TinyHistory: Lightweight Video History Embeddings via Two-Stage Context Learning

Lvmin Zhang[1], Shengqu Cai[1], Muyang Li[2], Chong Zeng[1], Beijia Lu[3], Anyi Rao[4], Song Han[2], Gordon Wetzstein[1], and Maneesh Agrawala[1]

[1]Stanford University [2]MIT [3]Carnegie Mellon University [4]HKUST

Abstract. History context is central to autoregressive video generation, driving consistency and storytelling for both commercial models and personal use cases. For example, personal users, offline workflows, and individual-scale finetuning need to encode longer video histories under tight compute and memory budgets. We observe that content and identity consistency is an essential requirement, and that complete, uninterrupted history coverage together with content query and interpretation capabilities is broadly desired. We present TinyHistory, a lightweight history embedding learned through two-stage context learning. In the first stage, we pretrain the encoder on large-scale video data with a randomized frame query objective; in the second stage, we repurpose the pretrained encoder within an autoregressive video diffusion model to learn content-level consistency. As a result, we show that the learned lightweight embeddings achieve consistency comparable (by VLM, VBench, ELO, *etc.*) to heavier alternatives, while reducing training overhead and extending the encodable history length within a given memory budget. We conduct ablation studies to analyze the influence and trade-offs of each component.

Keywords: video generation · autoregressive models · context learning

1 Introduction

Storytelling capability, narrative coherence, and context consistency are essential to the video generation community. The latest commercial models like Sora2 [44], Veo3.1 [19], and Seedance 2.0 [4] enable storyboard-based creation, scene planning, and dynamic camera work. Recent academic models have also focused on long video streaming [26, 68], scene planning [21, 25, 79, 81, 82], and video context consistency [27, 30, 40]. Among these directions, autoregressive models are a key paradigm for video storytelling, supporting native video continuation and storyboard streaming. Autoregressive models treat video history as context, and they face particular challenges when handling long-form content. These challenges affect both commercial-scale systems and personal use cases: for personal users, offline workflows, and individual-scale finetuning, encoding longer video histories is constrained by limited compute and memory.

A range of approaches have been explored to handle long video histories. A naive sliding window, which discards all distant-enough frames, maintains a fixed

context length but loses long-range history. Compressed VAEs like LTXV [22] and DC-AE [10] can produce more compact contexts, and hybrid approaches [1,31,83] like FramePack [74] operate at multiple levels, but at the cost of high-frequency image details. Another strategy is to reduce computation while preserving context length (*e.g.*, using sparse [60,61,73,76] or linear [5,11,33,55,64,70] attention), though the context footprint in memory remains unchanged. Token merging [2,3] demonstrates that a higher merging rate leads to greater detail loss. These efforts indicate that the balance among history coverage, visual fidelity, and memory cost remains an open problem, and that the goal of a lightweight history representation is driven by the demands of different generative applications.

We observe that autoregressive video generation places three demands on a practical history embedding. First, content and identity consistency is an essential requirement: generative models need dense history features to maintain character identity, clothing, and scene layout across steps. Second, maximizing video context continuity needs the history to be represented in a complete and uninterrupted manner, covering the full temporal extent rather than relying on windowed or sparse keyframe sampling. Third, the embedding needs content query and interpretation capabilities, allowing different generation tasks to prioritize their most critical content within a feature manifold of constrained dimension.

We present TinyHistory, a lightweight history embedding learned through a two-stage context learning framework. The two stages are designed for the aforementioned demands: in the first stage, we pretrain the encoder on large-scale video data using a randomized frame query objective, to facilitate query capability and history coverage; in the second stage, we repurpose the pretrained encoder within an autoregressive video diffusion model on natural video data to establish content-level consistency. This approach allows the model to learn from the training data distribution how to prioritize different demands within the constrained feature manifold: *e.g.*, models trained on game footage with repetitive scenes may prioritize scene appearance, while models trained on movies may tend to dedicate more features to face and character consistency.

Several challenges arise in learning such a lightweight history embedding. First, aligning a history representation with the generation model's hidden states requires careful design to avoid degradation and artifacts. We reuse the DiT's hidden manifold: the encoder outputs directly at the DiT's inner hidden states, bypassing the VAE bottleneck, to manipulate deep DiT features instead of the latent space. Second, training with full-length video contexts incurs a large computational cost. We observe that a useful indicator of the embedding's query and interpretation capabilities is its ability to attend to frames at arbitrary temporal positions, leading to a partial supervision strategy: the encoder is pretrained to query frames at randomly sampled time indices, ensuring dense coverage while avoiding processing the entire history at once. After pretraining on millions of videos, the encoder is integrated into the autoregressive system and repurposed for content-level consistency via low-rank adaptation.

Experiments show that the learned lightweight embeddings achieve content consistency, identity preservation, and user preference scores (measured by VLM

evaluation, VBench, and ELO) comparable to heavier alternatives, while reducing training overhead and extending the encodable history length within a given memory budget. We compare with previous baselines and discuss the trade-offs among different approaches.

In summary, we (1) discuss the demands for a lightweight video history embedding in autoregressive generation, and the requirements arising from local and personal use cases; (2) present TinyHistory, a lightweight history embedding learned through a two-stage context learning framework, in which pretraining addresses query capability and history coverage, and joint repurposing within an autoregressive video diffusion model addresses content-level consistency; and (3) conduct extensive experiments across seven baseline paradigms, including quantitative evaluations, user studies, and ablation studies, to analyze the trade-offs of each component.

2 Related Work

Autoregressive video diffusion and long videos. Diffusion has become the driving force behind video synthesis, where a central challenge is length scaling. Training-free length-extension methods [41, 42, 46, 47, 80] reschedule noise or re-balance temporal frequency to stretch pre-trained models beyond their training horizon. A complementary thread blends diffusion with causal prediction: Diffusion Forcing [7] and HistoryGuidance [49] enable variable-horizon conditioning and stable long rollouts by noise injection. These approaches are adapted in industrial systems such as SkyReels-V2 [8] and Magi-1 [52]. Additionally, StreamingT2V [23] augments existing models with short- and long-term memory with randomized blending to generate longer videos. FAR [20] studies long-context AR with flexible RoPE [50] decay and mixed short/long windows, while CausVid [68] distills a bidirectional teacher to a few-step causal generator. StreamDiT [34] combines multi-step distillation with a moving frame buffer and mixed partition training to generate results in real-time. To mitigate error accumulation during AR generation, Self-Forcing [26] simulates AR rollout during training, while its extensions [13, 39, 65] further improve length generalization.

Context learning and history representation. Long-context persistence is crucial as we extend video generation beyond a few seconds. One option is retrieval, which grounds generation in a persistent state. WorldMem [63] and Context-as-Memory [69] augment AR video world models with FoV-based history retrieval, while VMem [37] indexes past views by surfels to retrieve the most relevant viewpoints. Memory Forcing [24] pairs geometry-indexed spatial memory with tailored training regimes to balance exploration and revisits. Beyond fixed retrieval rules, Mixture-of-Contexts [6] learns a dynamic sparse attention context router so that tokens attend only to the most salient chunks. Similarly, MoGA [29] and Holocine [43] also propose token-level sparse attending policies. Pack-and-Force [59] instead proposes a learnable context semantic retriever. Image-level conditioning approaches such as IP-Adapter [67] and its video extensions [32] inject CLIP vision embeddings as cross-attention conditions.

Orthogonal to retrieval, compression of history turns an unbounded context into a compact state. FramePack [74] compresses prior frames into a fixed-size latent "packed" context. Captain Cinema [62] uses a similar compression for keyframes. StateSpaceDiffuser [48] and Po et al. [45] swap quadratic attention for recurrent states to maintain long-term memory. TTTVideo [14] and LaCT [77] use light MLP test-time training layers as a learned context representation.

Efficient video diffusion designs. As we scale video generation to long horizons, large context windows are bottlenecked, driving a wave of efficient computational designs. Kernel advances such as FlashAttention [15, 16] improve throughput. Static or hardware-friendly sparse patterns include sliding/tiling 3D windows [76], radial spatiotemporal masks [38], and training-free head/pruning heuristics [60,66]. Dynamic or learned pruning/routing further select salient token pairs or blocks [58], coarse-to-fine sparse token selection [75], Sage/SpargeAttention families [71–73], blockified routing with cached search [61], and progressive block carving [78]. Another line of work leverages compressing the latent space or token sequence: token merging and patch scaling [2, 36], compact/variable-rate tokenizers [1], highly compressed latent space [22], or multiscale pyramids with re-noising [31]. SANA [64] introduced a linear-attention Diffusion Transformer for images, while SANA-Video [9] extends this with block-linear attention and a constant-memory KV cache.

3 Method

We learn a history encoder $\phi(\cdot)$ through two-stage context learning (Fig. 1). The encoder maps a video history $\boldsymbol{H}$ into a lightweight embedding $\phi(\boldsymbol{H})$ that conditions an autoregressive DiT generator. Section 3.1 details the encoder architecture. In Stage I (Section 3.2), the encoder is pretrained on large-scale video data with a randomized frame query objective. In Stage II (Section 3.3), the encoder is embedded into an autoregressive video diffusion model and repurposed to establish content-level consistency.

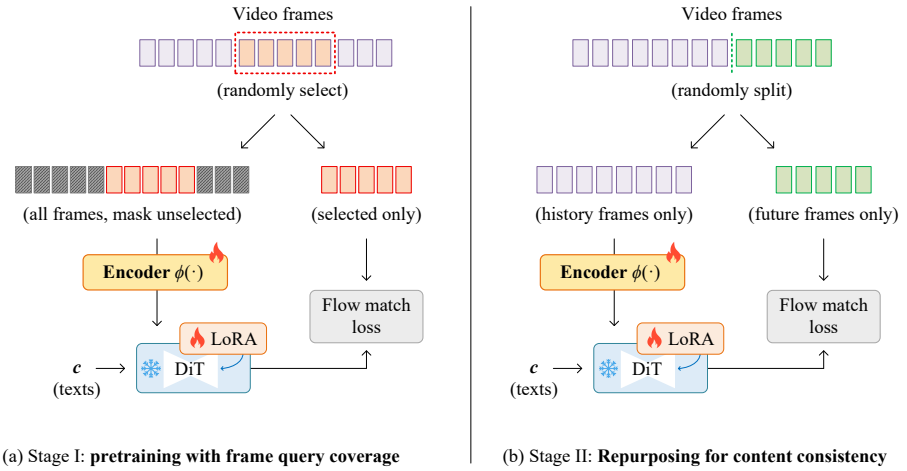
Preliminaries. Unless otherwise noted, all "frames", "pixels", *etc.*, refer to latent concepts. We consider Diffusion Transformers (DiTs) such as Wan [53] and HunyuanVideo [35] with rectified-flow scheduling. The noisy latents $\boldsymbol{X}_{t_i} \in \mathbb{R}^{T\times H\times W\times C}$ are formed by

$$\boldsymbol{X}_{t_i} = (1-t_i)\boldsymbol{X}_0 + t_i\boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}\big(\mathbf{0}, \mathbf{I}\big), \tag{1}$$

from clean latents $\boldsymbol{X}_0$, where $t_i \in (0, 1]$ is the diffusion timestep. In autoregressive generation, the model conditions on a video history $\boldsymbol{H} \in \mathbb{R}^{T_h\times H\times W\times C}$. The generator $\boldsymbol{G}_\theta(\cdot)$ is trained by finding

$$\arg\min_\theta \mathbb{E}_{\boldsymbol{X}_0,\boldsymbol{H},\boldsymbol{c},\boldsymbol{\epsilon},t_i\sim\mathcal{L}(0,1)} \Big\| (\boldsymbol{\epsilon} - \boldsymbol{X}_0) - \boldsymbol{G}_\theta\big(\boldsymbol{X}_{t_i}, t_i, \boldsymbol{c}, \boldsymbol{H}\big) \Big\|_2^2, \tag{2}$$

where $\boldsymbol{c}$ denotes conditions such as text prompts, and $t_i \sim \mathcal{L}(0, 1)$ is the shifted logit-normal distribution [18].



(a) Stage I: **pretraining with frame query coverage** (b) Stage II: **Repurposing for content consistency**

Fig. 1: Overview of TinyHistory. (a) In Stage I, the history encoder $\phi(\cdot)$ is pretrained on large-scale video data: random frame indices $\boldsymbol{\Omega}$ are selected as query targets, and the DiT learns to query these frames conditioned on the encoded history $\phi(\boldsymbol{H})$. (b) In Stage II, the pretrained encoder is integrated into an autoregressive video diffusion model and repurposed jointly (via LoRA) with natural video data to establish content-level consistency. Since the encoder is almost fully convolutional, new segments can be concatenated to the history embedding without recomputation.

3.1 History Encoder Architecture

The history encoder $\phi(\cdot)$ maps a video history $\boldsymbol{H}$ of T_h frames into a history embedding $\phi(\boldsymbol{H})$ that serves as context for the DiT generator. As shown in Fig. 2, the encoder uses a dual-branch design. The first branch processes a low-resolution, low-frame-rate version of the history through the DiT's own VAE, patchifier, and first-layer projection, producing a coarse feature sequence. The second branch takes the original-resolution frames and passes them through a lightweight module (3D convolutions, SiLU activations, and attention layers) to produce a residual feature. The two branches merge via element-wise addition *after* the DiT's first-layer projection. The encoder outputs are connected to the DiT's *inner* hidden states (*e.g.*, 3072 or 5120 channels) rather than being limited to the VAE channel bottleneck (*e.g.*, 4, 16, or 64 channels), which is a key difference between this method and video compression methods like VAEs. Furthermore, an optional cross-attention refinement from the encoder's last hidden states to the DiT hidden states is discussed in ablation studies.

3.2 Stage I: Pretraining with Frame Query Coverage

The first stage pretrains the history encoder $\phi(\cdot)$ on large-scale video data. The pretraining objective is a randomized frame query task that encourages the encoder to learn dense history features covering the full temporal extent.

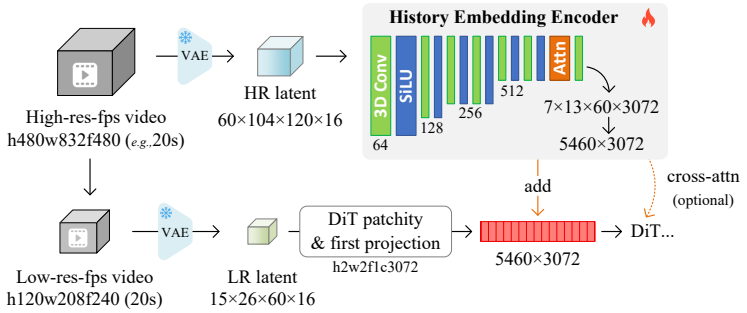


Fig. 2: History encoder architecture. The LR branch reuses the DiT's VAE, patchifier, and first-layer projection on a downsampled input. The HR branch encodes the original-resolution frames through a lightweight module. The two branches are summed after the DiT's first-layer projection, so the output operates in the DiT's inner hidden states rather than passing through the VAE channel bottleneck. Alternative architectures are discussed in ablation studies.

Given a video history $\boldsymbol{H}$, we randomly sample a set of frame indices $\boldsymbol{\Omega}$. Frames at the selected indices are kept clean, while all remaining frames are masked with noise sampled from $\mathcal{L}(0.2, 1)$, resulting in a masked history $\texttt{mask}(\boldsymbol{H}, \boldsymbol{\Omega})$. The DiT learns to query the selected frames via a conditional mapping

$$\boldsymbol{G}_\theta : \{\phi(\texttt{mask}(\boldsymbol{H}, \boldsymbol{\Omega})), \boldsymbol{c}\} \mapsto \boldsymbol{H}_{[\boldsymbol{\Omega}]}, \tag{3}$$

trained by finding

$$\arg\min_{\theta,\phi} \mathbb{E}_{\boldsymbol{H},\boldsymbol{\Omega},\boldsymbol{c},\boldsymbol{\epsilon},t_i} \left\| (\boldsymbol{\epsilon} - \boldsymbol{H}_{[\boldsymbol{\Omega}]}) - \boldsymbol{G}_\theta\big((\boldsymbol{H}_{[\boldsymbol{\Omega}]})_{t_i}, t_i, \boldsymbol{c}, \phi(\texttt{mask}(\boldsymbol{H}, \boldsymbol{\Omega}))\big) \right\|_2^2, \tag{4}$$

where $\boldsymbol{G}_\theta(\cdot)$ is the DiT with low-rank adaptation (LoRA).

The randomness of $\boldsymbol{\Omega}$ is important: if only frames at fixed positions (*e.g.*, the last few frames) are used as targets, the encoder can learn a cheating shortcut by dedicating all capacity to those positions while ignoring the rest of the history. Random selection forces the embedding to maintain dense coverage across the entire temporal extent. Since training on all frames at once incurs considerable overhead for minute-level histories even with efficient infrastructure, this partial supervision through $\boldsymbol{\Omega}$ provides a pretraining strategy that is both scalable and budget-friendly.

3.3 Stage II: Repurposing for Content Consistency

The second stage integrates the pretrained encoder into an autoregressive video diffusion model and repurposes the system jointly to establish content-level consistency. Unlike Stage I, the history $\boldsymbol{H}$ is now complete and noise-free, and no frame indices $\boldsymbol{\Omega}$ are involved. The DiT generator learns to produce the next

video segment $\boldsymbol{X}$ conditioned on the history embedding

$$\boldsymbol{G}_\theta : \{\phi(\boldsymbol{H}), \boldsymbol{c}\} \mapsto \boldsymbol{X}, \tag{5}$$

by the diffusion objective

$$\arg\min_{\theta,\phi} \mathbb{E}_{\boldsymbol{X}_0,\boldsymbol{H},\boldsymbol{c},\boldsymbol{\epsilon},t_i} \left\| (\boldsymbol{\epsilon} - \boldsymbol{X}_0) - \boldsymbol{G}_\theta\big(\boldsymbol{X}_{t_i}, t_i, \boldsymbol{c}, \phi(\boldsymbol{H})\big) \right\|_2^2, \tag{6}$$

where newly initialized LoRA parameters are applied to adapt $\boldsymbol{G}_\theta(\cdot)$.

We note that the encoder $\phi(\cdot)$ is *not* frozen during Stage II: both the encoder and the LoRA-adapted DiT are jointly optimized. This distinguishes our approach from video compression and reconstruction research (*e.g.*, VAEs, Codec, *etc.*), where the encoder is trained once and then locked. Because $\phi(\cdot)$ remains trainable, its features are repurposed from frame-level query targets (Stage I) toward learning content-level consistency (Stage II). The training data distribution then naturally shapes what the encoder prioritizes within the constrained feature manifold: for instance, models trained on game footage may dedicate more capacity to scene appearance, while models trained on cinematic data may prioritize face and character consistency.

3.4 Autoregressive Inference

At inference time, video is generated autoregressively: the model generates a segment $\boldsymbol{X}$, appends it to the history $\boldsymbol{H}$, and generates the next segment conditioned on the updated history embedding. Since the encoder is almost fully convolutional, the embedding for newly appended frames can be computed independently and concatenated to the existing embedding, without reprocessing the full history. This reduces the per-step encoding cost.

4 Experiments

4.1 Experimental Details

Implementation. We conduct Stage I pretraining with 8×H100 GPUs and Stage II LoRA training with 1×H100 or A100. The pipeline uses pure PyTorch with Accelerate, precomputed latents and conditions, and no gradient accumulation. The base models are Wan 2.1 14B [53] and HunyuanVideo 12.8B [35], both flow-match DiTs at 480p resolution. On a single 8×A100-80G node, a smaller model (*e.g.*, Wan 2.1 1.3B) at 480p achieves a batch size of approximately 64 for LoRA training with a window size of 8 (or batch size 32 with window size 16). LoRA rank is 128 for all DiT adaptations. We use Wan 2.1 14B as the default; additional results with HunyuanVideo 12.8B are also reported.

Hyper-parameters. We denote encoding rates as $H{\times}W{\times}T$: for example, 4×4×2 means reducing the latent height by 4×, width by 4×, and temporal length by 2×. The 3D convolution layers first reduce the temporal dimension,

then the spatial dimensions via strided convolutions. Hidden channels follow the pattern $64 \to 128 \to 256 \to 512$ and remain at 512 (*e.g.*, in the attention layer) until a final 1×1 convolution projects the features to the DiT's inner hidden dimension (*e.g.*, 3072 or 5120).

Data preparation. The training set consists of approximately 5 million internet videos, roughly half vertical short-form and half horizontal. After quality filtering, the high-quality portion is captioned with Gemini-2.5-flash, and the remainder with QwenVL [54]. Captions are generated in storyboard format with timestamps; the prompt nearest to each timestamp is used for autoregressive training.

Test set. The test set comprises 1300 storyboard prompts generated by Gemini-2.5-pro and Gemini-3.1-pro, together with 4096 unseen videos independent from the training data. All quantitative metrics are computed on videos generated from these prompts.

4.2 Evaluation Metrics

Gemini-3.1-pro VLM judgements. Inspired by the VLM-as-judge idea in VBench [28], we design four consistency dimensions with custom gate and sub-questions, and query Gemini-3.1-pro with uniformly sampled frames from each video. For each dimension, a yes/no gating question first determines whether the aspect is present; if the gate passes, three yes/no sub-questions probe specific attributes. The per-video score is the fraction of "yes" answers among the three sub-questions (four levels: 0, 0.33, 0.67, 1.0); the dataset score is the mean over all gated-in videos. Full prompt templates are provided in the appendix.

Character consistency (Char). The VLM is first asked "is a clearly visible character present?"; if so, it is further asked whether the character's facial appearance, body structure, and overall identity remain coherent across frames.

Scene consistency (Scene). The VLM is first asked "is a visible background or environment present?"; if so, it is further asked whether the environment layout, lighting conditions, and background details remain stable.

Object consistency (Obj). The VLM is first asked "is a distinct non-person object prominently visible?"; if so, it is further asked whether the object's shape, color and texture, and physical plausibility remain consistent.

Cloth consistency (Cloth). The VLM is first asked "is a character with visible clothing present?"; if so, it is further asked whether the clothing color, style and design, and texture remain consistent.

VBench metrics. We additionally select three algorithmic metrics from VBench [28] that cover evaluation aspects distinct from the above VLM dimensions: text-video alignment, face identity, and motion dynamics.

Semantic alignment (Semantic). The cosine similarity between ViCLIP [56] video and text embeddings, measuring how well the generated content aligns with the text prompt.

Face identity (FaceID). Each frame's ArcFace [17] embedding is compared with the first frame's; the score is the fraction of frames whose cosine similarity exceeds a pre-defined value.

Dynamic degree (Dynamic). RAFT [51] optical flow is used to classify each video as dynamic or static; the score is the fraction of videos classified as dynamic. This serves as a sanity check that history conditioning does not suppress motion.

User preference (ELO). This is pairwise A/B preference tests and Bradley-Terry ELO ratings (base 1200, K=32). Methods with obvious artifacts or severe inconsistency are excluded from the user study to reduce human workloads, marked "/" in the tables.

4.3 Comparison Baselines

We organize baselines by seven architectural paradigms for handling video history, covering the spectrum from naive to learned approaches. For each baseline, we implement the pipeline on the same Wan 2.1 base model and training data to ensure a controlled comparison. Where a method has tunable hyperparameters, we align the context token length across methods for fairer comparison. Detailed CTX/s derivations and implementation details for all methods are provided in the supplementary material.

Sliding window. The most recent 8 latent frames (approximately 2 seconds) are kept as clean latent context; all earlier history is discarded. The DiT directly attends to these frames without any learned encoding (6240 tokens/s).

Patchifier. FramePack [74] packs history frames into a fixed-size latent context by enlarging the patchifying projection kernel, so that temporally distant frames receive coarser spatial patches (2340 tokens†).

Image planning + I2V. QwenEdit [57] takes the last K generated keyframes as input and produces the next keyframe; all keyframes are generated first, then each is animated by Wan 2.1 I2V and concatenated. We test K=1 and K=2; the history context is sparse (keyframe images only, no dense coverage).

Retrieval-based memory. Resembling WorldMem [63] and Context-as-Memory [69], this baseline estimates camera poses for all generated frames via COLMAP, predicts the next camera field-of-view (FOV) along the trajectory, and retrieves the history frames whose FOV is closest as context (1560 tokens/s; full history pool 4680 tokens†).

Image embeddings. Resembling IP-Adapter [67] and video-IP-Adapter [32], this baseline uniformly samples 2 frames per second from the history, extracts a CLIP Vision embedding for each, and injects the concatenated embeddings into the DiT via a projection layer (∼512 tokens/s).

Feature compression. Resembling APT [12] and ToMe [2], this baseline computes each history latent token's cosine similarity with its six $T\times H\times W$ neighbors (up, down, left, right, front, back); tokens exceeding a threshold α are removed and their features projected into the remaining neighbors (∼512 tokens/s). The operation is mathematically equivalent to a large-kernel convolution.

External VAE. The LTX-Video VAE [22] (see also DCAE [10]) encodes the video history at a higher compression rate than Wan 2.1's native VAE; a new randomly initialized linear projection then learns to map the compressed latent

Table 1: Comparison with baselines on content consistency. CTX/s denotes the history context length per second on Wan 2.1 at 480p. Best results in **bold**, second best underlined. "†": CTX covers the entire history rather than per second. "/": Excluded from user study due to severe artifacts.

			Gemini-3.1-pro VLM ↑				VBench ↑			User ↑
Method	Paradigm	CTX/s	Char	Scene	Obj	Cloth	Semantic	FaceID	Dynamic	ELO
Sliding Window	Naive	6240	46.65	57.02	48.52	45.00	22.55	57.87	**90.61**	1090
FramePack [74]	Patchifier	2340†	57.92	76.04	67.09	63.83	23.65	62.01	88.16	1182
QwenEdit [57]+I2V (K=1)	Image planning	Sparse	58.60	65.87	62.80	58.09	23.61	62.21	79.14	/
QwenEdit [57]+I2V (K=2)	Image planning	Sparse	74.67	81.81	72.61	75.54	25.46	68.81	79.49	1205
FOV Retrieval (resem. [63, 69])	Retrieval-based Memory	1560 (4680†)	55.80	**88.17**	59.38	55.83	21.68	65.78	83.29	/
CLIP Vision (resem. [32, 67])	Image Embeddings	512	49.06	60.37	53.59	**93.73**	22.17	59.89	83.57	/
Token Merging (resem. [2, 12])	Feature Compression	∼512	51.09	63.73	55.30	48.09	23.14	60.61	83.96	/
LTX-VAE [22]+Proj	External VAE	780	62.70	69.72	64.69	66.82	22.98	67.01	82.17	1147
TinyHistory (4×4×2)	Two-stage Learning	195	80.19	72.21	74.38	77.40	25.74	70.71	84.31	1262
TinyHistory (2×2×2)	Two-stage Learning	780	**85.90**	87.11	**85.45**	87.74	**26.84**	**73.05**	86.41	**1332**

to the DiT's hidden dimension. An alternative is to adopt LTX-Video directly as the base model, but a fair comparison would then require all methods to use the same LTX-Video backbone; this is discussed in the supplementary material.

4.4 Results

Comparison with baselines. Three baselines each top exactly one dimension in Table 1 — FOV Retrieval in Scene (88.17), CLIP Vision in Cloth (93.73), Sliding Window in Dynamic (90.61) — but all three suffer non-trivial degradation on multiple dimensions. TinyHistory (2×2×2) leads Char, Obj, Semantic, FaceID, and ELO, and ranks second in Scene and Cloth, at 780 tokens/s. The Sliding Window retains full-length latent (6240 tokens/s) yet reports the lowest Char (46.65) and FaceID (57.87), suggesting that longer temporal coverage is important for these metrics. At the same 780 tokens/s, TinyHistory (2×2×2) reports higher scores than LTX-VAE+Proj across all dimensions (Char 85.90 vs 62.70): learning projections for adapting external VAE latent space can cause local minima and quality degradation, and TinyHistory's encoder bypasses the latent channel bottleneck of the external VAE by directly connecting to the DiT inner hidden states. TinyHistory (4×4×2) at 195 tokens/s still leads all external baselines in five dimensions, at the cost of Scene (72.21, rank 5) and Cloth (77.40, rank 3) with 4× spatial rate; the gap with 2×2×2 (Scene 72.21 vs 87.11) illustrates the encoding rate trade-off examined in Section 4.5.

Equivalent per-second context length. CTX/s in Table 1 denotes the number of history context tokens per second on Wan 2.1 at 480p (derivations in the supplementary material). The full-length baseline is 6240 tokens/s. Methods whose context does not grow with video length are marked "†": FramePack packs all history into a fixed 2340-token context, and FOV Retrieval retrieves 1560 tokens/s with the full history clamped to 4680 tokens. Image planning methods (QwenEdit) operate on sparse keyframe images and are marked "Sparse" as their cost is not directly comparable in token units. TinyHistory (4×4×2) operates at

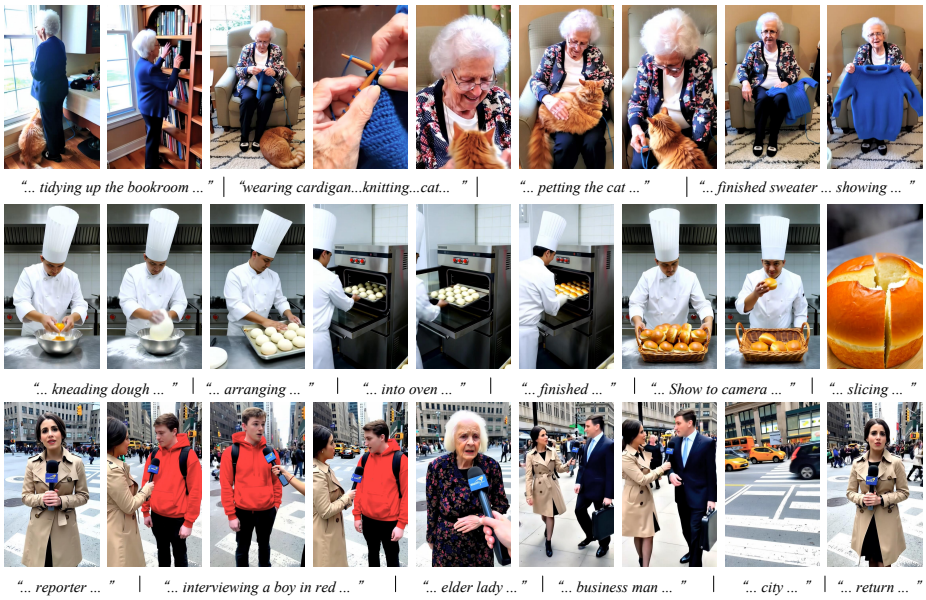


Fig. 3: Qualitative results on storyboard generation. Each row shows frames sampled from an autoregressively generated video driven by a storyboard of text prompts. This batch of results uses HunyuanVideo as base model. The history encoder maintains character identity, clothing, and scene layout across shots with diverse prompts. Storyboards are written by external language models.

195 tokens/s, a 32× reduction from the full-length baseline, while TinyHistory (2×2×2) at 780 tokens/s matches the per-second cost of LTX-VAE+Proj.

Qualitative results. Figure 3 shows storyboard-driven generation results, where each row is an autoregressively generated video driven by a sequence of text prompts. Across diverse prompts that introduce new activities, camera angles, and environments, this method preserves character identity, clothing, and scene layout across shots.

4.5 Ablation Studies

Table 2 presents ablation results addressing four design questions: (1) is Stage I pretraining necessary? (2) must the encoder be trainable in Stage II? (3) are both branches necessary? (4) how does encoding rate affect consistency?

Stage I: Influence of pretraining. Without Stage I pretraining, only Stage II with random initialization is insufficient to learn effective history representations due to local minima and difficulty in converging. Char drops from 80.19 to 52.64 and FaceID from 70.71 to 58.38. Dynamic rises to 95.69, the highest in Table 2; weaker history conditioning imposes less constraint on motion, a consistency-dynamics trade-off visible across all degraded variants. Figure 4 shows that the

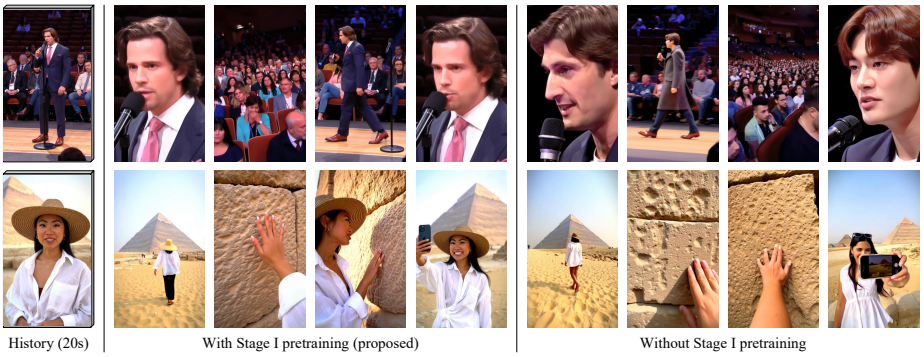


Fig. 4: Influence of Stage I pretraining on generation consistency. Given the same video history, the model with Stage I pretraining (left) preserves facial features, clothing, scene layout, and camera coordination, while the model without pretraining (right) fails to attend to relevant history content, resulting in identity and object inconsistency. Both models are trained for 100k steps.

Table 2: Ablation study. All variants use Wan 2.1 as the base model and are trained for 100k steps. CTX/s denotes the history context length per second on Wan 2.1 at 480p. Best results in **bold**, second best underlined.

		Gemini-3.1-pro VLM ↑				VBench ↑			User ↑
Variant	CTX/s	Char	Scene	Obj	Cloth	Semantic	FaceID	Dynamic	ELO
Without Stage I (4×4×2)	195	52.64	52.84	52.53	50.37	20.50	58.38	**95.69**	/
Frozen Encoder (4×4×2)	195	61.42	62.69	58.33	58.39	22.31	61.82	92.00	/
Only LR (4×4×2)	195	74.29	65.06	67.03	76.12	23.55	67.28	89.17	1223
Without LR (4×4×2)	195	67.73	58.01	64.57	54.86	22.70	65.36	90.72	/
Proposed (4×4×2)	195	80.19	72.21	74.38	77.40	25.74	70.71	84.31	1262
Proposed (2×2×2)	780	85.90	87.11	**85.45**	**87.74**	**26.84**	**73.05**	86.41	1332
Proposed (2×2×4)	390	90.24	81.02	77.06	79.35	25.14	71.59	85.04	1281
Proposed (2×2×1)	1560	**93.78**	**92.59**	84.56	85.54	26.01	72.20	88.90	**1342**

model without pretraining exhibits identity and consistency degradation, while the pretrained model maintains consistent characters and scenes.

Stage II: Frozen vs. trainable encoder. Freezing the encoder after Stage I — treating it as a locked encoder — drops Char from 80.19 to 61.42, indicating that Stage I features must be repurposed through joint Stage II training. *Frozen Encoder* ranks between *Without Stage I* and *Only LR* (Char $52.64 < 61.42 < 74.29$): pretrained initialization contributes but is insufficient without adaptation. Dynamic remains high at 92.00, consistent with the pattern that weaker consistency signals yield freer motion.

Branch ablation. Removing the LR branch (*Without LR*, Char 67.73) degrades more than removing the HR branch (*Only LR*, Char 74.29), suggesting that DiT manifold alignment through the LR path may contribute more to the overall feature structure. *Without LR* also exhibits Dynamic 90.72: degraded history embeddings impose less constraint on motion, yielding randomly dynamic

Table 3: Extensions and compatibility. Each row adds one extension to the base TinyHistory (4×4×2). All extensions are orthogonal and can be combined.

	Gemini-3.1-pro VLM ↑			VBench ↑	
Configuration	Char	Scene	Obj	Semantic	FaceID
Proposed (4×4×2)	80.19	72.21	74.38	25.74	70.71
+ Sliding Window	85.75	**82.51**	78.41	25.75	71.94
+ Cross-Attention	87.01	75.32	80.75	25.54	72.72
+ Multiple Encoders	**88.25**	79.50	**83.33**	**25.91**	**73.37**
+ KV-Cache	75.80	65.28	70.51	24.89	63.61

transitions with poor consistency. The full dual-branch design (Char 80.19) improves over both, indicating the two paths provide complementary features.

Encoding rate. The rate sweep is largely monotonic, but with diminishing returns: 2×2×2 at 780 tokens/s achieves comparable quality to 2×2×1 at 1560 tokens/s (ELO 1332 vs 1342) at half the context cost. 4×4×2 vs 2×2×4 (195 vs 390 tokens/s) isolates spatial from temporal rate: 2×2×4 reports Char 90.24 vs 80.19, indicating that 4× spatial rate discards more identity-relevant detail than 4× temporal rate.

4.6 Extensions and Compatibility

The history embedding design is modular and compatible with several orthogonal enhancements. Table 3 reports quantitative results when each extension is added on top of the base TinyHistory (4×4×2) configuration.

Sliding window. Adding a 3-frame latent overlap (∼0.75 seconds) between consecutive generation steps reduces the chance of camera shot shifts when the DiT receives only the history embedding as context. Scene consistency increases by 10.3 points (72.21→82.51), the largest single-extension gain in Table 3; Char and Obj also increase by 5.6 and 4.0 points respectively (Fig. 5).

Cross-attention enhancement. Adding an extra cross-attention connection from the encoder's last hidden states to the DiT layers allows performing spatial queries over history features and improves fine-grained consistency. Obj increases by 6.4 points (74.38→80.75) and Char by 6.8 points, at the cost of additional computation per DiT block (Fig. 6).

Multiple encoders. A second encoder with a complementary encoding pattern (2×2×8, prioritizing spatial detail over temporal coverage) provides measurable gains: Char +8.1, Obj +9.0, FaceID +2.7 (Fig. 7). The cost is a longer context (195+195 = 390 tokens/s total), as the two encoders' embeddings are concatenated before appending to the DiT context.

KV-Cache compatibility. Because the history embedding is fixed once encoded, its key-value pairs can be cached across diffusion steps, avoiding redundant attention computation. This trades a quality decrease (FaceID −7.1, Scene −6.9) for inference speedup, and is orthogonal to causal inference acceleration methods such as CausVid [68] and Self-Forcing [26].

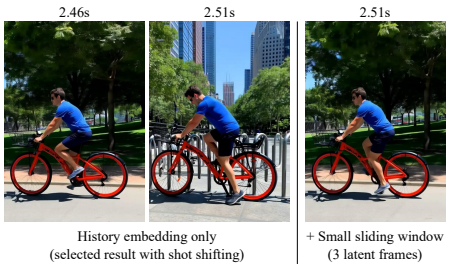


Fig. 5: Adding a small sliding window (3 latent frames). Without the sliding window, consecutive generation steps may exhibit camera shot shifts. The sliding window provides a short overlap that encourages temporal continuity.

Fig. 6: Adding cross-attention from the history encoder's last hidden states to the DiT. In detail-oriented scenarios (*e.g.*, the items on shelves), cross-attention provides additional consistency at the cost of increased computation.

Fig. 7: Using two history encoders with different encoding patterns ($4\times4\times2$ and $2\times2\times8$). The second encoder prioritizes spatial detail over temporal coverage, enabling fine-grained features such as text on newspapers to be maintained across generation steps, at the cost of a longer context.

5 Conclusion

This paper presents TinyHistory, a two-stage context learning framework for lightweight video history embeddings in autoregressive generation. In Stage I, the encoder is pretrained with a randomized frame query objective on large-scale video data to establish dense history coverage; in Stage II, the encoder is jointly repurposed with the DiT generator on natural video data to learn content-level consistency. Experiments on Wan 2.1 and HunyuanVideo show that this method leads five of eight evaluation dimensions (Char, Obj, Semantic, FaceID, ELO) and ranks second in two others (Scene and Cloth), using shorter context than heavier alternatives. Ablation studies confirm that pretraining, joint encoder training, and dual-branch design each contribute to the final quality, and that the encoding rate trade-off follows a diminishing-returns pattern. The architecture is modular: sliding window overlap, cross-attention, multiple encoders, and KV-caching can each be added independently.

Acknowledgements

This work was partially supported by the Brown Institute for Media Innovation, by Google through their affiliation with Stanford Institute for Human-centered Artificial Intelligence (HAI) and by a Hoffman-Yee HAI grant.

References

1. Bachmann, R., Allardice, J., Mizrahi, D., Fini, E., Kar, O.F., Amirloo, E., El-Nouby, A., Zamir, A., Dehghan, A.: Flextok: Resampling images into 1d token sequences of flexible length. In: arXiv (2025)
2. Bolya, D., Fu, C.Y., Dai, X., Zhang, P., Feichtenhofer, C., Hoffman, J.: Token merging: Your ViT but faster. In: International Conference on Learning Representations (2023)
3. Bolya, D., Hoffman, J.: Token merging for fast stable diffusion. CVPR Workshop on Efficient Deep Learning for Computer Vision (2023)
4. ByteDance: Seedance 2.0 (2026), https://seed.bytedance.com/en/seedance2_0, accessed on February 28, 2026
5. Cai, H., Li, J., Hu, M., Gan, C., Han, S.: Efficientvit: Lightweight multi-scale attention for high-resolution dense prediction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 17302–17313 (2023)
6. Cai, S., Yang, C., Zhang, L., Guo, Y., Xiao, J., Yang, Z., Xu, Y., Yang, Z., Yuille, A., Guibas, L., Agrawala, M., Jiang, L., Wetzstein, G.: Mixture of contexts for long video generation. In: arXiv (2025)
7. Chen, B., Martí Monsó, D., Du, Y., Simchowitz, M., Tedrake, R., Sitzmann, V.: Diffusion forcing: Next-token prediction meets full-sequence diffusion. Advances in Neural Information Processing Systems **37**, 24081–24125 (2025)
8. Chen, G., Lin, D., Yang, J., Lin, C., Zhu, J., Fan, M., Zhang, H., Chen, S., Chen, Z., Ma, C., Xiong, W., Wang, W., Pang, N., Kang, K., Xu, Z., Jin, Y., Liang, Y., Song, Y., Zhao, P., Xu, B., Qiu, D., Li, D., Fei, Z., Li, Y., Zhou, Y.: Skyreels-v2: Infinite-length film generative model (2025), https://arxiv.org/abs/2504.13074
9. Chen, J., Zhao, Y., Yu, J., Chu, R., Chen, J., Yang, S., Wang, X., Pan, Y., Zhou, D., Ling, H., et al.: Sana-video: Efficient video generation with block linear diffusion transformer. In: arXiv (2025)
10. Chen, J., Cai, H., Chen, J., Xie, E., Yang, S., Tang, H., Li, M., Lu, Y., Han, S.: Deep compression autoencoder for efficient high-resolution diffusion models. arXiv preprint arXiv:2410.10733 (2024)
11. Choromanski, K., Likhosherstov, V., Dohan, D., Song, X., Gane, A., Sarlos, T., Hawkins, P., Davis, J., Mohiuddin, A., Kaiser, L., et al.: Rethinking attention with performers. arXiv preprint arXiv:2009.14794 (2020)
12. Choudhury, R., Kim, J., Park, J., Yang, E., Jeni, L.A., Kitani, K.M.: Accelerating vision transformers with adaptive patch sizes. arXiv preprint arXiv:2510.18091 (2025)
13. 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: arXiv (2025)
14. Dalal, K., Koceja, D., Hussein, G., Xu, J., Zhao, Y., Song, Y., Han, S., Cheung, K.C., Kautz, J., Guestrin, C., Hashimoto, T., Koyejo, S., Choi, Y., Sun, Y., Wang, X.: One-minute video generation with test-time training (2025), https://arxiv.org/abs/2504.05298

15. Dao, T.: FlashAttention-2: Faster attention with better parallelism and work partitioning. In: ICLR (2024)
16. Dao, T., Fu, D.Y., Ermon, S., Rudra, A., Ré, C.: Flashattention: Fast and memory-efficient exact attention with io-awareness (2022), https://arxiv.org/abs/2205.14135
17. Deng, J., Guo, J., Xue, N., Zafeiriou, S.: Arcface: Additive angular margin loss for deep face recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4690–4699 (2019)
18. Esser, P., Kulal, S., Blattmann, A., Entezari, R., Müller, J., Saini, H., Levi, Y., Lorenz, D., Sauer, A., Boesel, F., et al.: Scaling rectified flow transformers for high-resolution image synthesis. In: Forty-first International Conference on Machine Learning (2024)
19. Google: Veo3.1 (2025), accessed on November 9, 2025
20. Gu, Y., Mao, W., Shou, M.Z.: Long-context autoregressive video modeling with next-frame prediction (2025), https://arxiv.org/abs/2503.19325
21. Guo, Y., Yang, C., Yang, Z., Ma, Z., Lin, Z., Yang, Z., Lin, D., Jiang, L.: Long context tuning for video generation. In: ICCV (2025)
22. HaCohen, Y., Chiprut, N., Brazowski, B., Shalem, D., Moshe, D., Richardson, E., Levin, E., Shiran, G., Zabari, N., Gordon, O., et al.: Ltx-video: Realtime video latent diffusion. arXiv preprint arXiv:2501.00103 (2024)
23. Henschel, R., Khachatryan, L., Hayrapetyan, D., Poghosyan, H., Tadevosyan, V., Wang, Z., Navasardyan, S., Shi, H.: Streamingt2v: Consistent, dynamic, and extendable long video generation from text. arXiv preprint arXiv:2403.14773 (2024)
24. Huang, J., Hu, X., Han, B., Shi, S., Tian, Z., He, T., Jiang, L.: Memory forcing: Spatio-temporal memory for consistent scene generation on minecraft. In: arXiv (2025)
25. Huang, L., Wang, W., Wu, Z.F., Shi, Y., Dou, H., Liang, C., Feng, Y., Liu, Y., Zhou, J.: In-context lora for diffusion transformers. arXiv preprint arXiv:2410.23775 (2024)
26. Huang, X., Li, Z., He, G., Zhou, M., Shechtman, E.: Self forcing: Bridging the train-test gap in autoregressive video diffusion. In: arXiv (2025)
27. Huang, Y., Yuan, Z., Liu, Q., Wang, Q., Wang, X., Zhang, R., Wan, P., Zhang, D., Gai, K.: Conceptmaster: Multi-concept video customization on diffusion transformer models without test-time tuning. arXiv preprint arXiv:2501.04698 (2025)
28. Huang, Z., He, Y., Yu, J., Zhang, F., Si, C., Jiang, Y., Zhang, Y., Wu, T., Jin, Q., Chanpaisit, N., Wang, Y., Chen, X., Wang, L., Lin, D., Qiao, Y., Liu, Z.: VBench: Comprehensive benchmark suite for video generative models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2024)
29. Jia, W., Lu, Y., Huang, M., Wang, H., Huang, B., Chen, N., Liu, M., Jiang, J., Mao, Z.: Moga: Mixture-of-groups attention for end-to-end long video generation. In: arXiv (2025)
30. Jiang, Y., Wu, T., Yang, S., Si, C., Lin, D., Qiao, Y., Loy, C.C., Liu, Z.: Videobooth: Diffusion-based video generation with image prompts. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6689–6700 (2024)
31. Jin, Y., Sun, Z., Li, N., Xu, K., Xu, K., Jiang, H., Zhuang, N., Huang, Q., Song, Y., Mu, Y., Lin, Z.: Pyramidal flow matching for efficient video generative modeling (2024)
32. kaaskoek232: IP-Adapter for Wan Video. https://github.com/kaaskoek232/IPAdapterWAN (2025), accessed: 2026-03-05

33. Katharopoulos, A., Vyas, A., Pappas, N., Fleuret, F.: Transformers are rnns: Fast autoregressive transformers with linear attention. In: International conference on machine learning. pp. 5156–5165. PMLR (2020)
34. Kodaira, A., Hou, T., Hou, J., Tomizuka, M., Zhao, Y.: Streamdit: Real-time streaming text-to-video generation. In: arXiv (2025)
35. Kong, W., Tian, Q., Zhang, Z., Min, R., Dai, Z., Zhou, J., Xiong, J., Li, X., Wu, B., Zhang, J., et al.: Hunyuanvideo: A systematic framework for large video generative models. arXiv preprint arXiv:2412.03603 (2024)
36. Lee, S.H., Wang, J., Zhang, Z., Fan, D., Li, X.: Video token merging for long-form video understanding. In: arXiv (2024)
37. Li, R., Torr, P., Vedaldi, A., Jakab, T.: Vmem: Consistent interactive video scene generation with surfel-indexed view memory. In: ICCV (2025)
38. Li, X., Li, M., Cai, T., Xi, H., Yang, S., Lin, Y., Zhang, L., Yang, S., Hu, J., Peng, K., Agrawala, M., Stoica, I., Keutzer, K., Han, S.: Radial attention: $\mathcal{O}(n \log n)$ sparse attention with energy decay for long video generation. In: arXiv (2025)
39. Liu, K., Hu, W., Xu, J., Shan, Y., Lu, S.: Rolling forcing: Autoregressive long video diffusion in real time. In: arXiv (2025)
40. Long, F., Qiu, Z., Yao, T., Mei, T.: Videostudio: Generating consistent-content and multi-scene videos (2024), https://arxiv.org/abs/2401.01256
41. Lu, Y., Liang, Y., Zhu, L., Yang, Y.: Freelong: Training-free long video generation with spectralblend temporal attention. Advances in Neural Information Processing Systems **37**, 131434–131455 (2025)
42. Lu, Y., Yang, Y.: Freelong++: Training-free long video generation via multi-band spectralfusion. In: arXiv (2025)
43. Meng, Y., Ouyang, H., Yu, Y., Wang, Q., Wang, W., Cheng, K.L., Wang, H., Li, Y., Chen, C., Zeng, Y., Shen, Y., Qu, H.: Holocine: Holistic generation of cinematic multi-shot long video narratives. In: arXiv (2025)
44. OpenAI: Sora2 (2025), accessed on November 8, 2025
45. Po, R., Nitzan, Y., Zhang, R., Chen, B., Dao, T., Shechtman, E., Wetzstein, G., Huang, X.: Long-context state-space video world models. In: ICCV (2025)
46. Qiu, H., Xia, M., Zhang, Y., He, Y., Wang, X., Shan, Y., Liu, Z.: Freenoise: Tuning-free longer video diffusion via noise rescheduling. arXiv preprint arXiv:2310.15169 (2023)
47. Ruhe, D., Heek, J., Salimans, T., Hoogeboom, E.: Rolling diffusion models (2024)
48. Savov, N., Kazemi, N., Zhang, D., Paudel, D.P., Wang, X., Gool, L.V.: Statespaced-iffuser: Bringing long context to diffusion world models. In: NeurIPS (2025)
49. Song, K., Chen, B., Simchowitz, M., Du, Y., Tedrake, R., Sitzmann, V.: History-guided video diffusion. arXiv preprint arXiv:2502.06764 (2025)
50. Su, J., Ahmed, M., Lu, Y., Pan, S., Bo, W., Liu, Y.: Roformer: Enhanced transformer with rotary position embedding. Neurocomputing **568**, 127063 (2024)
51. Teed, Z., Deng, J.: Raft: Recurrent all-pairs field transforms for optical flow (2020), https://arxiv.org/abs/2003.12039
52. Teng, H., Jia, H., Sun, L., Li, L., Li, M., Tang, M., Han, S., Zhang, T., Luo, W., Sun, Y., Cao, Y., Huang, Y., Lin, Y., Fang, Y., Tao, Z., Zhang, Z., et al.: Magi-1: Autoregressive video generation at scale. In: arXiv (2025)
53. Wang, A., Ai, B., Wen, B., Mao, C., Xie, C.W., Chen, D., Yu, F., Zhao, H., Yang, J., Zeng, J., Wang, J., Zhang, J., Zhou, J., Wang, J., Chen, J., Zhu, K., Zhao, K., Yan, K., Huang, L., Feng, M., Zhang, N., Li, P., Wu, P., Chu, R., Feng, R., Zhang, S., Sun, S., Fang, T., Wang, T., Gui, T., Weng, T., Shen, T., Lin, W., Wang, W., Wang, W., Zhou, W., Wang, W., Shen, W., Yu, W., Shi, X., Huang, X., Xu, X.,

Kou, Y., Lv, Y., Li, Y., Liu, Y., Wang, Y., Zhang, Y., Huang, Y., Li, Y., Wu, Y., Liu, Y., Pan, Y., Zheng, Y., Hong, Y., Shi, Y., Feng, Y., Jiang, Z., Han, Z., Wu, Z.F., Liu, Z.: Wan: Open and advanced large-scale video generative models. In: arXiv (2025)
54. Wang, P., Bai, S., Tan, S., Wang, S., Fan, Z., Bai, J., Chen, K., Liu, X., Wang, J., Ge, W., et al.: Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. arXiv preprint arXiv:2409.12191 (2024)
55. Wang, S., Li, B.Z., Khabsa, M., Fang, H., Ma, H.: Linformer: Self-attention with linear complexity. arXiv preprint arXiv:2006.04768 (2020)
56. Wang, Y., He, Y., Li, Y., Li, K., Yu, J., Ma, X., Li, X., Chen, G., Chen, X., Wang, Y., et al.: Internvid: A large-scale video-text dataset for multimodal understanding and generation. In: The Twelfth International Conference on Learning Representations (2023)
57. Wu, C., Li, J., Zhou, J., Lin, J., Gao, K., Yan, K., ming Yin, S., Bai, S., Xu, X., Chen, Y., Chen, Y., Tang, Z., Zhang, Z., Wang, Z., Yang, A., Yu, B., Cheng, C., Liu, D., Li, D., Zhang, H., Meng, H., Wei, H., Ni, J., Chen, K., Cao, K., Peng, L., Qu, L., Wu, M., Wang, P., Yu, S., Wen, T., Feng, W., Xu, X., Wang, Y., Zhang, Y., Zhu, Y., Wu, Y., Cai, Y., Liu, Z.: Qwen-image technical report (2025), https://arxiv.org/abs/2508.02324
58. Wu, J., Hou, L., Yang, H., Tao, X., Tian, Y., Wan, P., Zhang, D., Tong, Y.: Vmoba: Mixture-of-block attention for video diffusion models. In: arXiv (2025)
59. Wu, X., Zhang, G., Xu, Z., Zhou, Y., Lu, Q., He, X.: Pack and force your memory: Long-form and consistent video generation. In: arXiv (2025)
60. Xi, H., Yang, S., Zhao, Y., Xu, C., Li, M., Li, X., Lin, Y., Cai, H., Zhang, J., Li, D., et al.: Sparse videogen: Accelerating video diffusion transformers with spatial-temporal sparsity. arXiv preprint arXiv:2502.01776 (2025)
61. Xia, Y., Ling, S., Fu, F., Wang, Y., Li, H., Xiao, X., Cui, B.: Training-free and adaptive sparse attention for efficient long video generation (2025)
62. Xiao, J., Yang, C., Zhang, L., Cai, S., Zhao, Y., Guo, Y., Wetzstein, G., Agrawala, M., Yuille, A., Jiang, L.: Captain cinema: Towards short movie generation. In: arXiv (2025)
63. Xiao, Z., Lan, Y., Zhou, Y., Ouyang, W., Yang, S., Zeng, Y., Pan, X.: Worldmem: Long-term consistent world simulation with memory. In: arXiv (2025)
64. Xie, E., Chen, J., Chen, J., Cai, H., Tang, H., Lin, Y., Zhang, Z., Li, M., Zhu, L., Lu, Y., et al.: Sana: Efficient high-resolution image synthesis with linear diffusion transformers. arXiv preprint arXiv:2410.10629 (2024)
65. 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: arXiv (2025)
66. Yang, S., Xi, H., Zhao, Y., Li, M., Zhang, J., Cai, H., Lin, Y., Li, X., Xu, C., Peng, K., et al.: Sparse videogen2: Accelerate video generation with sparse attention via semantic-aware permutation. In: NeurIPS (2025)
67. Ye, H., Zhang, J., Liu, S., Han, X., Yang, W.: Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. arXiv preprint arXiv:2308.06721 (2023)
68. Yin, T., Zhang, Q., Zhang, R., Freeman, W.T., Durand, F., Shechtman, E., Huang, X.: From slow bidirectional to fast causal video generators. arXiv preprint arXiv:2412.07772 (2024)
69. Yu, J., Bai, J., Qin, Y., Liu, Q., Wang, X., Wan, P., Zhang, D., Liu, X.: Context as memory: Scene-consistent interactive long video generation with memory retrieval. In: arXiv (2025)

70. Yu, W., Luo, M., Zhou, P., Si, C., Zhou, Y., Wang, X., Feng, J., Yan, S.: Metaformer is actually what you need for vision. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10819–10829 (2022)
71. Zhang, J., Huang, H., Zhang, P., Wei, J., Zhu, J., Chen, J.: Sageattention2: Efficient attention with thorough outlier smoothing and per-thread int4 quantization. In: ICML (2025)
72. Zhang, J., Wei, J., Zhang, P., Zhu, J., Chen, J.: Sageattention: Accurate 8-bit attention for plug-and-play inference acceleration. In: International Conference on Learning Representations (ICLR) (2025)
73. Zhang, J., Xiang, C., Huang, H., Wei, J., Xi, H., Zhu, J., Chen, J.: Spargeattn: Accurate sparse attention accelerating any model inference. In: ICML (2025)
74. Zhang, L., Agrawala, M.: Packing input frame contexts in next-frame prediction models for video generation. In: arXiv (2025)
75. Zhang, P., Chen, Y., Huang, H., Lin, W., Liu, Z., Stoica, I., Xing, E., Zhang, H.: Vsa: Faster video diffusion with trainable sparse attention. In: arXiv (2025)
76. Zhang, P., Chen, Y., Su, R., Ding, H., Stoica, I., Liu, Z., Zhang, H.: Fast video generation with sliding tile attention. In: arXiv (2025)
77. Zhang, T., Bi, S., Hong, Y., Zhang, K., Luan, F., Yang, S., Sunkavalli, K., Freeman, W.T., Tan, H.: Test-time training done right. In: arXiv (2025)
78. Zhang, Y., Xing, J., Xia, B., Liu, S., Peng, B., Tao, X., Wan, P., Lo, E., Jia, J.: Training-free efficient video generation via dynamic token carving. In: arXiv (2025)
79. Zhao, C., Liu, M., Wang, W., Chen, W., Wang, F., Chen, H., Zhang, B., Shen, C.: Moviedreamer: Hierarchical generation for coherent long visual sequence. arXiv preprint arXiv:2407.16655 (2024)
80. Zhao, M., He, G., Chen, Y., Zhu, H., Li, C., Zhu, J.: Riflex: A free lunch for length extrapolation in video diffusion transformers. In: arXiv (2025)
81. Zheng, M., Xu, Y., Huang, H., Ma, X., Liu, Y., Shu, W., Pang, Y., Tang, F., Chen, Q., Yang, H., et al.: Videogen-of-thought: A collaborative framework for multi-shot video generation. arXiv preprint arXiv:2412.02259 (2024)
82. Zhou, Y., Zhou, D., Cheng, M.M., Feng, J., Hou, Q.: Storydiffusion: Consistent self-attention for long-range image and video generation. Advances in Neural Information Processing Systems **37**, 110315–110340 (2024)
83. Zhou, Z., Yang, Y., Yang, Y., He, T., Peng, H., Qiu, K., Dai, Q., Qiu, L., Luo, C., Liu, L.: Hitvideo: Hierarchical tokenizers for enhancing text-to-video generation with autoregressive large language models. arXiv preprint arXiv:2503.11513 (2025)