), or generate key frames first then connect them into a visual story. While several solutions have been developed recently [1,7,9,14,26,28,31], they each address a specific temporal conditioning mode. An open question remains: *can we*

develop a unified autoregressive model capable of flexibly generating video conditioned on past context, future context, or both, thereby supporting a wide range of generation tasks? In this work, we aim to enable autoregressive video generation in any temporal direction. A natural starting point is to extend existing forward-only asymmetric distillation [12] to also support backward generation. However, we find that such a naive approach leads to noticeable inter-block discontinuities and boundary flickering in the generated content, particularly in scenes with high motion dynamics. We trace the root cause of these visual artifacts to *Causal 3D VAE* [42], a component widely adopted in modern video diffusion models [15, 19, 29, 35, 43] for efficient spatio-temporal encoding. Causal 3D VAE encodes video frames into latents that are strictly conditioned on its preceding context. While this causal structure aligns naturally with forward generation, it creates a fundamental mismatch for backward generation: when generating backward, the past context that the decoder expects is unavailable, causing visible discontinuities at block boundaries. Retraining a non-causal VAE is impractical, as it would forfeit compatibility with existing pre-trained teacher models.

To address this issue, we introduce blockwise **anchor latents**, a mechanism that restores the missing past context at block boundaries during backward generation. The key intuition is to jointly denoise a small set of auxiliary latents standing for preceding video tokens alongside the target block under full-sequence attention, allowing the generated frames to attend to a proxy of prior local context even when generating in reverse order. These anchor latents act as a bridge between consecutive blocks, providing smooth, artifact-free transitions without requiring any modification to the pre-trained VAE or teacher model.

Built on this design, we present **UniTemp**, a unified distillation framework that trains a single autoregressive student model, supervised by a frozen teacher diffusion model, to support generation in any direction. UniTemp adopts an efficient training procedure via shared models between forward and backward generation tasks. During inference, UniTemp flexibly conditions on past context, future context, or both, enabling any-direction video generation and unlocking a diverse set of applications all in one model (see Fig. 1).

Our contributions are summarized as follows.

1. We introduce UniTemp, *a unified distillation framework that enables autoregressive video generation in any temporal direction.* The framework maintains training efficiency while delivering a single model for flexible test-time conditioning with fast inference.
2. We propose using *anchor latent* to substantially reduce inter-block flickering when extending existing training frameworks to backward video generation. We trace this artifact to the design of Causal 3D VAE, and introduce blockwise anchor latents that restore the missing past context at block boundaries.
3. We demonstrate competitive performance on short, long, and inbetween video generation tasks in comparison to forward-only methods, while unlocking *new capabilities* such as bidirectional video extension, scene transition, looping video generation, and visual story generation.

2 Related Work

Temporal dependency in video generation. Video generation tasks differ in their temporal conditioning: text-to-video [2,21,29] generates from scratch, video extension [1, 9] conditions on the past, and inbetween generation [6, 22, 28, 31] conditions on both endpoints. Supporting all these modes in a single model remains an open challenge. Prior methods [11,14,17,28,41] for flexible conditioning adopt mask-guided inpainting with full-sequence attention. MCVD [28] concatenates past and future frames with noisy targets along the channel dimension, randomly masking past or future frames to handle both extension and interpolation. RaMViD [11] randomly masks subsets of frames during training, applying diffusion only to the masked frames to enable conditioning on arbitrary positions. VACE [14] unified a broad range of generation and editing tasks through an input video mask that distinguishes guided regions from generated ones. However, these approaches require full-sequence denoising with sophisticated input formats (*e.g.*, masks) and redundant feature extraction, and thus are slow at inference. They also pose training difficulties because they require massive real video data for pretraining or fine-tuning.

Our method instead unifies these tasks within a single distillation framework. During training, this distillation approach remains efficient and does not require real video data, unlike previous pretraining- or fine-tuning-based approaches. During inference, we adopt temporal position indices in RoPE [25] to specify conditioning positions, and use KV caches to avoid redundant computation over previously generated blocks, thereby achieving *significantly faster inference*.

VAE in latent video diffusion models. Generating videos directly in pixel space is computationally prohibitive. Latent diffusion models [23] address this by compressing videos into a compact latent space via a variational autoencoder (VAE) and training a diffusion model in that space. State-of-the-art models [2, 15, 29, 35] adopt diffusion transformers (DiT) [20] as the backbone, operating on spatio-temporal latent tokens with self-attention and text conditioning via cross-attention. A critical design choice is the VAE architecture. Causal 3D VAEs [42] have become the dominant choice for their advantages in (1) unified image and video tokenization, (2) improved generation quality, and (3) reduced memory usage. They are thus widely adopted by state-of-the-art video diffusion models including CogVideoX [35], Cosmos [19], HunyuanVideo [15], and Wan [29]. However, a causal 3D VAE introduces a temporal bias: each latent's encoding and decoding depend only on its past context. This bias, which has not been well studied in prior work, becomes a critical obstacle when generation proceeds backward, as we will show in Sec. 4.1. In this work, we address this issue with blockwise anchor latents, a sequence of preceding video latents that participate in the denoising of each block, effectively stabilizing reverse block generation and largely reducing artifacts.

Autoregressive video generation via distillation. Pretraining video generation models on massive long-video datasets is challenging. Asymmetric distillation [40] was proposed as a post-training technique that distills a multi-step full-

attention teacher trained on short horizons into a few-step autoregressive student for long video generation. The student generates videos block by block with KV caching, enabling efficient streaming synthesis for long videos. Self-Forcing [12] further bridges the train-test distribution gap by conditioning on the student's own rollouts rather than ground truth videos during training, largely reducing quality degradation over long horizons. Subsequent work has stabilized long generation through training-based [4, 16, 18, 34] and training-free [3, 36, 37] techniques. These methods, however, support only forward generation. Our UniTemp builds on the Self-Forcing framework for efficient training but departs from prior work by training a single student for any-direction generation.

3 Background and Preliminaries

We provide background on latent video diffusion models with Causal 3D VAEs and on self-forcing for autoregressive video diffusion models. Readers who are knowledgeable about video generation may skip the rest of this section.

3.1 Latent Video Diffusion Models and Causal 3D VAEs

Latent video diffusion models [23] (LVDMs) encode frames $\{x_i\}$ into VAE latents $\{z_i\}$, then use a denoiser (*e.g.*, U-Net [24] or DiT [20]) to generate latents from noise. At each noise level t, the denoiser maps noisy latents $\{z_i^t\}$ to cleaner latents $\{z_i^{t-1}\}$ conditioned on text.

Most recent LVDMs adopt a Causal 3D VAE [15, 19, 29, 35, 42, 43], which uses an encoder $\mathcal{E}$ and a decoder $\mathcal{D}$ to encode and decode each latent z_i using its past context $z_{\leq i}$, as shown in Fig. 2. With a temporal compression factor of 4, z_0 is a special image latent encoding the initial frame x_0, and z_i ($i > 0$) encodes every four consecutive frames $(x_{4i}, \ldots, x_{4i+3})$. This supports streaming and joint image-video tokenization [42], but creates a forward temporal bias (Sec. 4.1).

3.2 Self-Forcing

Self-Forcing [12] presents a widely adopted training paradigm for autoregressive video diffusion models. It distills a slow teacher, such as Wan2.1 [29], into a few-step autoregressive student G^θ through two training stages. Stage 1 initializes G^θ with L teacher-generated ODE trajectories $\{z_l^t\}_{l=1}^L$, where t covers the student's denoising steps $\{t_i\}_{i=1}^N$, by regressing each noisy latent to its clean target

$$L_{\text{init}} = \mathbb{E}_{z^0,i} \left\| G^\theta(z^{t_i}, t_i) - z^0 \right\|_2^2, \tag{1}$$

where z^0 denotes the clean latent and z_{t_i} is the noisy latent at denoising step t_i along the teacher's ODE trajectory. Stage 2 trains G^θ in the same blockwise autoregressive rollout used at inference, where each block is denoised conditioned on previously generated blocks rather than ground truth. For block size $B = 3$, this follows $(z_0, z_1, z_2) \rightarrow (z_3, z_4, z_5) \rightarrow \cdots \rightarrow (z_{18}, z_{19}, z_{20})$. Distribution

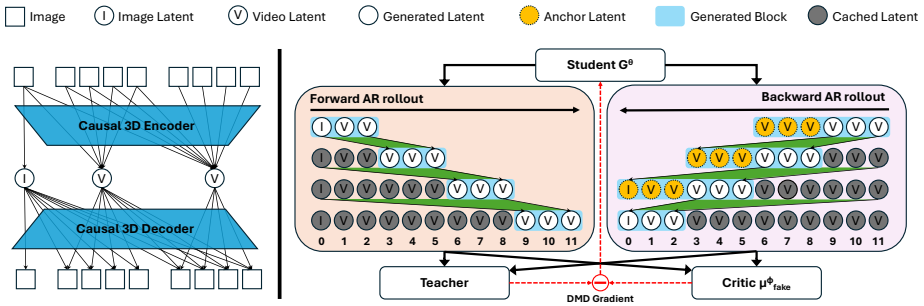

Fig. 2: Left: Causal design of the frozen 3D VAE. It encodes video into spatial-temporal latents (V) with a leading image latent (I). Each latent is dependent on its past context. **Right**: Overview of **UniTemp**. We distill a teacher model into a unified autoregressive student G^θ trained on its self-rollout in both forward and backward directions. In backward generation, we introduce **blockwise anchor latents (dashed circles)** to reduce inter-block flickering. The anchor latents only serve to stabilize generated content by providing approximate missing past context. After being denoised with the current block, they are discarded and not included as outputs.

matching distillation (DMD) [38, 39] matches the student's output distribution to the teacher's by updating G^θ with

$$\nabla_\theta D_{\text{KL}} = \mathbb{E}_{z\sim\mathcal{N}(0;I),x=G^\theta(z)} \left(s_{\text{real}}(x) - s_{\text{fake}}(x)\right) \frac{\partial G^\theta}{\partial \theta}, \tag{2}$$

where s_{real} is the score computed from the frozen teacher, and s_{fake} is the score from a fake denoiser μ^ϕ_{fake}. μ^ϕ_{fake} is trained to approximate the student's distribution using

$$L^\phi_{\text{denoise}} = \left\| \mu^\phi_{\text{fake}}(z^{t_i}, t_i) - z^0 \right\|_2^2. \tag{3}$$

As in GAN training [8], μ^ϕ_{fake} is updated with L^ϕ_{denoise}, while G^θ is updated with Eq. (2) to match the teacher, as shown in Fig. 2.

4 UniTemp: Design, Training and Inference

Our goal is to distill a single model that supports video generation in arbitrary temporal orders at test time, thereby enabling a wide range of practical video creation workflows. In Sec. 4.1, we show that directly extending an existing distillation framework [12] to backward generation leads to severe inter-block flickering. We trace this artifact to the latent causality induced by the Causal 3D VAE, and propose blockwise anchor latents to effectively stabilize backward generation. In Sec. 4.2, we further integrate anchor latents into UniTemp, our unified distillation framework that trains a single student model to support generation in any temporal directions. Finally, in Sec. 4.3, we discuss the versatility of UniTemp in any-order generation tasks at test time. An overview of our approach is shown in Fig. 2.

4.1 Blockwise Anchor Latents for Backward Generation

A unified generation framework necessitates backward generation capability. An intuitive approach is to extend Self-Forcing [12] for backward generation by reversing the student's block generation order, *i.e.*, $(z_0, z_1, z_2) \leftarrow \cdots \leftarrow (z_{15}, z_{16}, z_{17}) \leftarrow (z_{18}, z_{19}, z_{20})$, with each block generated conditioned on its future blocks[5]. In practice, however, we observe severe flickering in backward-generated videos, as shown in Fig. 3. This flickering appears periodically at block boundaries, often as ghosting/flashing artifacts covering the whole image, and is more obvious in high-motion local regions.

Evaluating periodic flickering. To better understand this issue, we need a metric to evaluate these flickering artifacts. Temporal flickering in VBench [13] serves as a good starting point, but with two drawbacks: (1) it averages frame-frame pixel differences over the whole video rather than in local temporal context, and thus fails to identify when the flickering happens, (2) it is highly influenced by video dynamics, where high-dynamic scenes are more likely to have lower flickering scores. We thus define Flickering Ratio on top of temporal flickering to address these two problems. For two neighboring latents z_i and z_{i+1} corresponding to frames $(x_{4i}, x_{4i+1}, x_{4i+2}, x_{4i+3})$ and $(x_{4i+4}, x_{4i+5}, x_{4i+6}, x_{4i+7})$, respectively, we compute the flickering ratio (FR) as

$$\mathrm{FR}(z_i, z_{i+1}) = \frac{\sum_{h,w,c} |x_{4i+4}(h,w,c) - x_{4i+3}(h,w,c)|}{\sum_{h,w,c} |x_{4i+3}(h,w,c) - x_{4i+2}(h,w,c)|}. \tag{4}$$

The numerator measures the pixel L1 distance between decoded frames at latent boundary $x_{4i+3} \rightarrow x_{4i+4}$. The denominator measures the pixel L1 difference between two frames decoded by the same latent, which serves as a good indicator of dynamics in the local clip. By dividing, we measure cross-latent flickering relative to the local dynamics, which is more robust than simple pixel differences.

We sample a set of dynamic prompts to measure FR. For each video, we split FR into the following two groups and average the value within each group: (1) inter-block FR, where z_i and z_{i+1} belong to different generated blocks, and (2) intra-block FR, where z_i and z_{i+1} belong to the same generated block. Specifically, since intra-block latents can mutually attend to each other with full-sequence self-attention at every layer, they are expected to demonstrate higher temporal smoothness, thus lower FR. Inter-block latents can only attend in one direction through cached keys and values, and therefore should demonstrate higher FR. We validate this empirically in Tab. 1, where both forward and backward generation show inter-block FRs larger than 1 (1.15/1.42), while intra-block FRs remain at smaller values (0.93/0.89).

Flickering due to Causal 3D VAE. The abnormally high inter-block FR in backward generation (1.42) aligns with our aforementioned empirical observation of cross-block flickering. We attribute this to the causal dependency in the encoded latents $\{z_i\}$, which is introduced by a Causal 3D VAE. A Causal 3D VAE imposes a strong dependency of each generated latent z_i on its past

[5] By default, we use a block size of $B = 3$ in the rest of this section.

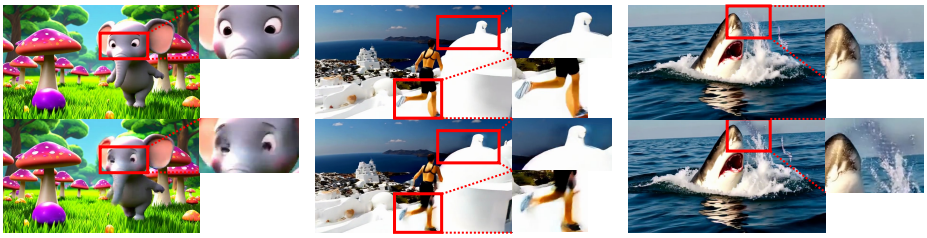

Fig. 3: Visualization of **inter-block flickering in backward generation**. Without anchor latents, visible discontinuities appear at block boundaries.

context $z_{<i}$ but not future context $z_{>i}$, as shown in Fig. 2 left. In the forward process, generation of each latent z_i naturally follows this causality since it can always attend to past latents $z_{<i}$. However, in backward generation, the latent z_i near the left boundary of a block (z_i, z_{i+1}, z_{i+2}) can only attend to its future context $z_{>i}$, without past context. In contrast, full-sequence attention is applied within the block so that z_{i+1} and z_{i+2} have local past context (z_i) and (z_i, z_{i+1}), respectively. Intra-block FR in backward generation can therefore maintain a low value, as in forward generation.

Our solution: blockwise anchor latents. Motivated by our analysis, we seek to solve this inter-block flickering issue via an approximation of the missing past context. Specifically, in every backward-generated block, we expand its block size from B to $(P + B)$, where P is the number of anchor latents. Each block is now $(z_{i-P}, ..., z_{i-1}, z_i, z_{i+1}, z_{i+2})$ instead of (z_i, z_{i+1}, z_{i+2}). This expanded block is jointly denoised with full-sequence self-attention within the block, while attending to stored KV cache entries of future latents $z_{>i+2}$ to receive future conditioning. Note that while these anchor latents $(z_{i-P}, ..., z_{i-1})$ can fill in the missing past context for z_i, they themselves (especially z_{i-P}) now stay near the left boundary of the block, lacking their own past context. Therefore, after denoising, we discard the denoised anchor latents and do not include them as outputs. This idea is illustrated in Fig. 2 right.

We empirically verify that these anchor latents effectively reduce inter-block flickering, as shown in Tab. 1. We defer the detailed discussion of our validation to Sec. 5.2.

4.2 Unified Training with Bidirectional Distillation

We now integrate blockwise anchor latents into UniTemp, our unified distillation framework for any-order temporal generation.

Bidirectional distillation. During training, we adopt Self-Forcing [12] as the training algorithm and train on both forward and backward tasks, as shown in Fig. 2. The forward generation pipeline faithfully follows the original Self-Forcing with blockwise ($B = 3$) autoregressive generation. In backward generation, we

introduce anchor latents to expand the block size from B to $(P+B)$, while the DMD [38, 39] loss is applied to the student's output without anchor latents.

Unified models. Since training includes two directional targets, a natural choice is to consider separate models for forward and backward generation, *i.e.*, (1) separate generators: we can use two generators $G^{\theta}_{\rightarrow}$ and $G^{\theta}_{\leftarrow}$ to generate forward and backward videos, respectively, (2) separate fake critics: we can use two fake critics $\mu^{\phi}_{\text{fake},\rightarrow}$ and $\mu^{\phi}_{\text{fake},\leftarrow}$ to approximate the distributions of the forward- and backward-generated videos, respectively. However, separating models as in previous methods [31] incurs significantly higher memory cost during training and inference. Separate generators also lose the ability to support flexible temporal conditioning (*e.g.*, inbetween generation) at test time. Therefore, we adopt a unified model design, sharing a single student model G^{θ} and a single fake critic model μ^{ϕ}_{fake} across both generation tasks. Such a shared design can also reduce the performance gap between forward and backward generation (see Sec. 5.2).

4.3 Versatile Test-Time Capabilities

By training a shared generator in both directions, UniTemp allows generating each block conditioned on any combination of past and future KV caches, thereby unlocking versatile test-time capabilities. We discuss these new capabilities and defer to Sec. 5.1 and Sec. 5.3 for further validation and demonstration.

Both-end-guided sink latents. Sink latents [32] have proven to significantly improve inference performance and efficiency over long horizons. While previous methods [3, 4, 16, 18, 34, 36, 37] can only place sink latents at past positions to stabilize long video generation, we can now shift these latents into the future by reapplying their RoPE [3, 25, 36] temporal indices, simulating future sink latents for forward generation. Similarly, in backward generation, we can reposition earlier generated future latents into the past. This yields both-end-guided sink latents, further reducing quality drift and content variation in long video generation.

Inbetween generation. Surprisingly, we observe that, although the block size is set to B (or $P+B$ with anchor latents) during training, the test-time block size can be flexibly set to any value as long as the maximum attention horizon is less than that in training, *i.e.* 21. This enables inbetween generation [5,6,22,28,29,31], which infills the content of K latents between a head block and a tail block. Specifically, given a head block of B latents and a tail block of B latents, we can first obtain their KV caches by feeding them to the denoiser with clean context noise and a frame gap of K in their RoPE temporal indices. Then, we can produce the intermediate K latents from noise by applying full-sequence self-attention within the K latents and attending to the head and tail KV caches.

5 Experiments and Results

Training details. We train UniTemp following the training procedure of Self-Forcing [12]. Specifically, we use Wan2.1 T2V 14B [29] as the pretrained teacher

model. The student is a 1.3B model initialized from Wan2.1 T2V 1.3B with 4 denoising steps $(1000.0, 937.5, 833.3, 625.0)$. In the first stage, we use the teacher to generate 16K ODE pairs, and train the student model for 6K iterations with a batch size of 128. Both forward and backward attention masks are applied in each training iteration. In the second stage, we train the student for 600 iterations with a batch size of 64 on the rewritten VidProm [30] prompts. We apply exponential moving average (EMA) to the student model with a decay rate of 0.999 starting from iteration 200. We choose a block size of $B = 3$ and an anchor latent size of $P = 3$ for backward generation, unless otherwise specified.

Evaluation protocol, metrics and baselines. We primarily evaluate using metrics in VBench [13]. While our trained model supports a diverse set of applications, we focus on the following four dimensions.

1. **Flickering ratio**. This is the metric described in Sec. 4.1. Since flickering is more obvious in high-dynamic video, we use an LLM to generate 128 high-dynamic video prompts to evaluate the flickering ratio.
2. **Single-direction short video generation**. Following conventional evaluation protocol in previous works [12, 34], we generate 5s videos using 946 official VBench prompts rewritten by Qwen2.5-7B-Instruct [33]. Metrics are computed following the official VBench codebase.
3. **Single-direction long video generation**. Following previous approaches [4, 16, 36], we randomly sample 128 prompts from MovieGenBench [21] to generate 100s long videos for each prompt. We then use metrics in VBench to evaluate video quality at 10s, 30s, and 100s. We compare to a set of strong long video generation baseline approaches based on Self-Forcing [12], including LongLive [34], Rolling-Forcing [16], and Infinity-Rope [36].
4. **Inbetween video generation**. We use Wan2.1 T2V 14B to generate 5s ground-truth videos using the same 128 prompts from MovieGenBench. We keep the head block (9 frames) and tail block (12 frames) as conditions, and fill the inbetween region. In addition to VBench metrics, we additionally measure head / tail frame flickering, the pixel consistency between generated and ground-truth condition frames, and FID / FVD between generated and ground-truth videos. We compare to Wan FLF2V 14B [29] and Generative Inbetweening (GI) [31].

5.1 Main Results and Discussion

Single-direction short video generation. Tab. 2 reports VBench results on 5s video generation. The configuration with separate G^θ and separate μ^ϕ_{fake} (first row) corresponds to the Self-Forcing [12] baseline. In comparison, our unified method achieves competitive performance in both directions compared to the Self-Forcing baseline. We also observe slightly higher performance in backward generation. This can be attributed to learning asymmetry rooted in forward bias in the latent space.

Table 1: Ablation study on anchor latents for backward generation.

Direction	Num Anchor P	Anchor as Output		Inter-block FR ↓	Intra-block FR ↓
		Train	Inference		
→	0	✗	✗	1.15	0.93
←	0	✗	✗	1.42	**0.89**
←	1	✗	✗	1.26	0.93
←	2	✗	✗	1.23	0.93
←	3	✗	✗	**1.07**	0.97
←	3	✗	✓	1.46	0.99
←	3	✓	✗	1.08	0.97
←	3	✓	✓	1.48	0.98

Table 2: Ablation study on training with shared models. Results evaluated with VBench [13] on short (5s) video generation.

Direction	Shared G^θ	Shared μ^ϕ_{fake}	Quality ↑	Semantics ↑	Total ↑
→	✗	✗	84.47	79.23	83.42
	✓	✗	84.64	79.18	83.54
	✓	✓	**84.89**	**79.51**	**83.81**
←	✗	✗	**85.17**	80.61	84.26
	✓	✗	85.02	79.74	83.96
	✓	✓	85.12	**80.69**	**84.23**

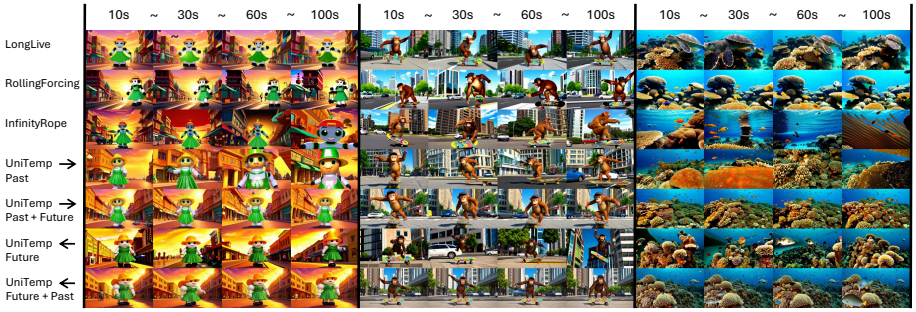

Fig. 4: Long video generation results. Past + future sink latents provide strong conditioning to reduce content variation over long durations.

Single-direction long video generation. Tab. 3 compares long video generation at 10s, 30s, and 100s. Existing training-based long video generation methods (LongLive [34], Rolling-Forcing [16]) achieve high temporal consistency but produce extremely low-dynamic content (36.72/28.12). Note that this is a trade-off in VBench metrics: more dynamic content usually leads to lower temporal consistency, because there are larger differences between neighboring frames. In comparison, training-free methods with past sink latents [3] can already achieve strong consistency with slightly degraded dynamics and aesthetic quality. E.g., from 10s to 100s, Infinity-Rope's dynamic degree / aesthetic quality drops from 64.06/61.17 to 55.47/56.26, while Self-Forcing's dynamic degree / aesthetic quality decreases from 64.06/62.74 to 62.74/56.85. Our UniTemp inherits such stable long video generation capability but with increased dynamic degree, which leads to lower frame consistency. However, we can control such dynamics-consistency tradeoff by switching between single-end sink latents and both-ends sink latents. In forward generation, by enabling both-end-guided sink latents, the generated content faithfully follows the sink frames, effectively reducing drifting from 10s to 100s with (1) stable and high subject consistency (97.60 → 97.30) (2) lower but stable dynamic degree (46.80 → 44.50) and (3) stable aesthetic quality (62.70 → 62.30). Backward generation shows similar patterns. Furthermore, we can see the visual results in Fig. 4, where UniTemp can generate stable long videos with content variations controlled by single / both sink latents.

Table 3: Long video generation results on VBench [13]. We compare forward-only baselines with UniTemp in both directions under various sink latent configurations. Past/future sink latents provide temporal anchoring to control content variation over long durations.

Direction	Method	Past Sink	Future Sink	Dur.	Subj. ↑	BG ↑	Flicker ↑	Smooth ↑	Dynamic ↑	Aes. ↑	Img. ↑
→	LongLive [34]	✓	✗	10s	96.84	96.26	**97.77**	**98.84**	36.72	62.53	70.65
				30s	96.45	95.45	**97.65**	**98.75**	38.28	61.12	69.20
				100s	96.48	95.52	**97.71**	**98.76**	36.72	61.81	69.22
	Rolling-Forcing [16]	✓	✗	10s	96.78	95.48	97.65	98.70	28.12	62.03	71.16
				30s	96.60	95.37	**97.65**	98.72	31.25	60.98	**70.96**
				100s	96.06	95.19	97.45	98.57	29.69	58.96	70.99
	Infinity-Rope [36]	✓	✗	10s	95.48	94.74	96.29	97.93	64.06	61.17	70.18
				30s	95.11	94.63	96.13	97.82	59.38	58.78	69.38
				100s	94.93	94.53	96.05	97.74	55.47	56.26	69.39
	Self-Forcing [12]	✓	✗	10s	96.47	95.36	96.36	98.31	64.06	62.74	**71.23**
				30s	95.75	95.01	95.88	98.08	64.06	59.84	70.33
				100s	96.14	94.84	96.01	98.12	62.50	56.85	70.75
	UniTemp	✓	✗	10s	95.40	94.37	95.11	97.69	**81.25**	61.64	70.22
				30s	94.36	93.54	94.53	97.43	**92.97**	57.70	69.10
				100s	93.95	93.23	94.43	97.37	**87.50**	56.55	69.53
	UniTemp	✓	✓	10s	**97.60**	**96.30**	97.50	98.50	46.80	62.70	70.70
				30s	**97.40**	**96.00**	97.00	98.10	44.50	**62.90**	70.80
				100s	**97.30**	**95.80**	96.90	98.00	44.50	62.30	**71.30**
←	UniTemp	✗	✓	10s	94.91	94.57	95.94	98.10	71.09	**63.35**	70.38
				30s	93.89	93.67	94.93	97.73	81.25	60.31	70.71
				100s	93.94	93.50	94.67	97.64	85.16	59.02	70.51
	UniTemp	✓	✓	10s	96.87	96.03	97.13	98.34	55.47	62.95	70.39
				30s	96.80	95.60	97.04	98.28	60.16	62.69	70.49
				100s	96.75	96.00	97.04	98.28	56.25	**62.83**	70.31

Inbetween video generation. Tab. 4 evaluates inbetween generation given the first and last frames. Wan FLF2V [29] is a large 14B-parameter model with high computational cost that has been trained for inbetween generation tasks on a large amount of video data. GI is built on Stable Video Diffusion [1] with a fine-tuned backward image-to-video diffusion model. At inference, the denoising steps need to be run for both forward and backward image-to-video generation, leading to inefficiency in both memory and speed. Both Wan FLF2V and GI support only a single frame on each side for conditioning. In comparison, UniTemp is not specifically trained for inbetween generation tasks and has only one unified model. The shared architecture allows simultaneously conditioning on both head and tail frames, while its autoregressive nature allows conditioning on flexible numbers of head and tail frames. UniTemp shows strong performance by outperforming GI and Wan FLF2V in background consistency, image quality, and video distribution. Different from GI and Wan FLF2V, UniTemp can reuse latents of conditioning frames and thus does not need to reproduce the conditioning frames, leading to much higher boundary consistency. Qualitatively, as shown in Fig. 5, GI often assumes linear motion between head and tail frames, resulting in sometimes unnatural content and sudden blending effects, especially when the head and tail frames are far apart (the second case). In contrast, Wan

Table 4: Inbetween video generation results. We compare UniTemp against Wan FLF2V [29] and GI [31] on bounded generation given head and tail frames. Head/tail flickering measures boundary consistency with the ground-truth frames.

	VBench Dimensions							Boundary		Distribution	
Method	Subj. ↑	BG ↑	Flicker ↑	Smooth ↑	Dynamic ↑	Aes. ↑	Img. ↑	Head ↑	Tail ↑	FID ↓	FVD ↓
GI [31]	**94.48**	92.90	97.27	**98.75**	62.50	60.19	62.88	96.29	97.47	34.21	34.66
Wan FLF2V [29]	93.44	93.63	**97.38**	98.54	**64.84**	64.38	64.59	98.88	**98.52**	**23.31**	27.77
UniTemp	93.81	**93.71**	97.18	98.46	63.28	**65.75**	**66.68**	**99.52**	98.16	24.94	**25.00**
Ground-truth	94.11	94.78	97.64	98.66	57.81	61.17	64.25	100.00	100.00	–	–

Fig. 5: Visualization of inbetween video generation. Given the head (leftmost) and tail (rightmost) frames, UniTemp infills temporally coherent content.

FLF2V and UniTemp can generate intermediate content with natural variations, while UniTemp is much more efficient in both training and inference.

5.2 Ablation Studies

Effect of anchor latents Tab. 1 reports inter-block and intra-block flickering ratio for varying numbers of anchor latents P. The baseline without anchor latents exhibits a backward inter-block flickering ratio of 1.42, confirming substantial boundary artifacts. Increasing P progressively reduces this: P=1 (1.26), P=2 (1.23), P=3 (**1.07**). At P=3=B, the ratio approaches the ideal value of 1.0. In addition, we investigate whether the anchor latents should be included as outputs in training and inference. When included as outputs in training, the loss is applied to the anchor latents. As discussed in Sec. 4.1, anchor latents themselves are generated without past context. Therefore, once included in the outputs at test time, they are positioned near block boundaries and still show inter-block flickering, as validated in Tab. 1 (1.07 → 1.46, 1.08 → 1.48). We show video visualizations of this flickering in the appendix.

Unified vs. separate models. Tab. 2 compares three training configurations. Overall, training direction-specific separate models for either the student generator G^θ or fake diffusion model μ^ϕ_{fake} does not bring significant performance gains but introduces higher training cost. Our unified training framework can also reduce the performance difference between forward (84.89/79.51/83.81) and backward (85.12/80.69/84.23) generation. This demonstrates the effectiveness and efficiency of our unified training framework.

Table 5: User study for evaluating cross-block flickering. We report preference between videos generated with and without anchor latents, together with agreement between user preference and the proposed inter-block flickering ratio (FR).

Number of Responses	Preference Rate		Agreement with inter-block FR
	w/ Anchor	w/o Anchor	
200	87%	13%	80.5%

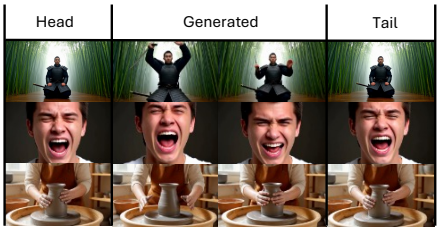

Fig. 6: Application: looping video generation with same head/tail frames.

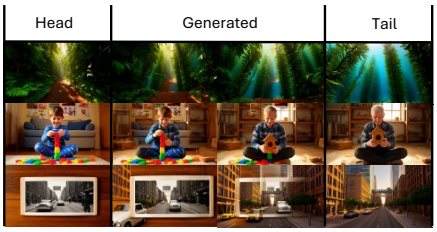

Fig. 7: Application: scene transition given distinct head and tail frames.

Subjective assessment of cross-block flickering We conduct a user study with A/B tests to subjectively assess cross-block flickering. Each user watches 20 random pairs of videos. Each pair of videos are generated with and without anchor latents, with the same input prompt. We ask users to opt for the one with less cross-block flickering. Tab. 5 shows an 87% user preference on videos generated with anchor latents. Additionally, we choose the videos with less inter-block flickering ratio (FR), and compare it with users' preference. This shows 80.5% agreement between human preference with our inter-block flickering ratio (FR), suggesting it as a valid metric to assess inter-block flickering.

More analysis. Additionally, we evaluate training and inference efficiency, demonstrating superior efficiency of UniTemp in both stages. We show the results and details in the supplementary material.

5.3 Applications

Since UniTemp enables flexible conditioning at inference time, we demonstrate its effectiveness across several applications. Implementation details for these applications are provided in the supplementary material.

Looping video generation. By shifting the head conditioning frames to the future and then generating the intermediate content between them, we can easily achieve looping video generation, as shown in Fig. 6.

Scene transition. When the head and tail frames are highly different, we face the challenging task of scene transition. Surprisingly, we find that UniTemp can

perform reasonably well in this task between contrasting scenes. As shown in Fig. 7, UniTemp can associate visual similarities and apply a smooth transition.

Long-shot visual story generation. With versatile generation orders and conditioning frames, we are now able to build visual stories in any order we want, instead of only extending forward. For example, in Fig. 1, we can generate multiple key scenes first, then flexibly extend forward / backward / infill between any two existing scenes, allowing looping long-shot visual story generation across different numbers of scenes and durations.

6 Conclusion and Discussion

We presented UniTemp, a unified distillation framework for training autoregressive video generation models that support arbitrary temporal generation orders. We identified the cross-block flickering issue in backward-generated videos, which is associated with the causal temporal bias of Causal 3D VAE latents. We proposed blockwise anchor latents to help restore missing past context at block boundaries. Built on these anchor latents, UniTemp trains a single student model for both forward and backward generation. Computed features can then be shared across any-direction generation workflows, enabling efficient inference. Empirically, UniTemp achieves competitive performance on short- and long-video generation, as well as in-between generation, while enabling applications such as looping video generation, scene transitions, and visual story generation, all without task-specific fine-tuning.

Limitations. Backward generation incurs additional per-block computation due to the jointly denoised anchor latents; forward generation is unaffected. The causal VAE and teacher model remain inherently forward-biased: anchor latents mitigate but do not fully eliminate this asymmetry. Exploring bidirectional VAE architectures that maintain compatibility with existing pre-trained teachers is a promising direction for future work.

Acknowledgment. This material is partially based on work supported by the Army Research Office funded Assured Autonomy Innovation Institute (A2I2) under Contract number W911NF-2020-221, and the National Science Foundation under Grant Numbers CNS-2333491 and IIS-2442739. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsors.

References

1. Blattmann, A., Dockhorn, T., Kulal, S., Mendelevitch, D., Kilian, M., Lorber, D., Levi, Y., English, Z., Voleti, V., Letts, A., Jampani, V., Rombach, R.: Stable video diffusion: Scaling latent video diffusion models to large datasets. In: Proceedings of the International Conference on Learning Representations (ICLR) (2024)

2. Brooks, T., Peebles, B., Holmes, C., DePue, W., Guo, Y., Jing, L., Schnurr, D., Taylor, J., Luhman, T., Luhman, E., Ng, C., Wang, R., Ramesh, A.: Video generation models as world simulators. OpenAI Technical Report (2024), https://openai.com/index/video-generation-models-as-world-simulators/, 1 July 2026
3. Chen, J., Fu, Z., He, X.: Infinite-forcing: Towards infinite-long video generation (2025), https://github.com/SOTAMak1r/Infinite-Forcing, 1 July 2026
4. Cui, J., Wu, J., Li, M., Yang, T., Li, X., Wang, R., Bai, A., Ban, Y., Hsieh, C.J.: Self-forcing++: Towards minute-scale high-quality video generation. In: Proceedings of the International Conference on Learning Representations (ICLR) (2026)
5. Danier, D., Zhang, F., Bull, D.: Ldmvfi: Video frame interpolation with latent diffusion models. In: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI) (2024)
6. Feng, H., Ding, Z., Xia, Z., Niklaus, S., Abrevaya, V., Black, M.J., Zhang, X.: Explorative inbetweening of time and space. In: Proceedings of the European Conference on Computer Vision (ECCV) (2024)
7. Geyer, M., Bar-Tal, O., Bagon, S., Dekel, T.: Tokenflow: Consistent diffusion features for consistent video editing. In: Proceedings of the International Conference on Learning Representations (ICLR) (2024)
8. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in Neural Information Processing Systems (NeurIPS) (2014)
9. Henschel, R., Khachatryan, L., Poghosyan, H., Hayrapetyan, D., Tadevosyan, V., Wang, Z., Navasardyan, S., Shi, H.: Streamingt2v: Consistent, dynamic, and extendable long video generation from text. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2025)
10. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. In: Advances in Neural Information Processing Systems (NeurIPS) (2020)
11. Höppe, T., Mehrjou, A., Bauer, S., Nielsen, D., Dittadi, A.: Diffusion models for video prediction and infilling. Transactions on Machine Learning Research (TMLR) (2022)
12. Huang, X., Li, Z., He, G., Zhou, M., Shechtman, E.: Self-forcing: Bridging the train-test gap in autoregressive video diffusion. In: Advances in Neural Information Processing Systems (NeurIPS) (2025)
13. Huang, Z., He, Y., Yu, J., Zhang, F., Si, C., Jiang, Y., Zhang, Y., Wu, T., Jin, Q., Chanpaisit, N., et al.: Vbench: Comprehensive benchmark suite for video generative models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2024)
14. Jiang, Z., Han, Z., et al.: Vace: All-in-one video creation and editing. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2025)
15. Kong, W., Tian, Q., Zhang, Z., Min, R., et al.: Hunyuanvideo: A systematic framework for large video generative models. arXiv preprint arXiv:2412.03603 (2024)
16. Liu, K., Hu, W., Xu, J., Shan, Y., Lu, S.: Rolling forcing: Autoregressive long video diffusion in real time. In: Proceedings of the International Conference on Learning Representations (ICLR) (2026)
17. Lu, H., Yang, G., Fei, N., Huo, Y., Lu, Z., Luo, P., Ding, M.: Vdt: General-purpose video diffusion transformers via mask modeling. In: Proceedings of the International Conference on Learning Representations (ICLR) (2024)

18. Lu, Y., Zeng, Y., Li, H., Ouyang, H., Wang, Q., Cheng, K.L., Zhu, J., Cao, H., Zhang, Z., Zhu, X., Shen, Y., Zhang, M.: Reward forcing: Efficient streaming video generation with rewarded distribution matching distillation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2026)
19. NVIDIA: Cosmos world foundation model platform for physical ai. arXiv preprint arXiv:2501.03575 (2025)
20. Peebles, W., Xie, S.: Scalable diffusion models with transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2023)
21. Polyak, A., Zohar, A., Brown, A., Tjandra, A., Sinha, A., Lee, A., Vyas, A., Shi, B., Ma, C.Y., Chuang, C.Y., et al.: Movie gen: A cast of media foundation models. arXiv preprint arXiv:2410.13720 (2024)
22. Reda, F., Kontkanen, J., Tabellion, E., Sun, D., Pantofaru, C., Curless, B.: FILM: Frame interpolation for large motion. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 250–266 (2022)
23. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2022)
24. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention (MICCAI) (2015)
25. Su, J., Ahmed, M., Lu, Y., Pan, S., Bo, W., Liu, Y.: Roformer: Enhanced transformer with rotary position embedding. Neurocomputing **568**, 127063 (2024)
26. Tanveer, M., Zhou, Y., Niklaus, S., Amiri, A.M., Zhang, H., Singh, K.K., Zhao, N.: Multicoin: Multi-modal controllable video inbetweening. Computer Graphics Forum (Special Issue of Eurographics) (2026)
27. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems (NeurIPS) (2017)
28. Voleti, V., Jolicoeur-Martineau, A., Pal, C.: Mcvd: Masked conditional video diffusion for prediction, generation, and interpolation. In: Advances in Neural Information Processing Systems (NeurIPS) (2022)
29. Wan-AI: Wan2.1: Text-to-video generation model (2024), `https://github.com/Wan-AI/Wan2.1`, 1 July 2026
30. Wang, W., Yang, Y.: Vidprom: A million-scale real-world video prompt-gallery dataset for text-to-video diffusion models. In: Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks (2024)
31. Wang, X., Zhou, B., Curless, B., Kemelmacher-Shlizerman, I., Holynski, A., Seitz, S.M.: Generative inbetweening: Adapting image-to-video models for keyframe interpolation. In: Proceedings of the International Conference on Learning Representations (ICLR) (2025)
32. Xiao, G., Tian, Y., Chen, B., Han, S., Lewis, M.: Efficient streaming language models with attention sinks. In: Proceedings of the International Conference on Learning Representations (ICLR) (2024)
33. Yang, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Li, C., Liu, D., Huang, F., Wei, H., et al.: Qwen2.5 technical report. arXiv preprint arXiv:2412.15115 (2024)
34. Yang, S., Huang, W., Chu, R., Xiao, Y., Zhao, Y., Wang, X., Li, M., Xie, E., Chen, Y., Lu, Y., Han, S., Chen, Y.: Longlive: Real-time interactive long video generation. In: Proceedings of the International Conference on Learning Representations (ICLR) (2026)

35. Yang, Z., Teng, J., Zheng, W., Ding, M., Huang, S., Xu, J., Yang, Y., Hong, W., Zhang, X., Feng, G., et al.: Cogvideox: Text-to-video diffusion models with an expert transformer. In: Proceedings of the International Conference on Learning Representations (ICLR) (2025)
36. Yesiltepe, H., Meral, T.H.S., Akan, A.K., Oktay, K., Yanardag, P.: Infinity-rope: Action-controllable infinite video generation emerges from autoregressive self-rollout. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2026)
37. Yi, J., Jang, W., Cho, P.H., Nam, J., Yoon, H., Kim, S.: Deep forcing: Training-free long video generation with deep sink and participative compression. In: Proceedings of the International Conference on Machine Learning (ICML) (2026)
38. Yin, T., Gharbi, M., Park, T., Zhang, R., Shechtman, E., Durand, F., Freeman, W.T.: Improved distribution matching distillation for fast image synthesis. In: Advances in Neural Information Processing Systems (NeurIPS) (2024)
39. Yin, T., Gharbi, M., Zhang, R., Shechtman, E., Durand, F., Freeman, W.T., Park, T.: One-step diffusion with distribution matching distillation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2024)
40. Yin, T., Zhang, Q., Zhang, R., Freeman, W.T., Durand, F., Shechtman, E., Huang, X.: From slow bidirectional to fast autoregressive video diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2025)
41. Yu, L., Cheng, Y., Sohn, K., Lezama, J., Zhang, H., Chang, H., Hauptmann, A.G., Yang, M.H., Hao, Y., Essa, I., Jiang, L.: Magvit: Masked generative video transformer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2023)
42. Yu, L., Lezama, J., Gundavarapu, N.B., Versari, L., Sohn, K., Minnen, D., Cheng, Y., Gupta, A., Gu, X., Hauptmann, A.G., et al.: Language model beats diffusion – tokenizer is key to visual generation. In: Proceedings of the International Conference on Learning Representations (ICLR) (2024)
43. Zheng, Z., Peng, X., Yang, T., Shen, C., Li, S., Liu, H., Zhou, Y., Li, T., You, Y.: Open-sora: Democratizing efficient video production for all. arXiv preprint arXiv:2412.20404 (2024)