TOWARDS END-TO-END PRETRAINING OF STREAMING VIDEO MODELS FROM SPARSE CAUSAL HISTORIES

Anonymous authors
Paper under double-blind review

ABSTRACT

We propose Sparse Predictive Transformers (SPT), a simple framework for efficient, end-to-end pretraining of streaming video generators and video representations. A single flow loss jointly trains a ViT, a Causal Transformer, and a DiT to predict a complete chunk sampled from a short future horizon using sparse history features. Streaming generation reuses the history state across denoising steps. At 355M and 775M parameters, SPT achieves lower FVD than teacher-forcing and diffusion-forcing baselines with roughly half their training compute and lower streaming latency. Generative pretraining also improves frozen history-encoder SSv2 linear-probe Top-1 by 12.71 points over random initialization with the same pretrained codec. Pixel-space rendered-text generation and GLUE transfer provide initial evidence for language modeling through video.

1 INTRODUCTION

Efficient end-to-end training is central to streaming video modeling: future prediction must jointly shape how a model represents its past and generates new observations.

Streaming generation supports interactive simulation and long-horizon video modeling. MiniWorld trains streaming video world models from scratch, while MIRA demonstrates multiplayer interactive

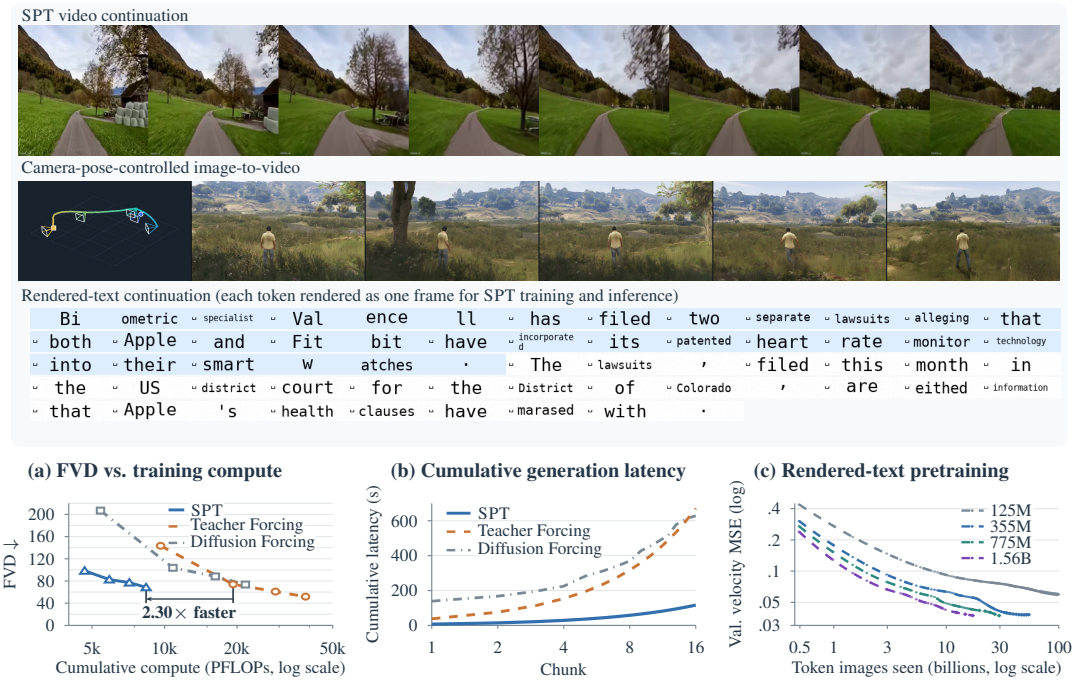


Figure 1: **Video/text continuations and measured results.** Examples: 775M SPT video, camera-pose-controlled MiniMax-H3 image-to-video, and 775M SPT rendered text (blue: prompt; gray: generated). (a) FVD versus training compute (arrow: 2.30×). (b) Latency over 16 chunks (DF: 4-chunk window). (c) Rendered-text validation velocity MSE versus processed token images.

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

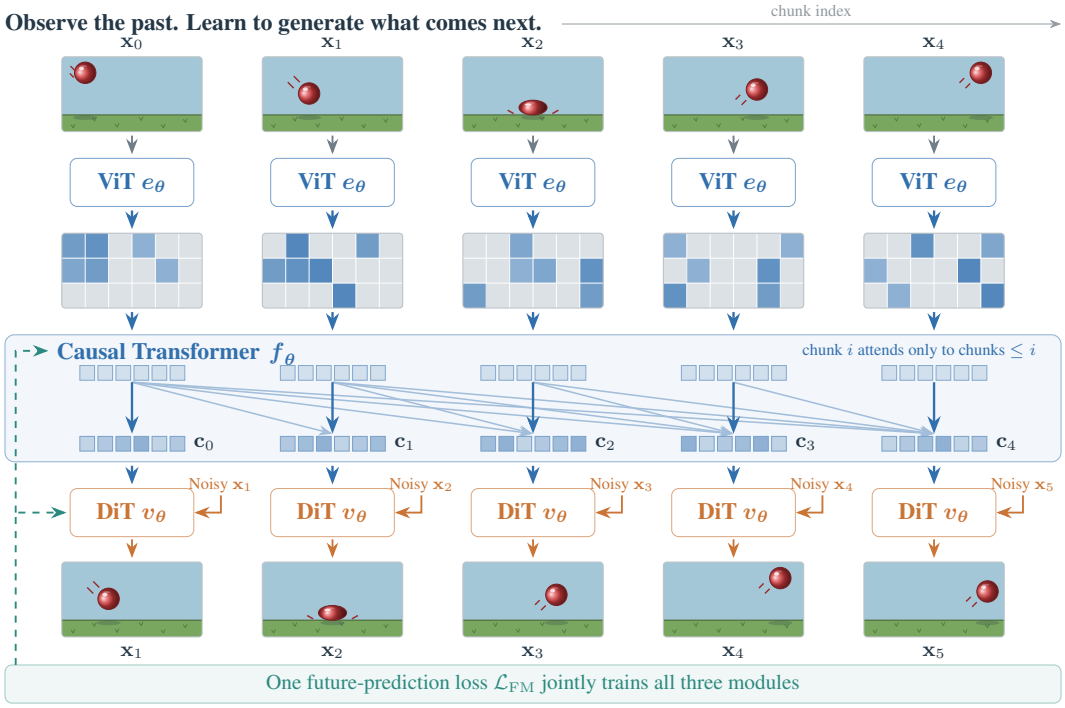


Figure 2: **Sparse past, dense future.** Each image represents a latent chunk. A shared ViT encodes each chunk; independent masks retain six of eighteen feature tokens before Causal Transformer in this illustration. Causal Transformer integrates retained features causally into $\mathbf{c}_i$. The DiT cross-attends to this state while denoising a future chunk; one flow loss trains all three modules. Token counts and offset-one targets are schematic; training samples one target from the next K chunks. Noise-time and offset conditioning are omitted.

generation (Zhao et al., 2026; Hu et al., 2026a). These systems build on causal diffusion modeling, with teacher forcing and diffusion forcing offering ways to train on temporal sequences (Chen et al., 2024). Scaling this approach requires controlling both pretraining compute and the recurring cost of generating each new chunk.

A central source of cost is the coupling of causal history integration with dense denoising in a single DiT. During training, the backbone operates on dense spatiotemporal sequences at sampled noise levels. During inference, the full denoising backbone is evaluated repeatedly for each new chunk; even with cached history keys and values, target tokens attend to the retained history at every step. Separating history encoding from denoising allows their capacity and evaluation frequency to differ (Bai et al., 2026). We investigate a complementary training principle: integrate sparse history features, but supervise a complete future chunk. This limits temporal-integration work while retaining dense generative supervision for end-to-end learning.

We introduce Sparse Predictive Transformers (SPT), a simple framework that jointly learns within-chunk features, causal history integration, and future generation. As illustrated in Fig. 2, a ViT encodes each complete visual chunk, an independently sampled subset of its features enters a Causal Transformer, and the resulting state conditions a DiT. Each training origin supervises one complete target sampled uniformly from the next K chunks. A single flow-matching loss trains all three modules end to end, in pixel or latent space, without an auxiliary reconstruction loss or teacher-model distillation.

The design supports efficient pretraining and incremental generation. Masking reduces temporal-integration and cross-attention work during pretraining. After continuation fine-tuning with unmasked histories, the Causal Transformer updates once per generated chunk, and the DiT reuses its fixed-length state across denoising steps. The encoder's cache still grows with history, but the DiT's sequence lengths do not. Fig. 1(a,b) shows the resulting quality–compute and streaming-latency benefits alongside generation examples. The same future-prediction loss also trains the history encoder (ViT + Causal Transformer) as a video representation extractor.

Experiments show that SPT uses roughly half the training compute of teacher-forcing and diffusion-forcing baselines, with lower FVD at 355M and 775M and lower streaming latency. Controlled ablations support strong but incomplete history masking and sampling targets from a short future horizon. The trained 775M model's frozen history encoder improves SSv2 linear-probe Top-1 by 12.71 percentage points over random initialization with the same pretrained codec. Finally, by rendering each text token as a frame, we explore language modeling through video generation. Rendered-text pretraining curves (Fig. 1(c)), text continuations, and GLUE transfer provide initial evidence of its feasibility.

2 Related Work

Streaming and causal video generation. MiniWorld combines a block-causal DiT with diffusion-forcing training and cached streaming inference (Zhao et al., 2026). MIRA studies multiplayer conditioning in a latent diffusion world model, using few-step distillation and a rolling context for real-time generation (Hu et al., 2026a). Diffusion Forcing, MAGI-1, and Resampling Forcing develop causal generation and rollout training (Chen et al., 2024; Sand.ai et al., 2025; Guo et al., 2025); Self Forcing addresses exposure to generated histories (Huang et al., 2025). SPT studies sparse-history pretraining and the allocation of computation between history encoding and denoising. Its teacher-forced training leaves the generated-history exposure gap open.

Predictive and masked visual representations. MAE and VideoMAE show that sparse observations can support effective visual pretraining (He et al., 2022; Tong et al., 2022). VideoMAE uses high tube-mask ratios, and VideoMAE V2 further reduces decoder work (Wang et al., 2023). DiffMAE learns masked image and video representations with diffusion reconstruction (Wei et al., 2023); masked future prediction also appears in Carreira et al. (2024); and V-JEPA 2 learns predictive features through a different target and training pipeline (Assran et al., 2025). SPT changes the information boundary: it predicts an entirely future visual chunk from a causal history, with representation learning driven by the same loss used for generation.

Masked history and autoregressive diffusion. MiMo is particularly close (Luo et al., 2026): it drops clean-history tokens using tube masks and adds current/future reconstruction to an autoregressive video diffusion objective. SPT instead uses independently sampled history sparsity and a single future flow loss, without an auxiliary reconstruction objective. We evaluate the resulting tradeoff between sparse temporal integration, continuation quality, and frozen-feature transfer.

Encoder–denoiser separation and continuous prediction. SCD jointly learns a causal context encoder and a frame-wise diffusion decoder, reusing context throughout denoising (Bai et al., 2026); MarDini likewise uses asymmetric planning and generation modules (Liu et al., 2024). MAR, NOVA, and VideoMAR combine autoregression with continuous generative prediction (Li et al., 2024; Deng et al., 2024; Yu et al., 2025), while VideoMAR's spatial target masking differs from our sparse history input. Toto studies transferable autoregressive video pretraining (Rajasegaran et al., 2025). Building on these architectural precedents, SPT investigates a single sampled-future loss with sparse temporal integration, evaluating both generation efficiency and representation quality.

Language modeling through visual inputs. PIXEL, PIXAR, PixelGPT, and MIXAR study language learning from rendered text (Rust et al., 2023; Tai et al., 2024; Chai et al., 2024; Hu et al., 2026b). Our exploratory extension renders each token as one frame and treats language modeling as visual sequence generation, evaluating token-image continuation and GLUE transfer through the predictive interface of SPT.

3 Learning from Sparse Histories

SPT consists of a history encoder combining a ViT for chunk encoding with a Causal Transformer for temporal integration, and a conditional DiT flow generator. One future-prediction loss jointly trains all three modules. We first define the prediction task, then describe this split architecture, the joint training procedure, and incremental generation.

3.1 PREDICTION TASK

Let $\mathbf{x}_i$ denote one clean visual chunk in the modeling space: either pixel values or latents from a fixed codec. SPT requires neither a particular codec nor latent compression. All time indices, history lengths, and future offsets count chunks in the chosen space. At origin i, the model forms a state $\mathbf{c}_i$ from the history $\mathbf{x}_{0:i}$ and predicts one complete chunk $\mathbf{x}_{i+k}$, with $k \in \{1, \dots, K\}$. The state excludes all observations after i, including intervening chunks. Our video experiments use latent chunks (Sec. 4.1); rendered-text experiments directly model each token image in pixel space (App. B).

Training uses finite, teacher-forced windows and requires no class, text, or action labels for video pretraining. End-to-end learning jointly trains the history encoder and conditional generator; any optional codec remains frozen. Causality is enforced between chunks. When a codec is used, causality in the original video additionally requires that encoding the observed prefix use no future frames.

3.2 ENCODING SPARSE HISTORY FEATURES

The split encoder separates within-chunk feature extraction from temporal integration (Fig. 2). Partition each chunk into S tokens with within-chunk positions $\mathbf{p}_{j,s}$. ViT $\boldsymbol{e}_{\boldsymbol{\theta}}$ encodes all tokens independently of other chunks. A retained set $\mathcal{V}_j$ is sampled independently for each chunk, leaving $M = |\mathcal{V}_j| > 0$ feature tokens for the Causal Transformer $\boldsymbol{f}_{\boldsymbol{\theta}}$:

$$\begin{aligned} \mathbf{E}_j &= \boldsymbol{e}_{\boldsymbol{\theta}}(\{(\mathbf{x}_{j,s}, \mathbf{p}_{j,s}) : s = 1, \dots, S\}), \\ \mathbf{H}_i &= \left[\boldsymbol{f}_{\boldsymbol{\theta}}\left(\{(\mathbf{E}_{j,\mathcal{V}_j}, j) : j \le i\}\right)\right]_i, \qquad \mathbf{c}_i = \boldsymbol{p}_{\boldsymbol{\theta}}(\mathbf{H}_i). \end{aligned} \tag{1}$$

Here $\mathbf{E}_j \in \mathbb{R}^{S \times d_e}$ contains the encoded chunk features, $\mathbf{E}_{j,\mathcal{V}_j}$ selects retained rows, and $[\cdot]_i$ extracts the normalized output tokens at origin i. Causal Transformer permits attention within a chunk and to all earlier chunks. Its output $\mathbf{H}_i \in \mathbb{R}^{M \times d_f}$ therefore summarizes the observed prefix at the retained positions of chunk i. The projection $\boldsymbol{p}_{\boldsymbol{\theta}}$ maps these tokens to the generator's conditioning space. Frozen-feature evaluation uses $\mathbf{H}_i$ before projection (App. F).

Because masking follows ViT, retained features may contain information from other positions in the same chunk. Masking restricts the tokens available for temporal integration while preserving full-chunk feature extraction. The generator accesses history exclusively through $\mathbf{c}_i$; no raw-history or intermediate-feature connection bypasses Causal Transformer.

The main TF/DF comparison and encoder-allocation study use this split architecture. The mask-ratio and future-horizon studies use a single causal ViT with a conditional DiT (Sec. 4.3). Rendered-text modeling instead mean-pools each chunk's ViT features into one history token and uses unmasked next-chunk prediction (App. B). All variants retain causal history conditioning and the future flow loss; the equations and algorithm describe the main sparse configuration.

3.3 SPARSE HISTORY, DENSE FUTURE PREDICTION

The conditional DiT processes all S noisy target tokens and cross-attends to the M tokens in $\mathbf{c}_i$. Target tokens supply queries; conditioning tokens supply keys and values. Self-attention connects all positions within the target chunk. Both sequence lengths are independent of total history length.

Noise time modulates the DiT blocks, while positional and future-offset embeddings identify target locations and the prediction horizon. The same state conditions every denoising step. With unmasked histories, $M = S$; the target always remains dense.

The 100% masking ablation is a separate endpoint with no observed-history conditioning during pretraining. It undergoes the same unmasked-history fine-tuning as the other mask-ratio runs before evaluation.

3.4 A SINGLE CONDITIONAL FLOW LOSS

For origin i, uniformly sample $k \in \{1, \dots, K\}$ and set $\mathbf{y} = \mathbf{x}_{i+k}$. Draw Gaussian noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and a noise fraction $t \sim p_t$ on $(0, 1)$, where p_t denotes the training noise-time distribution. We use a linear flow path (Lipman et al., 2022):

$$\mathbf{z}_t = (1 - t)\mathbf{y} + t\boldsymbol{\epsilon}, \qquad \mathbf{u}_t = \boldsymbol{\epsilon} - \mathbf{y}, \tag{2}$$

Algorithm 1 Causal pretraining from sparse visual histories

Require: Clean chunks $\mathbf{x}_{0:T-1}$ in pixel or latent space; parameters $\boldsymbol{\theta}$; horizon K; noise-time law p_t
1: Set $\mathcal{I} = \{0, \dots, T-K-1\}$; encode $\mathbf{x}_i$ with $\boldsymbol{e}_{\boldsymbol{\theta}}$ for $i \in \mathcal{I}$.
2: Sample independent retained sets $\mathcal{V}_i$ and select feature tokens before $\boldsymbol{f}_{\boldsymbol{\theta}}$.
3: Compute all $\mathbf{c}_i$ in one causal pass of $\boldsymbol{f}_{\boldsymbol{\theta}}$ and projection (1).
4: **for** each $i \in \mathcal{I}$ in parallel **do**
5: Draw one $k_i \sim \mathrm{Unif}\{1, \dots, K\}$, $t_i \sim p_t$, and $\boldsymbol{\epsilon}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.
6: Set $\mathbf{y}_i = \mathbf{x}_{i+k_i}$ and $\mathbf{z}_i = (1-t_i)\mathbf{y}_i + t_i\boldsymbol{\epsilon}_i$.
7: Predict $\widehat{\mathbf{u}}_i = \boldsymbol{v}_{\boldsymbol{\theta}}(\mathbf{z}_i, t_i, \mathbf{c}_i, k_i)$ via cross-attention to $\mathbf{c}_i$.
8: **end for**
9: Average $D^{-1}\|\widehat{\mathbf{u}}_i - (\boldsymbol{\epsilon}_i - \mathbf{y}_i)\|^2$ over origins and videos.
10: Backpropagate through the DiT, state projection, and both encoders; update $\boldsymbol{\theta}$.

so $t = 0$ is clean data and $t = 1$ is noise. The joint objective averages over the sampled future offset as well as masks and noise:

$$\mathcal{L}_{\mathrm{FM}}(\boldsymbol{\theta}) = \mathbb{E}_{\mathbf{x},\mathcal{V},i,k,t,\boldsymbol{\epsilon}}\left[\frac{1}{D}\left\|\boldsymbol{v}_{\boldsymbol{\theta}}(\mathbf{z}_t, t, \mathbf{c}_i, k) - (\boldsymbol{\epsilon} - \mathbf{x}_{i+k})\right\|_2^2\right], \tag{3}$$

where D counts scalar coordinates in the complete target chunk. The parameter vector $\boldsymbol{\theta}$ collects the distinct parameters of ViT, Causal Transformer, the state projection, and the DiT. Gradients propagate through cross-attention and $\mathbf{c}_i$ into both encoders, jointly learning the history representation and its generative readout.

The noisy target provides partial evidence about the future, and the state supplies evidence from the past. For a fixed encoder, the squared-loss optimum is $\mathbb{E}[\boldsymbol{\epsilon} - \mathbf{y} \mid \mathbf{z}_t, t, \mathbf{c}_i, k]$ (App. G). Historical features help when they resolve uncertainty remaining in the noisy target, providing a learning signal for predictive information. The conditional flow can generate different continuations from different noise draws, allowing the state to support an uncertain future.

3.5 Sampling future targets in one causal pass

For a window of T chunks and maximum offset $1 \le K < T$, we select origins with a full K-chunk future available and independently draw one offset per origin:

$$\mathcal{I} = \{0, \dots, T-K-1\}, \qquad k_i \sim \mathrm{Unif}\{1, \dots, K\}. \tag{4}$$

The DiT predicts $\mathbf{x}_{i+k_i}$ directly from $\mathbf{c}_i$ and the offset embedding. Resampling k_i exposes a history state to supervision at different temporal distances without conditioning on intervening observations or rolling out intermediate predictions.

All origins share one pass of chunk encoding, token masking, and block-causal temporal integration. The DiT processes their selected targets as separate batch elements, with no attention between targets. Each origin contributes one denoiser evaluation at one sampled noise time; training does not integrate a flow trajectory. For a fixed window length, increasing K reduces the number of supervised origins to $B = |\mathcal{I}| = T - K$, so the number of target evaluations per window also changes. Alg. 1 summarizes the computation.

Training computation. The shared causal pass reuses history features across overlapping prefixes. Masking reduces Causal Transformer's input from BS to BM tokens and each target's conditioning sequence from S to M tokens. ViT still processes all S tokens per chunk, and each DiT target remains dense. Consequently, masking saves temporal-integration and cross-attention work while retaining the costs of full-chunk encoding and target denoising. Our comparisons account for the forward and backward computation of all trainable modules (Sec. 4.2).

3.6 Reusing the state for incremental generation

Sequential continuation uses unmasked histories and $k = 1$; video models are fine-tuned with these settings before inference. Repeated prediction implements $p(\mathbf{x}_{1:T-1} \mid \mathbf{x}_0) = \prod_{i=0}^{T-2} p(\mathbf{x}_{i+1} \mid \mathbf{x}_{0:i})$.

We encode the observed prefix once and cache Causal Transformer's keys and values. Starting from $\mathbf{z}^{(0)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, Euler integration on $1 = t_0 > \cdots > t_N = 0$ gives

$$\mathbf{z}^{(n+1)} = \mathbf{z}^{(n)} - (t_n - t_{n+1}) \boldsymbol{v}_{\boldsymbol{\theta}}(\mathbf{z}^{(n)}, t_n, \mathbf{c}_i, 1). \tag{5}$$

The state $\mathbf{c}_i$ stays fixed across all N denoiser calls. Once generation is complete, we encode the new chunk and update the causal cache to obtain $\mathbf{c}_{i+1}$. Let $C_{\mathrm{v}}(H, M)$ denote the cost of chunk encoding, a cached causal update, and projection with H history chunks and M retained or pooled tokens per chunk, and let $C_{\mathrm{d}}(S, M)$ denote one DiT evaluation with S target and M conditioning tokens. The cost per generated chunk is

$$C_{\mathrm{next}} = C_{\mathrm{v}}(H, M) + N C_{\mathrm{d}}(S, M) + C_{\mathrm{codec}}. \tag{6}$$

where C_{codec} is the cost of decoding latents to pixels, or zero for pixel-space generation. Unmasked video continuation uses $M = S$; rendered-text generation uses $M = 1$ after pooling. Encoder capacity is paid for once per chunk, whereas denoiser capacity is paid for at every integration step. Both DiT sequence lengths stay fixed during rollout; the encoder's KV cache and attention cost grow with history. Streaming denotes this incremental procedure, whose latency is measured separately from training compute.

4 Experiments: Generation Quality and Training Compute

We evaluate SPT on video continuation and rendered-text modeling. For video, we compare generation quality, training compute, and streaming latency against teacher forcing and diffusion forcing, probe frozen history-encoder features on SSv2, and ablate key design choices. For rendered text, we report validation velocity MSE and autoregressive continuations, then assess downstream transfer through GLUE fine-tuning.

4.1 Experimental setup

Datasets. Our video experiments use nine subsets of SolarWM-Data (Huang et al., 2026), covering ABOT, MIND, MiraData, OmniWorld, RealCam, Sekai game and walking videos, and SpatialVid, including the available cleaned subsets. We divide source videos into non-overlapping 160-frame clips and resize frames to 256×256 without temporal resampling. The resulting corpus contains approximately 820k clips. Rendered-text pretraining uses OpenWebText (Gokaslan & Cohen, 2019) rendered as grayscale token images.

Model settings. Video models share a frozen, pretrained 3D VideoRAE codec (Xie et al., 2026), with each latent chunk representing 16 RGB frames. We use the V-JEPA 2.1-based 3D variant from the official release. We evaluate 125M, 355M, and 775M versions of the architecture described in Sec. 3. ViT encodes each chunk before 90% of its feature tokens are masked for Causal Transformer during pretraining; continuation tuning uses unmasked histories and next-chunk targets. App. A lists the model dimensions and shared training settings.

Evaluation. Our video continuation benchmark uses a fixed set of 2,048 test clips excluded from training: each model observes the first 16 frames and generates the next 128 at 256×256, using 50 Euler steps per chunk. We do not use classifier-free guidance. Metrics cover only generated frames. We report FVD (Unterthiner et al., 2019) using Kinetics-400 I3D features (Carreira & Zisserman, 2017), VGG-based LPIPS (Zhang et al., 2018), PSNR, and SSIM (Wang et al., 2004), averaging the paired framewise metrics across clips and generated frames. Cumulative rollout latency measures streaming efficiency. For rendered text, we report validation velocity MSE and development-set performance on 8 GLUE tasks to assess pretraining and language transfer.

4.2 Comparison with Teacher Forcing and Diffusion Forcing

We compare SPT with teacher forcing (TF) and diffusion forcing (DF) (Chen et al., 2024) at approximately 125M, 355M, and 775M parameters. SPT uses a split ViT/Causal Transformer encoder and a conditional DiT, sampling one target uniformly from the next $K{=}2$ latent chunks at each training origin, while TF and DF use a single causal DiT. To match total parameter counts, TF

Table 1: **Video continuation quality with lower training compute.** SPT uses roughly half the training compute of teacher forcing and diffusion forcing, with lower FVD at 355M and 775M, higher PSNR at every scale, and comparable LPIPS and SSIM. Training compute is in EFLOP (10^{18} operations).

Method	Model size	EFLOP	FVD ↓	LPIPS ↓	PSNR ↑ (dB)	SSIM ↑
Teacher forcing (DiT)	125M	7.37	110.3	0.516	14.89	0.343
	355M	19.27	74.1	0.511	14.93	0.348
	775M	36.15	53.5	0.509	14.95	0.351
Diffusion forcing (DiT)	125M	6.22	114.6	0.521	14.67	0.334
	355M	16.26	88.3	0.514	14.74	0.338
	775M	30.50	73.0	0.512	14.86	0.344
SPT	125M	3.32	118.3	0.517	14.98	0.338
	355M	8.38	68.2	0.511	15.04	0.345
	775M	17.01	51.3	0.510	14.99	0.346

and DF reallocate the parameter budget of the omitted history encoders to their DiT, increasing its depth and width. All models use the same video data and frozen codec and are evaluated under the same continuation protocol. Tab. 1 reports model size, training compute, and the four continuation-quality metrics. Training compute includes the history encoders, state interface, denoiser, backward passes, and any recomputation.

SPT uses roughly half the training compute of TF/DF, with lower FVD at 355M and 775M and higher PSNR at every scale (Tab. 1). LPIPS remains comparable, while SSIM exceeds DF and is close to TF. Fig. 1(a) shows the 355M training trajectories.

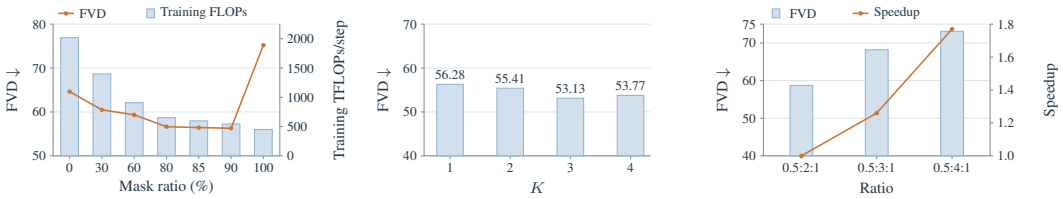
Streaming cost. Fig. 1(b) shows substantially slower growth in cumulative generation latency for SPT than for TF and DF. Our hierarchical architecture assigns causal history modeling to the Causal Transformer, which updates once per generated chunk and summarizes the observed history into a fixed-length condition. The DiT reuses this condition through cross-attention throughout denoising, keeping both its target sequence and conditioning length independent of rollout length. Growing-history computation therefore remains in the encoder rather than being repeated across all denoising steps, substantially reducing the DiT's recurring cost. This separation gives SPT a latency advantage over the single causal DiT baselines: generating 16 chunks takes 116.53 s, compared with 671.13 s for TF and 628.08 s for DF with a four-chunk window. The encoder's KV cache still grows with history.

4.3 Ablation Studies of Key Design Choices in SPT

Our training ablations examine history masking, future-target sampling, and split-ViT configurations in latent video models with approximately 355M parameters. Within each comparison, models share the pretraining data. All models are then fine-tuned under the same configuration, with 0% history masking and next-chunk-only prediction, before evaluation. We report continuation FVD (↓).

History mask ratio. We separately pretrain causal ViT–DiT models with the same architecture, varying the fraction of independently masked history tokens across 0/30/60/80/85/90/100%. In Fig. 3(a), 90% masking improves continuation FVD from 64.63 to 56.28 while reducing pretraining compute per optimizer step from 2,018.11 to 545.03 TFLOPs with the same batch size, a 73% saving. Fully masking the history further reduces per-step compute to 450.13 TFLOPs, but yields FVD 75.21 after the same unmasked-history fine-tuning. Thus, strong but incomplete masking improves both continuation quality and computational efficiency, while preserving historical evidence useful for future prediction.

Future-chunk target. At a fixed 90% history mask during pretraining, we vary the future-target horizon $K \in \{1, 2, 3, 4\}$ in the same causal ViT–DiT architecture, sampling the target uniformly from the next K latent chunks. Fig. 3(b) shows that continuation quality improves as K increases to 3 and then declines at $K{=}4$, with the lowest FVD of 53.13 at $K{=}3$. This pattern suggests that including moderately distant future targets during pretraining encourages the ViT to extract longer-



(a) History mask ratio. (b) Future-target horizon. (c) Split-ViT configuration.

Figure 3: **Continuation ablations in approximately 355M-parameter models.** (a) History-mask ratio: the line shows continuation FVD (left axis), and bars show pretraining FLOPs per optimizer step (right axis, TFLOPs). At 90% masking, FVD is lowest and per-step compute is 73% lower than without masking. (b) Future-target horizon K at 90% masking, with targets sampled uniformly from the next K latent chunks; K=3 gives the lowest FVD. (c) Split-ViT results: "Ratio" denotes ViT:Causal Transformer:DiT, bars show FVD (left axis), the line shows inference speedup (right axis), and speedups are relative to the 0.5:2:1 configuration.

term predictive information from sparse histories, providing stronger conditioning to the DiT and improving the model's overall generation quality.

Split-ViT configuration. Using the K=2 future-target setting, we replace the single causal ViT with a spatial ViT applied independently to each chunk followed by a Causal Transformer, retaining the conditional DiT. We keep the total parameter count approximately fixed while allocating a larger share to Causal Transformer, thereby reducing the size of the DiT. The history encoders process the context once per generated chunk, whereas the DiT is evaluated at each of the 50 diffusion denoising steps; reducing its size therefore lowers the repeated inference cost. We compare ViT:Causal Transformer:DiT ratios of 0.5:2:1, 0.5:3:1, and 0.5:4:1, each pretrained with 90% history masking and the K=2 target distribution. Fig. 3(c) shows inference speedups of 1.00×, 1.26×, and 1.77×, respectively, with corresponding FVDs of 58.71, 68.18, and 73.09.

History order at inference. For the 775M model, we provide eight ground-truth latent chunks and generate only the ninth. Randomly shuffling the history raises mean latent-space MSE from 0.4055 to 1.2996 (approximately 3.2×). This sensitivity to temporal order shows that generation uses information encoded from the history; App. E provides the protocol and average results.

4.4 Frozen history-encoder action recognition

We linearly probe the frozen history encoder (ViT + Causal Transformer) from the 775M SPT model on Something-Something V2 (Goyal et al., 2017), using pooled Causal Transformer outputs before conditioning projection. With the same frozen VideoRAE and probe protocol, training raises validation Top-1 from 17.05% for a randomly initialized history encoder to 29.76% (+12.71 percentage points), and Top-5 from 37.92% to 57.01%. These gains show that future prediction learns action-relevant history representations; App. F provides the protocol and full results.

4.5 Rendered-text generation

We tokenize OpenWebText (Gokaslan & Cohen, 2019) with GPT-2 byte-level BPE (Radford et al., 2019) via `tiktoken` (50,257 IDs), appending an end-of-text token per document. A fixed table renders each ID as a 32 × 128 antialiased, 8-bit grayscale image, black on white. For rendered text, SPT pools the ViT features of each image's 8 patches into 1 vector for the Causal Transformer; the DiT predicts the next complete token image directly in pixel space from unmasked histories, without a latent codec. Validation velocity MSE decreases as more token images are processed across the 4 model sizes (Fig. 1(c)).

For 8 GLUE development tasks (Wang et al., 2019), we fine-tune the visual encoder and a task head on rendered inputs with settings adapted from PixelGPT (Chai et al., 2024) (App. B). Tab. 2 compares text-token BERT (Devlin et al., 2019) and GPT-2 (Radford et al., 2019) with pixel-input PIXEL (Rust et al., 2023), PIXAR (Tai et al., 2024), PixelGPT, MIXAR (Hu et al., 2026b), and SPT.

BERT, GPT-2, PIXEL, and PixelGPT scores follow Chai et al. (2024); PIXAR and MIXAR scores follow their original studies.

Table 2: **GLUE development-set performance by input modality.**

Model	MNLI-m/mm	QQP	QNLI	SST-2	CoLA	STS-B	MRPC	RTE
Text-token Transformers								
BERT (110M)	84.0/84.2	87.6	91.0	92.6	60.3	88.8	90.2	69.5
GPT-2 (126M)	81.0	89.4	87.7	92.5	77.0	74.9	71.5	52.0
Pixel-input Transformers								
PIXEL (86M)	78.1/78.9	84.5	87.8	89.6	38.4	81.1	88.2	60.5
PIXAR (85M)	79.7/80.1	86.3	85.7	89.3	37.0	82.4	82.8	57.7
PixelGPT (317M)	79.0/78.2	86.0	85.6	90.1	35.3	80.3	84.6	63.9
MIXAR (477M)	79.9/80.5	86.7	86.7	89.8	30.6	84.3	84.0	66.8
SPT (125M)	79.1/80.6	85.9	87.2	89.7	37.4	84.0	87.7	63.5

The 125M model's GLUE scores are broadly within the range of prior pixel-input models across tasks (Tab. 2). Fig. 4 shows autoregressive token-image continuations from 4 model sizes, each conditioned on a distinct held-out prefix of 32 token images.

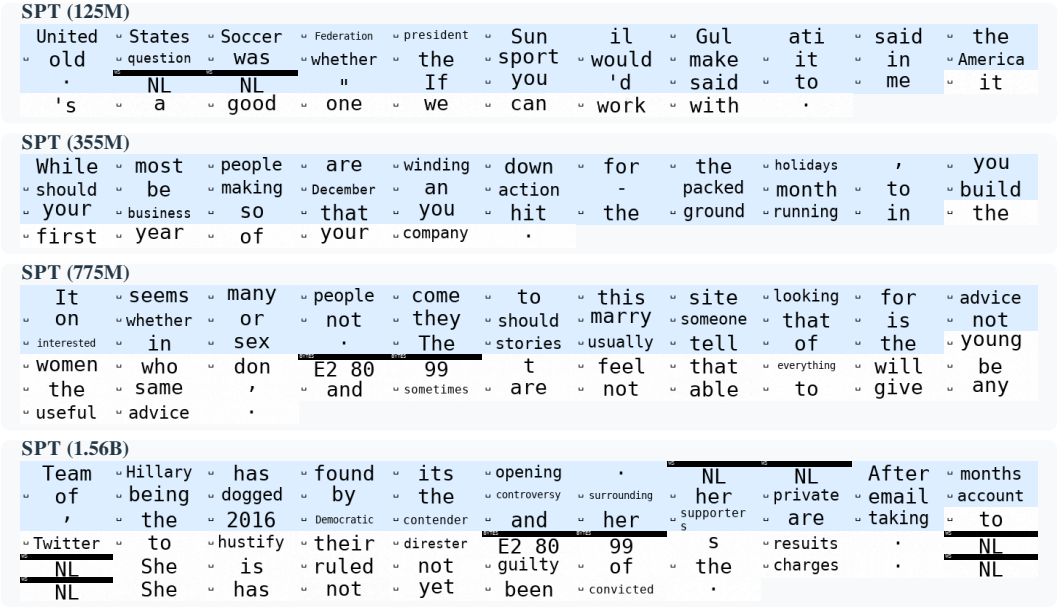


Figure 4: **Rendered-text continuations.** From top to bottom: 125M, 355M, 775M, and 1.56B SPT models. Blue token images show the 32-token prompt; grayscale images show generated tokens. Each model uses a different held-out prompt.

5 Discussion and Conclusion

SPT jointly pretrains streaming generation and history representations through one future flow loss in pixel or latent space. Sparse history integration reduces training computation, while state reuse lowers recurring denoising costs. Experiments show favorable video quality–compute tradeoffs, improved frozen history-encoder features, and initial evidence for language modeling through rendered video. Future work includes longer-rollout evaluation and isolating the contributions of sparsity and codec pretraining to representation transfer (App. H).

Reproducibility statement

The appendices detail video and rendered-text training, GLUE fine-tuning, SSv2 probing, the MiniMax-H3 camera-control demo, the history-order inference ablation, and causality and gradient flow; Sec. 4 reports the quantitative evaluations.

AI use statement

Generative AI tools assisted with manuscript editing, related-work literature review, selected theoretical derivations and numerical checks. The authors manually reviewed the AI-assisted content and take responsibility for all claims, proofs, code, and results in this work.

References

Mido Assran, Adrien Bardes, David Fan, et al. V-JEPA 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025. URL https://arxiv.org/abs/2506.09985.

Xingjian Bai, Guande He, Zhengqi Li, Eli Shechtman, Xun Huang, and Zongze Wu. Causality in video diffusers is separable from denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026. URL https://arxiv.org/abs/2602.10095.

Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the Kinetics dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6299–6308, 2017. URL https://openaccess.thecvf.com/content_cvpr_2017/html/Carreira_Quo_Vadis_Action_CVPR_2017_paper.html.

João Carreira, Michael King, Viorica Patraucean, Dilara Gokay, Catalin Ionescu, Yi Yang, Daniel Zoran, Joseph Heyward, Carl Doersch, Yusuf Aytar, Dima Damen, and Andrew Zisserman. Learning from one continuous video stream. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 28751–28761, 2024. URL https://arxiv.org/abs/2312.00598.

Yekun Chai, Qingyi Liu, Jingwu Xiao, Shuohuan Wang, Yu Sun, and Hua Wu. Autoregressive pre-training on pixels and texts. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 3106–3125. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.emnlp-main.182. URL https://aclanthology.org/2024.emnlp-main.182/.

Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-0759. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/2aee1c4159e48407d68fe16ae8e6e49e-Abstract-Conference.html.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4171–4186. Association for Computational Linguistics, 2019. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423/.

Aaron Gokaslan and Vanya Cohen. OpenWebText corpus. https://skylion007.github.io/OpenWebTextCorpus/, 2019.

Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, Florian Hoppe, Christian Thurau, Ingo Bax, and Roland Memisevic. The "something something" video database for learning and evaluating visual common sense. In

Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 5842–5850, 2017. URL https://openaccess.thecvf.com/content_iccv_2017/html/Goyal_The_Something_Something_ICCV_2017_paper.html.

Yuwei Guo, Ceyuan Yang, Hao He, Yang Zhao, Meng Wei, Zhenheng Yang, Weilin Huang, and Dahua Lin. End-to-end training for autoregressive video diffusion via self-resampling. *arXiv preprint arXiv:2512.15702*, 2025. URL https://arxiv.org/abs/2512.15702.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16000–16009, 2022.

Anthony Hu, Václav Volhejn, Adrien Ramanana Rahary, et al. Multiplayer interactive world models with representation autoencoders. *arXiv preprint arXiv:2607.05352*, 2026a. URL https://arxiv.org/abs/2607.05352.

Chen Hu, Yintao Tai, Antonio Vergari, Frank Keller, and Alessandro Suglia. MIXAR: Scaling autoregressive pixel-based language models to multiple languages and scripts. *arXiv preprint arXiv:2604.11575v1*, 2026b. URL https://arxiv.org/abs/2604.11575v1.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Junchao Huang, Guian Fang, Shengju Qian, Xianghao Kong, Zhuoran Zhao, Wei Huang, Yihua Du, Zixin Zhang, Justin Cui, Yuchao Gu, Yukang Chen, Xinting Hu, Tianyu He, Shaoshuai Shi, Zhuotao Tian, Xin Wang, Mike Zheng Shou, and Li Jiang. SolarWM: Open data and scalable training for long-horizon video world models, 2026. URL https://arxiv.org/abs/2609.02886.

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. *Advances in Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2506.08009.

Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive image generation without vector quantization. *Advances in Neural Information Processing Systems*, 37: 56424–56445, 2024.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Haozhe Liu, Shikun Liu, Zijian Zhou, Mengmeng Xu, Yanping Xie, Xiao Han, Juan C. Pérez, Ding Liu, Kumara Kahatapitiya, Menglin Jia, Jui-Chieh Wu, Sen He, Tao Xiang, Jürgen Schmidhuber, and Juan-Manuel Pérez-Rúa. MarDini: Masked autoregressive diffusion for video generation at scale. *arXiv preprint arXiv:2410.20280*, 2024. URL https://arxiv.org/abs/2410.20280.

Wenyang Luo, Haina Qin, Bing Li, Jiwen Lu, Xin Tao, Pengfei Wan, and Kun Gai. Improving autoregressive video modeling with history understanding. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/9d30c2def27b5c6a5fb21a9aa5c16f8f-Abstract-Conference.html.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners, 2019. URL https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

Jathushan Rajasegaran, Ilija Radosavovic, Rahul Ravishankar, Yossi Gandelsman, Christoph Feichtenhofer, and Jitendra Malik. An empirical study of autoregressive pre-training from videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 19108–19118, 2025. URL https://arxiv.org/abs/2501.05453.

Phillip Rust, Jonas F. Lotz, Emanuele Bugliarello, Elizabeth Salesky, Miryam de Lhoneux, and Desmond Elliott. Language modelling with pixels. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2207.06991.

Sand.ai, Hansi Teng, Hongyu Jia, Lei Sun, Lingzhi Li, et al. MAGI-1: Autoregressive video generation at scale. *arXiv preprint arXiv:2505.13211*, 2025. URL https://arxiv.org/abs/2505.13211.

Yintao Tai, Xiyang Liao, Alessandro Suglia, and Antonio Vergari. PIXAR: Auto-regressive language modeling in pixel space. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 14673–14695. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-acl.874. URL https://aclanthology.org/2024.findings-acl.874/.

Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. VideoMAE: Masked autoencoders are data-efficient learners for self-supervised video pre-training. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://arxiv.org/abs/2203.12602.

Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717v2*, 2019. URL https://arxiv.org/abs/1812.01717v2.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rJ4km2R5t7.

Limin Wang, Bingkun Huang, Zhiyu Zhao, Zhan Tong, Yinan He, Yi Wang, Yali Wang, and Yu Qiao. VideoMAE V2: Scaling video masked autoencoders with dual masking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14549–14560, 2023. URL https://arxiv.org/abs/2303.16727.

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612, 2004. doi: 10.1109/TIP.2003.819861. URL https://ieeexplore.ieee.org/document/1284395.

Chen Wei, Karttikeya Mangalam, Po-Yao Huang, Yanghao Li, Haoqi Fan, Hu Xu, Huiyu Wang, Cihang Xie, Alan Yuille, and Christoph Feichtenhofer. Diffusion models as masked autoencoders. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 16284–16294, 2023. URL https://arxiv.org/abs/2304.03283.

Zhihao Xie, Junfeng Wu, Xinting Hu, Junchao Huang, and Li Jiang. VideoRAE: Taming video foundation models for generative modeling via representation autoencoders, 2026. URL https://arxiv.org/abs/2607.14088.

Hu Yu, Biao Gong, Hangjie Yuan, DanDan Zheng, Weilong Chai, Jingdong Chen, Kecheng Zheng, and Feng Zhao. VideoMAR: Autoregressive video generation with continuous tokens. *arXiv preprint arXiv:2506.14168*, 2025. URL https://arxiv.org/abs/2506.14168.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 586–595, 2018. URL https://openaccess.thecvf.com/content_cvpr_2018/html/Zhang_The_Unreasonable_Effectiveness_CVPR_2018_paper.html.

Yian Zhao, Ruochong Zheng, Hongcan Guo, Yu Yan, Jian Zhang, and Jie Chen. MiniWorld: Democratizing the training of video world models from scratch. *arXiv preprint arXiv:2608.01127*, 2026. URL https://arxiv.org/abs/2608.01127.

A Main Video Training and Model Configurations

Tab. 3 summarizes the shared optimization settings for SPT and the teacher-forcing and diffusion-forcing baselines in the main video experiments. All models use the same frozen VideoRAE, training split, and cached video latents described in Sec. 4.1. Tab. 4 lists their architecture dimensions at each model size.

Table 3: **Main video training settings.** Values are shared by SPT, teacher forcing, and diffusion forcing.

Setting	Value
Optimizer	Muon with auxiliary Adam
Base learning rate	10^{-4}
Learning-rate schedule	Linear warmup, then constant
Warmup	100 optimizer steps
Muon momentum	0.95
Muon orthogonalization	5 Newton–Schulz iterations
Auxiliary Adam (β_1, β_2)	$(0.9, 0.95)$
Auxiliary Adam ϵ	10^{-8}
Weight decay	0
Global batch size	64 video clips
Gradient clipping	Global gradient norm of 1.0
Training precision	BF16 autocast; FP32 gradient norms
EMA decay	0.99

Table 4: **Model architectures for the main video comparison.** Teacher forcing and diffusion forcing share the same DiT architecture at each size. ViT and Causal Transformer share a hidden dimension; all attention heads have dimension 64. Model sizes refer to the complete trainable model and exclude the frozen VideoRAE.

Method	Size	ViT layers	Causal Transformer layers	Encoder hidden dimension	DiT layers	DiT hidden dimension
SPT	125M	3	18	576	7	448
	355M	3	18	1024	7	768
	775M	3	18	1536	7	1152
TF / DF	125M	—	—	—	14	768
	355M	—	—	—	22	1152
	775M	—	—	—	24	1536

B Rendered-Text Training and Evaluation

Corpus and rendering. Rendered-text pretraining uses OpenWebText (Gokaslan & Cohen, 2019). We partition documents into disjoint training, validation, and test sets using hashes of their content. We encode the text with GPT-2 byte-level BPE (Radford et al., 2019); the training split contains 7,933,495 documents and 8.949×10^9 token IDs. We append an end-of-text token to each document and sample windows from the packed stream; windows may cross document boundaries. A fixed rendering table maps each of the 50,257 IDs to a 32×128 antialiased, 8-bit grayscale token image with black text on white, divided into 8 16×32 patches. Sampled windows may overlap, so processed-token counts include repeated text.

Architecture and pretraining. Training and generation operate directly in pixel space, without a VAE or other latent codec. For each history token image, a shared ViT encodes its 8 patches, which are mean-pooled into 1 vector for the Causal Transformer. A conditional DiT predicts the 8 patches of the next token image. The noise-time law p_t is an equal mixture of $\mathcal{U}(0, 1)$ and $\mathcal{U}(0.8, 1)$, and the velocity loss has no noise-time weighting. We report velocity MSE on validation data. During rollout,

generated token images are fed back directly as history. Tab. 5 specifies the 4 architectures shown in Fig. 4.

Table 5: **Rendered-text model architectures.** ViT applies full attention within each token image, and Causal Transformer applies causal attention to the pooled token-image vectors. All attention heads have dimension 64.

Size	ViT layers	Causal Transformer layers	ViT width	DiT layers	DiT width
125M	4	10	704	6	576
355M	5	13	1,024	8	896
775M	7	20	1,280	12	1,024
1.56B	9	25	1,600	15	1,344

Table 6: **Rendered-text pretraining settings.**

Setting	Value
Context length (token images)	1,024
Global batch size	480
Per-GPU microbatch $\times$ accumulation	125M: 4×15; 355M: 2×30; 775M/1.56B: 1×60
Token images per update	491,520
Optimizer	AdamW, $(\beta_1, \beta_2) = (0.9, 0.95)$
Learning rate	3×10^{-4}; 200-step linear warmup, then constant
Weight decay; gradient-norm clipping	0.1; 1.0
Training precision	BF16 autocast; FP32 parameters; activation checkpointing
History masking; condition dropout	0; 0
EMA decay per update	0.99

Dependence on history. To assess how the 125M model uses its history, we measure validation velocity MSE for the next token image at fixed noise time $t = 0.95$. We compare the original prefix with the same token images randomly reordered. The increase under shuffling (Table 7) indicates that the Causal Transformer models the ordered history on which the DiT relies for next-token prediction.

Table 7: **Effect of history shuffling on prediction of the next token image.** Validation velocity MSE for the 125M model at $t = 0.95$; lower is better.

History	Velocity MSE ↓
Original	0.1199
Shuffled	0.2130

GLUE evaluation. GLUE evaluation (Wang et al., 2019) uses the standard training and development splits and the same fixed rendering table as pretraining. We separate sentence pairs with the rendered end-of-text image and limit inputs to 768 token images. For classification, we add each token image's pooled ViT vector to its normalized Causal Transformer output and apply a linear task head at the last non-padding position. We fine-tune the image patch embedding, ViT, Causal Transformer, Causal Transformer output normalization, and task head; the DiT is unused for classification. Each task is initialized independently from the same 125M EMA checkpoint with seed 42. The task-specific settings in Tab. 8 are adapted from PixelGPT (Chai et al., 2024). AdamW uses $(\beta_1, \beta_2) = (0.9, 0.999)$, zero weight decay, and gradient-norm clipping at 1.0. We report results from the best checkpoint on each task's development set.

Table 8: **GLUE fine-tuning settings for the 125M SPT checkpoint.** Batch sizes are global; steps are configured maxima.

Task	Batch	Learning rate	Steps	Warmup	Schedule	Head dropout
MNLI	64	5×10^{-5}	15,000	100	Linear	0
QQP	512	1×10^{-4}	1,500	100	Cosine	0.1
QNLI	64	5×10^{-5}	8,000	100	Linear	0
SST-2	64	5×10^{-5}	8,000	100	Cosine	0.1
CoLA	512	5×10^{-6}	2,000	10	Cosine	0.1
STS-B	512	3×10^{-5}	2,000	10	Cosine	0.1
MRPC	32	5×10^{-5}	2,000	10	Linear	0
RTE	32	3×10^{-5}	2,000	10	Linear	0

We report matched and mismatched accuracy for MNLI; F1 for QQP and MRPC; Matthews correlation for CoLA; Spearman correlation for STS-B; and accuracy for the remaining tasks.

C Camera-Pose-Controlled Image-to-Video with MiniMax-H3

Setup. Starting from the SolarWM H3 Stage-0.5 EMA checkpoint, we post-train an SPT variant of the 33B-parameter MiniMax-H3 video model on SolarWM-Data (Huang et al., 2026). We copy H3's initial 8 transformer layers to initialize a history ViT with causal self-attention. Its outputs condition the H3 generator through cross-attention, while the generator's DiT retains bidirectional self-attention. We train rank-384 low-rank adaptation (LoRA) modules (Hu et al., 2022) in the generator for 3,000 steps. For camera-pose-controlled image-to-video generation, we condition on a specified camera-pose trajectory and 1 clean latent frame; the frame condition is dropped with probability 0.5 during training. Optimization uses a global batch size of 4, AdamW with a peak learning rate of 3×10^{-5}, and EMA decay 0.999. At inference, we generate videos at 768×1344 resolution and 24 frames per second using 50 sampling steps.



Figure 5: **Camera-pose-controlled MiniMax-H3 SPT image-to-video samples.** The top 2 rows show training examples; the bottom 2 show held-out test examples. The left column shows the camera-pose trajectory (blue to yellow); the remaining columns show the conditioning frame followed by generated frames.

D Qualitative Video Continuations from the Pretrained SPT Model

Setup. We show qualitative video continuations from the pretrained 775M SPT model used in the main video experiments, following the video-continuation protocol in Sec. 4.1. Figure 6 presents six qualitative examples.

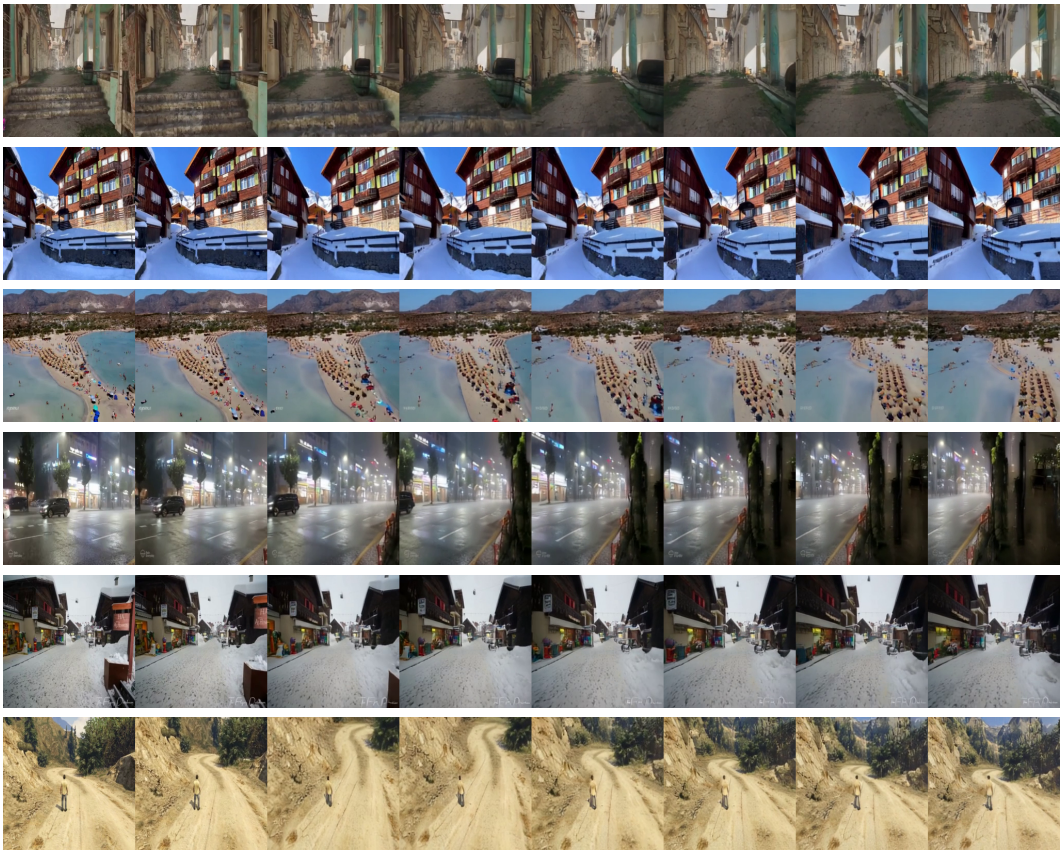


Figure 6: **Qualitative video continuations from the pretrained 775M SPT model.** Each row shows one example, with eight frames arranged from left to right in temporal order.

E History-Order Ablation at Inference

Protocol. We use the 775M SPT video model to test whether next-chunk generation depends on the temporal order of its history. For each example, we provide eight ground-truth latent chunks in either their original chronological order or a randomly shuffled order. Both conditions contain the same eight chunks. The model generates only one next latent chunk, which is compared with the ground-truth ninth chunk using latent-space mean squared error (MSE). We report the mean error across examples.

Table 9: **History-order sensitivity of the 775M SPT model at inference.** Mean latent-space MSE for predicting the ninth chunk from eight ground-truth history chunks.

History order	Mean latent MSE ↓
Chronological	0.4055
Randomly shuffled	1.2996

Results. Shuffling the history raises mean MSE from 0.4055 to 1.2996, approximately $3.2\times$ the chronological baseline (Tab. 9). The observed history chunks are unchanged, but their temporal arrangement is disrupted. This degradation shows that next-chunk prediction uses temporal information carried by the history encoder, supporting its role in providing history-dependent conditioning to the generator.

F Frozen History-Encoder Representation Evaluation

Task and comparison. We evaluate whether the history encoder trained through future prediction learns features useful for action recognition on Something-Something V2 (SSv2) (Goyal et al., 2017), using its 174 classes, 168,913 official training videos, and 24,777 official validation videos. We extract the frozen history encoder (ViT + Causal Transformer) from the 775M SPT model and compare it with a randomly initialized encoder of identical architecture (seed 42). Both groups use the same frozen, pretrained VideoRAE, latent normalization, feature extraction, pooling, and classifier-selection protocol. The DiT is not used. This comparison measures the contribution of jointly training both encoder modules with the pretrained codec held fixed.

Feature extraction. We resample each complete video to 16 FPS without changing playback speed, resize frames to 256×256, and encode consecutive, non-overlapping 16-frame chunks with VideoRAE. A final incomplete chunk repeats its last frame for padding. The encoder has three ViT blocks and eighteen Causal Transformer blocks, with width 1,536 and approximately 605M parameters; 775M denotes the complete generative model. With history masking disabled, we process all chunks using the native causal attention and read the final normalized Causal Transformer tokens before conditioning projection. Spatial and temporal mean pooling yields one 1,536-dimensional feature per video, weighting each latent temporal plane by its proportion of valid RGB frames. These weights adjust pooling; padded tokens still participate in encoder attention. Codec normalization uses the original video-training statistics. Feature extraction uses BF16, and pooled features are stored in FP32. VideoRAE and both encoder modules remain frozen throughout evaluation.

Linear-probe protocol. The only trainable component is an affine classifier from 1,536 features to 174 classes. A fixed, class-stratified split of the official training set provides 152,023 fitting videos and 16,890 development videos. Feature-wise mean and standard deviation are estimated only from the fitting subset during selection, with a standard-deviation floor of 10^{-6}. We search learning rates $\{0.01, 0.1, 1.0\}$ using SGD with momentum 0.9, no weight decay, batch size 1,024, and a 50-epoch cosine schedule. Every five epochs, development Top-1 selects the learning rate and epoch, breaking ties by Top-5, earlier epoch, then smaller learning rate. Each group searches the same grid independently. We then refit the classifier and normalization on the full official training set with classifier seeds 0, 1, and 2, retaining the selected prefix of the original schedule. The official validation set is used only for final reporting.

Table 10: **Frozen history-encoder linear probing on SSv2.** The encoder comprises ViT and Causal Transformer. Official validation accuracy (%) is reported as mean $\pm$ population standard deviation across three classifier seeds. The pretrained VideoRAE is identical and frozen in both groups.

Frozen history encoder	Top-1 ↑	Top-5 ↑
Random initialization	17.05 ± 0.08	37.92 ± 0.07
SPT trained	29.76 ± 0.03	57.01 ± 0.10

Results. Training improves Top-1 by 12.71 percentage points and Top-5 by 19.09 percentage points under the same frozen-codec protocol (Tab. 10). Thus, future-prediction training produces action-relevant information that a linear classifier can read from the history encoder, beyond that available through the same randomly initialized architecture. All 193,690 videos passed feature coverage and validity checks, and independent CPU replay reproduced the saved classifiers' validation prediction counts. The reported variation covers classifier randomness with fixed encoder weights; it does not measure variation across independent pretraining runs or isolate the contributions of the individual encoder modules or history masking.

G Causality, Gradient Flow, and Representation Limits

We analyze the split architecture in Sec. 3. All temporal indices refer to chunks in the chosen pixel or latent space. The results distinguish architectural dependencies and gradient paths from properties that require empirical evaluation.

Encoder causality. For fixed parameters and retained sets, ViT $\boldsymbol{e_\theta}$ encodes each chunk independently. In Causal Transformer $\boldsymbol{f_\theta}$, queries at chunk i attend only to keys and values from chunks $j \le i$. Tokenwise normalization, feed-forward maps, residual additions, and state projection preserve this dependency. Induction over layers therefore establishes that $\mathbf{c}_i$ depends only on $\mathbf{x}_{0:i}$ and the corresponding retained sets, excluding both the future target and any intervening chunks. The DiT reads $\mathbf{c}_i$ through cross-attention without feeding target features back into the history encoder. This is causality between chunks: retained features may incorporate all positions within their own chunk because masking follows ViT. If a codec is used, causality in the original video additionally requires that it introduce no dependence on frames beyond the observed prefix.

End-to-end supervision. Let $\mathbf{y} = \mathbf{x}_{i+k}$, $\widehat{\mathbf{u}} = \boldsymbol{v_\theta}(\mathbf{z}_t, t, \mathbf{c}_i, k)$, and $\mathbf{u}_t = \boldsymbol{\epsilon} - \mathbf{y}$. Whether $\mathbf{y}$ contains pixels or fixed-codec latents, $\mathbf{z}_t$ and $\mathbf{u}_t$ are independent of the trainable parameters $\boldsymbol{\theta}$. Applying the chain rule to (3), under the usual conditions for interchanging differentiation and expectation, gives

$$\nabla_{\boldsymbol{\theta}} \mathcal{L}_{\mathrm{FM}} = \mathbb{E}\left[\left(\left.\frac{\partial \widehat{\mathbf{u}}}{\partial \boldsymbol{\theta}}\right|_{\mathbf{c}_i} + \frac{\partial \widehat{\mathbf{u}}}{\partial \mathbf{c}_i}\frac{\partial \mathbf{c}_i}{\partial \boldsymbol{\theta}}\right)^{\top} \frac{2}{D}(\widehat{\mathbf{u}} - \mathbf{u}_t)\right]. \tag{7}$$

The first term differentiates the denoiser while holding $\mathbf{c}_i$ fixed. The second propagates the same future-prediction loss through cross-attention, state projection, Causal Transformer, and ViT. Thus, one objective provides a gradient path to all three modules without an auxiliary encoder loss. This path does not guarantee nonzero gradients or useful representations. The decomposition applies to history-conditioned training; the 100% masking pretraining endpoint has no observed-history conditioning.

Parallel and cached evaluation. With fixed parameters, identical inputs, positions, and retained sets, and stochastic layers disabled, parallel causal encoding and incremental cached encoding produce the same states in exact arithmetic. Each update processes all retained tokens of the new chunk together, preserving attention within that chunk, and reuses the preceding chunks' keys and values. Earlier states remain unchanged because they cannot attend to later chunks. Finite-precision implementations can differ numerically, so this equivalence is an architectural property rather than a reported numerical validation. It does not justify reusing caches after parameter updates; training uses separate teacher-forced windows.

Conditioning length and streaming memory. The DiT receives only the M tokens in $\mathbf{c}_i$, but Causal Transformer retains historical keys and values. For L causal layers, H history chunks, and hidden width d_f, the KV cache stores $O(LHMd_f)$ elements; each update still attends to past tokens. ViT has no cross-chunk cache. Thus, fixed-length DiT conditioning does not imply constant memory or constant total cost as the history grows. During pretraining, masking reduces temporal-integration and cross-attention work while retaining full-chunk ViT encoding and dense target denoising. Unmasked streaming uses $M = S$, so its efficiency depends on reusing the history state across denoising steps and is evaluated separately from sparse pretraining.

Predictive information and representation limits. For a fixed history encoder and state projection, set $A = (\mathbf{z}_t, t, \mathbf{c}_i, k)$ and $\mathbf{U} = \boldsymbol{\epsilon} - \mathbf{y}$. Among all square-integrable predictors of A, the squared-loss minimizer is $\boldsymbol{v}^*(A) = \mathbb{E}[\mathbf{U} \mid A]$, up to almost-sure equality. For finite second moments,

$$\mathbb{E}\|\mathbf{U} - \boldsymbol{v}(A)\|^2 = \mathbb{E}\|\mathbf{U} - \mathbb{E}[\mathbf{U} \mid A]\|^2 + \mathbb{E}\|\mathbb{E}[\mathbf{U} \mid A] - \boldsymbol{v}(A)\|^2. \tag{8}$$

The cross term vanishes because $\mathbb{E}[\mathbf{U} - \mathbb{E}[\mathbf{U} \mid A] \mid A] = 0$. The first term is the residual prediction error given A; the second measures deviation from the conditional mean. Compared with conditioning only on $(\mathbf{z}_t, t, k)$, access to $\mathbf{c}_i$ cannot increase the minimum population squared error and reduces it when the state changes the conditional mean on a set of positive probability. This provides a precise sense in which history can help future prediction. A finite-capacity DiT need not attain this optimum, and the objective alone does not guarantee an interpretable, minimal, or downstream-sufficient representation. Generation quality and representation transfer therefore remain empirical questions.

H LIMITATIONS

Sparse histories and streaming generation. History masking can remove information needed for future prediction. The favorable masking ratios in our video experiments need not transfer to other data or tasks. Sparsity reduces temporal-integration and cross-attention work, but retains dense chunk encoding and target denoising. During streaming generation, the history encoder's KV cache and attention cost still grow with the observed prefix. Training on finite, teacher-forced windows also leaves a mismatch with generated histories at inference; the reported continuation results do not establish reliability over arbitrarily long rollouts.

Representation learning. The SSv2 probe evaluates the complete frozen history encoder (ViT + Causal Transformer) against random initialization with the same pretrained codec. It does not isolate the contributions of the two encoder modules, history masking, or codec pretraining. The reported variation covers classifier seeds, not independent pretraining runs. The history-order ablation shows sensitivity to temporal order, but does not establish which semantic or physical properties the model represents. Broader transfer evaluations and controlled pretraining studies remain necessary.

Modeling space and experimental coverage. SPT can operate in pixel or latent space without requiring a particular codec. However, our natural-video experiments use one pretrained VideoRAE configuration, while pixel-space experiments use rendered text. The present results therefore do not establish the same quality–compute tradeoff across alternative codecs or pixel-space natural-video models.

Rendered-text modeling. The language experiments establish feasibility through validation velocity MSE, qualitative continuations, and GLUE fine-tuning. They do not establish competitiveness with text-token language models in likelihood, generation quality, or computational efficiency. The fixed token-image rendering and pooling scheme is one design choice; robustness to other renderings and longer generated sequences remains untested.