CanvasMAR: Improving Masked Autoregressive Video Prediction with Canvas

Zian Li [1,2] **Muhan Zhang**[1,3]*

[1]Institute for Artificial Intelligence, Peking University
[2]School of Intelligence Science and Technology, Peking University
[3]State Key Laboratory of General Artificial Intelligence

zian@stu.pku.edu.cn muhan@pku.edu.cn

Abstract

Masked autoregressive models (MAR) have emerged as a powerful paradigm for image and video generation, combining the flexibility of masked modeling with the expressiveness of continuous tokenizers. However, when sampling individual frames, video MAR models often produce highly distorted outputs due to the lack of a structured global prior, especially when using only a few sampling steps. To address this, we propose **CanvasMAR**, a novel autoregressive video prediction model that predicts high-fidelity frames with few sampling steps by introducing canvas—one-step prediction of the future-frame expectation that serves as a non-uniform mask during masked generation. The canvas supplies global structure early in sampling, enabling faster and more coherent frame synthesis. To further stabilize autoregressive sampling, we propose an easy-to-hard curriculum via a motion-aware sampling order that synthesizes stationary regions before attending to highly dynamic ones. We also integrate compositional classifier-free guidance that jointly strengthens the canvas and temporal conditioning to improve generation fidelity. Experiments on the BAIR, UCF-101, and Kinetics-600 benchmarks demonstrate that CanvasMAR produces higher-quality videos with fewer autoregressive steps. On the challenging Kinetics-600 dataset, CanvasMAR achieves remarkable performance among autoregressive models and rivals advanced diffusion models.

1 Introduction

Masked generative models have achieved remarkable success in both image generation [5, 31, 32] and video generation [67, 74, 7]. By representing images as tokens and generating them set by set in a random order,[2] these models achieve higher fidelity and scalability than approaches that rely on predefined orders [13]. Among these approaches, Masked Autoregressive Models (MAR) [32, 13][3] pioneer the generation of continuous image tokens with a diffusion head, in contrast to prior works that use discrete tokens [5, 67]. This eliminates the quantization errors inherent in discrete-token masked generative models and avoids the challenges of training high-quality VQ-VAEs [52, 12, 68, 18], demonstrating strong potential both theoretically and empirically.

*Corresponding author.

[2]For models such as MaskGiT [5], the generation order is not strictly random because it also relies on token probabilities interpreted as confidence scores. However, we adopt a generalized notion of "random" to contrast with strictly predefined orders like raster scan.

[3]We adopt the generalized notion of "autoregressive" introduced in [32], which treats masked generation as autoregression along the spatial axis in random order.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

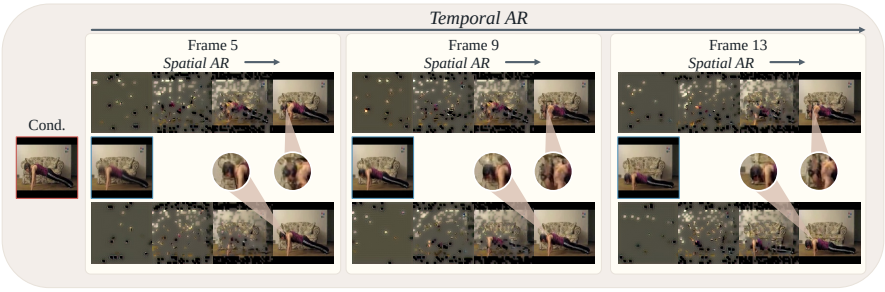


Figure 1: Video generation with Masked Autoregressive Models using only 8 autoregressive steps per 32×32 latent frame. Frames with red borders (left) denote the conditioning inputs. Without a canvas (top row), the model quickly loses global coherence for the body. Our approach (second and bottom rows) mitigates this by first predicting a blurred canvas (shown in the second row with blue borders) that captures coarse motion and global structure. The canvas then replaces the uniform mask in the masked autoregressive model, preserving global fidelity even with aggressive sampling regime.

Extending the MAR paradigm—continuous image tokens with random-order autoregressive generation—to video generation is promising, as evidenced by several recent studies [74, 7, 66]. However, a key challenge remains: MAR follows the MaskGIT [5] sampling paradigm, which begins from a fully masked image and autoregressively generates tokens in a random order. In early stages, when only a few tokens have been generated, the model must generate small sets of tokens at each step to maintain quality. The set size gradually increases only as more tokens are revealed, typically following a cosine schedule [5]. This creates a trade-off between fidelity and sampling speed: achieving high-quality generation typically requires significantly many sampling steps, while using only few steps often results in highly distorted outputs—especially for videos, where the temporal dimension further amplifies the issue. As illustrated in Figure 1, a simple video MAR model exhibits noticeable degradation after only 8 spatial autoregressive steps when generating a 32×32 latent frame (256×256 pixel frame), and the problem worsens as more frames are rolled out.

In this paper, we introduce **CanvasMAR**, a masked autoregressive video prediction model that sustains **higher fidelity** while using **fewer autoregressive steps** per frame. CanvasMAR operates through a two-level autoregressive process: temporally, it generates frames sequentially; spatially, it partitions each frame into image tokens and produces them set by set in an adaptive, motion-aware order. To maintain fidelity with few sampling steps, CanvasMAR first predicts the expectation of the future frame—the *canvas* (see Figure 1)—with only one model pass, and **replaces the uniform masks** in conventional masked generators with these non-uniform canvas tokens. The canvas captures the global structure of the target frame, enabling more aggressive yet coherent synthesis. To stabilize sampling, we further introduce an adaptive, motion-aware sampling order for next-frame generation: the model first synthesizes low-motion (near-stationary) regions, then progressively attends to highly dynamic areas, yielding an **easy-to-hard curriculum** that stabilizes autoregressive sampling and suppresses motion artifacts. We also apply compositional classifier-free guidance [22, 9, 1] that **reinforces the canvas and temporal conditioning**, further improving fidelity.

Experiments on standard video prediction datasets—including the lab-scale BAIR [51, 10], the moderate-scale UCF-101 [47], and the large-scale Kinetics-600 [4] benchmarks—validate the effectiveness of our approach. On BAIR and Kinetics-600, CanvasMAR delivers significant gains over prior autoregressive models such as MAGI [74] and rivals strong diffusion-based baselines like DFoT [45], underscoring the potential of global structure-aware masked generative models.

2 Related Work

Diffusion-Based Video Generative Models Diffusion models have demonstrated remarkable success in image generation [20, 41, 46], and have since been extended to video generation by treating the temporal axis as an additional dimension alongside the spatial ones [23, 2, 29, 56, 64, 38]. While effective, this paradigm generates entire videos as a whole, leading to high latency and limited interactivity for applications such as interactive simulators and game environments [14, 3]. To mitigate these issues, recent research has adopted an autoregressive approach, wherein videos are

generated sequentially by conditioning on previously generated clean frames. This design enables interactivity and supports longer, temporally coherent videos [42, 45, 65, 17, 71, 36, 34, 26]. However, these methods still remain diffusion-based, which generate individual frames or chunks as a whole through iterative frame-level denoising. In contrast, token-based autoregressive models that this work focuses on directly learn token-level distributions and produce image tokens, making them natively compatible with large language models and thus better suited for multimodal model designs [58, 27].

Autoregressive Video Generation An alternative line of work decomposes video frames into image tokens, shifting the generative task from producing full frames to predicting token sequences. Many approaches in this direction adopt quantized tokens, typically following paradigms such as next-token prediction [63, 24, 48, 60, 58, 27] or masked generative modeling [16, 67, 68, 3, 57, 53]. However, as observed by Yu et al. [68], these paradigms rely on carefully designed and trained VQ-VAEs [52, 12], which inevitably introduce quantization errors. Inspired by Masked Autoregressive Models (MAR) [32, 13], a growing body of work—including NOVA [7], VideoMAR [66], and MAGI [74]—adapts MAR to video generation, directly generating continuous image tokens and achieving remarkable generation performance.

Cascaded Generation The closest line of work that shares a similar idea with *canvas* is cascaded generation. Cascaded generation decomposes synthesis into stages that progressively increase resolution or temporal span [43, 15, 59, 21, 69, 72, 19]. Early image pipelines such as Cascaded Diffusion [21] train a low-resolution base model to capture global semantics before invoking specialized super-resolution and detail-enhancement modules. Such a coarse-to-fine strategy improves sample quality but requires separate networks, schedulers, and guidance hyperparameters per stage. The same underlying insight also extends to autoregressive approaches like VAR [49] for next-scale prediction and to masked generative models like HiMAR [72], though implementations differ. Recent large-scale video systems includes FlashVideo [69] and Waver [70] also adopt similar ideas for fast and high-fidelity video generation. However, **CanvasMAR is fundamentally different from these cascaded methods**: these works produce *multi-step generated samples* $x \sim p(x)$ for each low-resolution image/frame x, whereas CanvasMAR produces a *one-step predicted expectation* $\mathbb{E}[x]$ for the next frame x, which is computationally negligible and tailored for video MAR models.

3 Preliminaries

3.1 Factorizing Video Generation

In this paper, we focus on autoregressive video prediction. A video is denoted as $V \in \mathbb{R}^{N\times H\times W\times 3}$, which consists of N frames $[f^{(1)}, f^{(2)}, \ldots, f^{(N)}]$ of height H and width W. We employ latent generation [41]; however, for simplicity, we retain pixel-level notation. Autoregressive video generation models the video distribution via the following *temporal* factorization:

$$p(V) = \prod_{i=1}^{N} p\big(f^{(i)} \mid f^{(<i)}\big), \tag{1}$$

where $f^{(<i)} = \{f^{(1)}, f^{(2)}, \ldots, f^{(i-1)}\}$ denotes all preceding frames. Each frame $f^{(i)}$ is further decomposed into a sequence of image tokens in raster order, $f^{(i)} = [x_1^{(i)}, x_2^{(i)}, \ldots, x_n^{(i)}]$. In standard autoregressive generation [63, 24], tokens are generated in raster order token-by-token:

$$p(f^{(i)} \mid f^{(<i)}) = \prod_{j=1}^{n} p\big(x_j^{(i)} \mid f^{(<i)}, x_{<j}^{(i)}\big), \tag{2}$$

where $x_{<j}^{(i)} = \{x_1^{(i)}, x_2^{(i)}, \ldots, x_{j-1}^{(i)}\}$ denotes the tokens already generated within frame i.

Masked generative paradigms [5, 32, 7] generate tokens set by set in a random order. Specifically, let $\tau \in \mathcal{S}_n$ be a permutation of the n token indices, sampled uniformly for random-order MAR. The tokens are partitioned in this order into K sets of sizes $[n_1, n_2, \ldots, n_K]$, with cumulative index $s_k = \sum_{m=1}^{k} n_m$ and $s_0 = 0$, $s_K = n$. For a fixed order τ, the model uses the following *spatial* autoregression:

$$p(f^{(i)} \mid f^{(<i)}) = \prod_{k=1}^{K} p\big(X_k^{(i)} \mid f^{(<i)}, X_{<k}^{(i)}\big), \tag{3}$$

where $X_k^{(i)} = \{x_{\tau(r)}^{(i)} : s_{k-1} < r \leq s_k\}$ and $X_{<k}^{(i)} = \{X_1^{(i)}, \ldots, X_{k-1}^{(i)}\}$. Here $\mathcal{S}_n$ is the set of all permutations of $[1, \ldots, n]$. This sampling introduces stochasticity into the generation process.

3.2 Masked Autoregressive Models with Continuous Tokens

Unlike conventional autoregressive generative models that produce quantized image tokens [24, 63] using categorical distributions, MAR [32] directly generates continuous image tokens using a diffusion head, which models token distributions conditioned on token embeddings. Formally, the distribution in Equation (3) is further factorized as:

$$p\big(X_k^{(i)} \mid f^{(<i)}, X_{<k}^{(i)}\big) = \int p\big(X_k^{(i)} \mid Z_k^{(i)}\big)\, p\big(Z_k^{(i)} \mid f^{(<i)}, X_{<k}^{(i)}\big)\, \mathrm{d}Z_k^{(i)}. \tag{4}$$

where $Z_k^{(i)} = \{z_{\tau(s_{k-1}+1)}^{(i)}, z_{\tau(s_{k-1}+2)}^{(i)}, \ldots, z_{\tau(s_k)}^{(i)}\}$ denotes the token embeddings, and $X_{<k}^{(i)} = \{X_1^{(i)}, X_2^{(i)}, \ldots, X_{k-1}^{(i)}\}$. The conditional distribution $p(X_k^{(i)} \mid Z_k^{(i)}) = \prod_{r=s_{k-1}+1}^{s_k} p(x_{\tau(r)}^{(i)} \mid z_{\tau(r)}^{(i)})$ models per-token outputs given the embeddings, where correlations between tokens are captured when computing the embeddings. In practice, $p\big(Z_k^{(i)} \mid f^{(<i)}, X_1^{(i)}, X_2^{(i)}, \ldots, X_{k-1}^{(i)}\big)$ is a Dirac delta distribution centered at deterministic token-wise embeddings computed by a Vision Transformer [8], while $p(x_j^{(i)} \mid z_j^{(i)})$ is modeled by a lightweight MLP diffusion head that conditions on $z_j^{(i)}$ to generate $x_j^{(i)}$ [32].

4 Methods

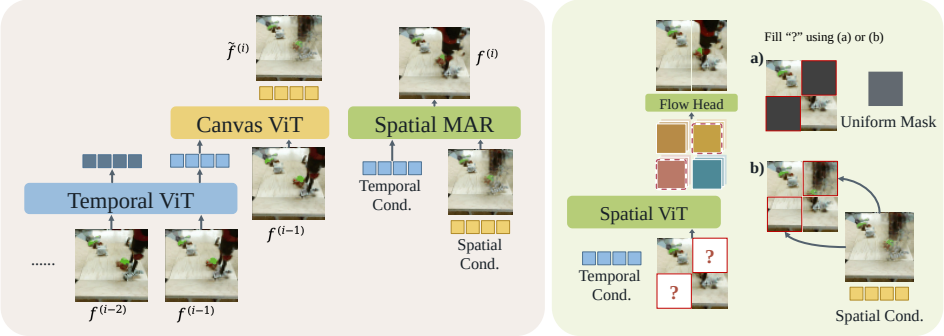


Figure 2: **Left:** Inference procedure of CanvasMAR. The framework consists of three modules: Temporal ViT, Canvas ViT, and Spatial MAR. The Temporal ViT encodes the historical frames and produces a temporal embedding that serves as a temporal condition for the subsequent modules. The Canvas ViT takes the temporal embedding together with the most recent frame to predict *canvas* in one step, i.e., the expectation of future frame, which, along with the temporal embedding, provides spatial conditioning for the Spatial MAR. **Right:** Training procedure of the Spatial MAR. A Spatial ViT learns to reconstruct randomly masked regions within frames. The slots marked "?" are filled with either a uniform learnable mask embedding in vanilla MAR (a), or the corresponding predicted canvas tokens in CanvasMAR (b).

4.1 Decoupled Temporal and Spatial Autoregression

Decoupled Causality. We design our model to generate videos through a decoupled temporal-spatial autoregressive framework, as illustrated in Figure 2. For clarity, we denote hidden representations by z and refer to the temporal and canvas latents as z_t and z_c. Specifically, CanvasMAR proceeds in two stages for generating the next frame i: **1)** a temporal embedding $z_t^{(i)} \in \mathbb{R}^{n \times d}$, where n denotes the number of frame tokens and d the hidden dimension, is autoregressively generated for frame i, using a Temporal ViT [8] that conditions on all past frames $f^{(<i)}$; **2)** this temporal embedding is provided as a conditioning sequence to the Spatial MAR, implemented as a masked autoregressive model [32], which generates image tokens set by set for the next frame $f^{(i)}$.

Attention Mask. The attention mask of the Temporal ViT follows the hybrid structure of Deng et al. [7] and Yu et al. [66]—causal across frames and bidirectional within each frame. This design preserves temporal causality while maintaining comprehensive spatial perception within frames, and it enables frame-level KV caching, further accelerating generation.

Training and Inference Efficiency. This temporal-spatial decoupling preserves both training and inference efficiency: **1)** During training, tokens within each frame can be independently masked, and all masked frames contribute to the loss in parallel, analogous to decoder-only language models [50, 68]. In contrast, models without decoupled autoregression can mask tokens only in the last frame of a sequence and compute loss of the only masked frame [66], leading to inefficient training. A potential remedy, as explored in MAGI [74], is to duplicate "clean" frames for each frame in the sequence and design a corresponding attention mask, which doubles the sequence length and substantially increases memory usage. **2)** During inference, the temporal embedding z_t captures motion information necessary to guide next-frame generation. As a result, spatial token generation only needs to attend to this fixed-length temporal embedding rather than all previously generated image tokens, whose size grows with sequence length.

4.2 Bridging Fast Temporal Autoregression and Slow Spatial Autoregression with Canvas

As shown in Figure 1, the absence of an initial global estimate of the target frame severely limits the size of the early token sets, i.e., n_k has to be small for early spatial steps k in Equation (3). In video generation, this further degrades both quality and speed. To address it, we introduce the *canvas* between temporal autoregression and spatial autoregression, as illustrated in Figure 2.

Specifically, after obtaining the temporal embedding $z_t^{(i)}$, we first pass it through the Canvas ViT module instead of feeding it directly to the Spatial MAR. This module predicts an estimate of the future-frame expectation $z_c^{(i)}$, which serves as an **explicit spatial condition** for subsequent modules. The Spatial MAR then receives both the temporal condition $z_t^{(i)}$ and the spatial canvas embedding $z_c^{(i)}$ to predict the next frame, as shown in the inference diagram of Figure 2.

Concretely, the Canvas ViT takes the temporal representation $z_t^{(i)}$ together with the most recent frame $f^{(i-1)}$ as input and outputs a canvas embedding $z_c^{(i)} \in \mathbb{R}^{n\times d}$, representing an initial guess of the next frame $f^{(i)}$. To supervise this process, we introduce a canvas reconstruction loss:

$$\mathcal{L}_{\text{canvas}} = \mathbb{E}_{f^{(<i+1)}}\Big[\sum_{j=1}^{n} \| f_\theta^{\text{canvas}}(z_{c,j}^{(i)}) - x_j^{(i)} \|_2^2\Big], \tag{5}$$

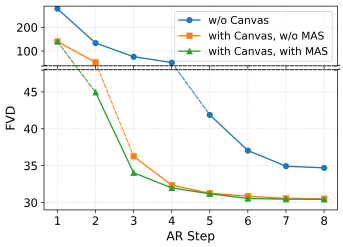
where $z_{c,j}^{(i)}$ denotes the j-th row of $z_c^{(i)}$, and f_θ^{canvas} is a simple linear projection that supervises each patch embedding $z_c^{(i)}$ to match the corresponding j-th patch in the ground-truth future frame $x_j^{(i)}$.

We note that, unlike the Spatial MAR—which is inherently slower and stochastic—the Canvas ViT is **fast and deterministic**: given the same temporal embedding $z_t^{(i)}$ and previous frame $f^{(i-1)}$, it always produces the same canvas with **negligible inference time** compared to the iterative decoding of Spatial MAR. Theoretically, **the optimal canvas approximates the expected future frame** $\mathbb{E}_{f^{(i)}\sim p(f^{(i)}|f^{(<i)})}[f^{(i)}]$ given past frames, effectively producing a superposition of plausible outcomes. This initialization allows the Spatial MAR to maintain global coherence early on and quickly "collapse" onto one plausible future frame.

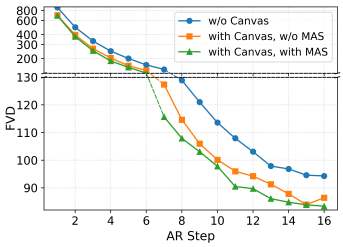
4.3 Motion-Aware Adaptive Sampling Order

The canvas approximates the conditional expectation of the next frame given the history. Consequently, highly dynamic regions can appear blurrier than stationary ones because they exhibit more uncertainty. Generating multiple patches in such blur regions simultaneously during early sampling steps can therefore be suboptimal, even with the canvas condition. We therefore design a motion-aware adaptive sampling order that prioritizes low-motion areas early and gradually attends to high-motion areas, implementing an easy-to-hard curriculum that stabilizes sampling and reduces motion artifacts.

Staticness Head. Specifically, we add a lightweight *staticness head* f_θ^{static} to the Canvas ViT that predicts a nonnegative scalar score per canvas patch, where larger values indicate lower motion. The



(a)



(b)

sCFG	tCFG	BAIR	UCF101
✓	✓	**29.58**	**86.03**
✗	✓	30.08	92.60
✓	✗	30.01	100.90
✗	✗	31.54	114.07

(c)

Figure 3: Component-level analysis. In (a) and (b), classifier-free guidance is not used, and MAS represents the motion-aware sampling order. (a) Effectiveness of canvas and motion-aware sampling on BAIR. (b) Effectiveness of canvas and motion-aware sampling on UCF-101. (c) Effectiveness of compositional classifier-free guidance, where sCFG represents spatial (canvas) CFG, and tCFG represents the temporal CFG. The metric is FVD; the lower, the better.

score is supervised with a training-time loss that resembles confidence prediction in AlphaFold [28]. Formally, let the per-patch canvas reconstruction error $\ell_j^{(i)} = \left\| f_\theta^{\text{canvas}}(z_{c,j}^{(i)}) - x_j^{(i)} \right\|_2^2$. We define the target staticness as $s_j^{(i)} = \exp\left(-\ell_j^{(i)}\right)$, where $0 \le s_j^{(i)} \le 1$. We then train with

$$\mathcal{L}_{\text{static}} = \mathbb{E}_{f^{(<i+1)}}\Big[\sum_{j=1}^{n}\left(f_\theta^{\text{static}}(z_{c,j}^{(i)}) - \text{stop}(s_j^{(i)})\right)^2\Big], \tag{6}$$

where f_θ^{static} denotes a simple motion head (a linear layer followed by a sigmoid) that probes each patch embedding to predict staticness, and $\text{stop}(\cdot)$ indicates stop-gradient operation. Intuitively, with f^{canvas} predicting the expectation of a future patch, f^{static} estimates **a confidence proxy related to the variance of the corresponding patch**.

MAS Sampling Order. In MAR [32], sampling proceeds in a purely random order. Although this aligns with the training distribution of masked generative modeling [32], it is suboptimal for next-frame generation because of the function-reuse approximation error (i.e., reusing a function evaluation for both regions with high and low motion) [73]. We therefore treat the staticness scores as confidence values that bias the sampling toward low-motion regions in early steps and high-dynamic regions in later steps. We follow the temperature-annealing method of MaskGiT [5] to retain stochasticity.

4.4 Compositional Classifier-Free Guidance

Guidance is known to significantly improve generation fidelity [22, 9, 1]. In CanvasMAR, the canvas provides a significant spatial condition in addition to the temporal one. We therefore adopt compositional classifier-free guidance to enhance robustness under this compositional conditioning.

For the Spatial MAR, compositional CFG up-weights the spatial canvas condition $z_c^{(i)}$ and temporal condition $z_t^{(i)}$ with guidance scales w_s and w_t, respectively. This yields the following score combination, with the sampling time omitted for brevity:

$$\mathbf{s}^{(w_s,w_t)} = \nabla_{f^{(i)}} \log p^{(w_s,w_t)}\left(f^{(i)} \mid z_c^{(i)}, z_t^{(i)}\right) = \mathbf{s}_\varnothing + w_t\left(\mathbf{s}_t - \mathbf{s}_\varnothing\right) + w_s\left(\mathbf{s}_{t,s} - \mathbf{s}_t\right), \tag{7}$$

where $\mathbf{s}_\varnothing = \nabla \log p(f^{(i)})$ is the unconditional score, $\mathbf{s}_t = \nabla \log p(f^{(i)} \mid z_t^{(i)})$ is the temporal-only score, and $\mathbf{s}_{t,s} = \nabla \log p(f^{(i)} \mid z_t^{(i)}, z_c^{(i)})$ is the fully conditioned score.

We evaluate three forward passes of the Spatial MAR head that differ only in their conditioning inputs (unconditional, temporal-only, and temporal+spatial), and apply the same combination in Equation (7) to the flow-head outputs before updating tokens. The derivation and training config are provided in Appendix B.

5 Experiments

5.1 Settings

Model Implementation. We implement CanvasMAR on the NOVA [7] codebase with three additional changes: (1) a flow head trained with flow matching [33, 35] instead of DDPM [20]; (2) expert LayerNorm [64] that treats the conditioning sequence, clean tokens, and mask tokens separately, or MM-DiT [11] that treats the clean and mask tokens separately; and (3) removal of the Scale-and-Shift layer for the temporal embedding [7].

Training Objective. According to Section 4, training CanvasMAR involves three objectives: (1) the canvas-masked reconstruction objective for the Spatial MAR head $\mathcal{L}_{\text{MAR}}$ [7]; (2) the canvas reconstruction loss $\mathcal{L}_{\text{canvas}}$ in Equation (5); and (3) the staticness prediction loss $\mathcal{L}_{\text{static}}$ in Equation (6). We therefore optimize a weighted objective:

$$\mathcal{L}_{\text{total}} \;=\; \mathcal{L}_{\text{MAR}} \;+\; \lambda_c\,\mathcal{L}_{\text{canvas}} \;+\; \lambda_s\,\mathcal{L}_{\text{static}}, \tag{8}$$

and set both weights λ_c and λ_s to 0.1 in all experiments.

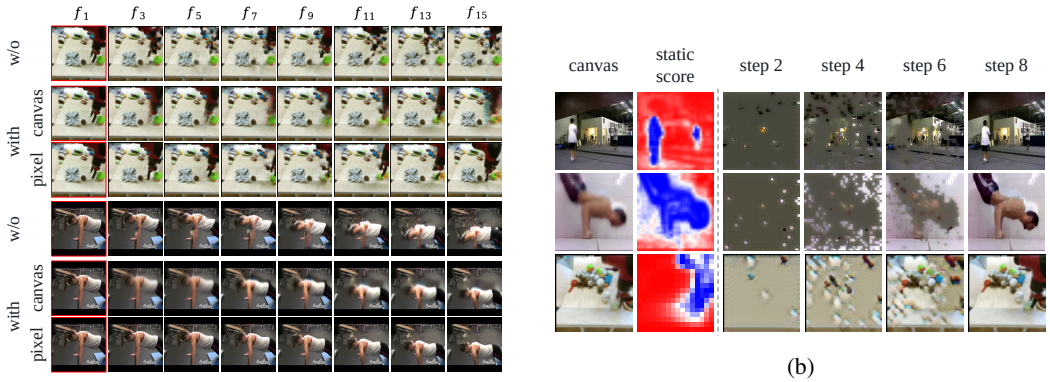
Tasks and Datasets. We evaluate CanvasMAR on video prediction tasks, where one or more context frames are given and the goal is to forecast subsequent frames. Our experiments span three datasets of increasing scale and complexity: 1) **BAIR.** The BAIR dataset [51, 10] is a toy benchmark containing 43K training videos and 256 evaluation videos of a robot arm pushing objects within a static camera view. 2) **UCF-101**. UCF-101 [47] is a moderate-scale real-world dataset with over 13,000 clips across 101 human action classes. 3) **Kinetics-600.** The Kinetics-600 dataset [4] is a large-scale collection of diverse real-world videos spanning 600 human action categories. It contains approximately 400K training clips and serves as a challenging benchmark for large-scale video generation. Since UCF-101 is typically used for conditional generation without context frames [74], which CanvasMAR does not currently support (see clarification in Section 6), we report comparisons only against the baseline method we implemented. For BAIR and Kinetics-600, we provide system-level comparisons with prior art. See Appendix A for more details.

5.2 Component-Level Analysis

Effectiveness of the canvas. Since the canvas module is central to CanvasMAR, we first study its impact. To achieve this, we train a model identical to CanvasMAR but without the canvas component from scratch, and compare the two under identical experimental settings except for the use of the canvas. Results on the BAIR and UCF-101 datasets (Figure 3 (a) and (b), blue and orange lines) show that the **canvas-equipped model consistently outperforms its non-canvas counterpart** across all autoregressive step counts. Notably, larger gains appear at small step counts (e.g., 2–4 on the BAIR dataset and 2–8 on the UCF101 dataset), supporting the importance of a global spatial prior in few-step settings. Qualitative comparisons in Figure 4 (a) demonstrate that the model without the canvas struggles to maintain global structure and object coherence, whereas the canvas-based model preserves good spatial and temporal consistency in few-step settings. Furthermore, CanvasMAR can generate long videos by autoregressively rolling out frames with stable temporal coherence; see Appendix A for a 60-frame example.

Effectiveness of adaptive motion-aware sampling. The adaptive motion-aware sampling strategy enables an easy-to-hard curriculum for next-frame generation. As shown in Figure 3 (a) and (b) (orange and green lines), it **consistently improves performance in fewer-step settings**—the case we focus on for efficient sampling. With more sampling steps, each batch of simultaneously generated tokens becomes smaller, reducing conflicts even without adaptive sampling and therefore narrowing the gap as we expected. Qualitative examples in Figure 4 (b) visualize the predicted staticness map and its induced ordering, showcasing that the lightweight static head successfully captures motion information. After that, low-motion regions (warm colors) are synthesized first, while high-motion regions (cool colors) are deferred to later steps as expected.

Effectiveness of compositional CFG. Figure 3 (c) summarizes the impact of the spatial and temporal branches of compositional classifier-free guidance. Disabling either branch degrades performance, validating the importance of both guidance for autoregressive video generation.



(a) (b)

Figure 4: (a) Qualitative examples of video clips generated by CanvasMAR (bottom two rows in each example) and its non-canvas variant (top row). Upper: 2-step generation on BAIR (16 × 16 latents); Lower: 8-step generation on UCF-101 (32 × 32 latents). The canvas-based model better preserves global structure and object coherence. (b) Qualitative illustration of the motion-aware staticness prediction and adaptive sampling schedule. Warmer colors denote regions predicted to remain relatively static and are therefore generated earlier, while cooler colors indicate high-motion regions deferred to later autoregressive steps.

Table 1: System-level comparison on BAIR. FVD↓ at 64×64. DFVD reports debiased FVD [67].

Type	Method	FVD↓	DFVD↓
GAN	DVD-GAN-FP [6]	109.8	-
	TrIVD-GAN-FP [37]	**103**	-
Diffusion	MCVD [54]	90	-
	RaMViD [25]	84	-
	VDM [23]	**66.9**	-
Spatial AR	Transframer [39]	100	-
	CCVS [30]	99	-
	Phenaki [53]	97	-
	MaskViT [16]	94	-
	NÜWA [62]	87	-
	MAGVIT-L [67]	**62**	31
	MAGVIT-B [67]	76	48
	Ours ($N_{AR} = 4$)	66.0	**26.6**
	Ours ($N_{AR} = 6$)	65.3	26.5
	Ours ($N_{AR} = 8$)	65.2	26.4

Table 2: System-level comparison on Kinetics-600. FVD↓ at 64×64 (methods with $*$ may be evaluated at 128×128).

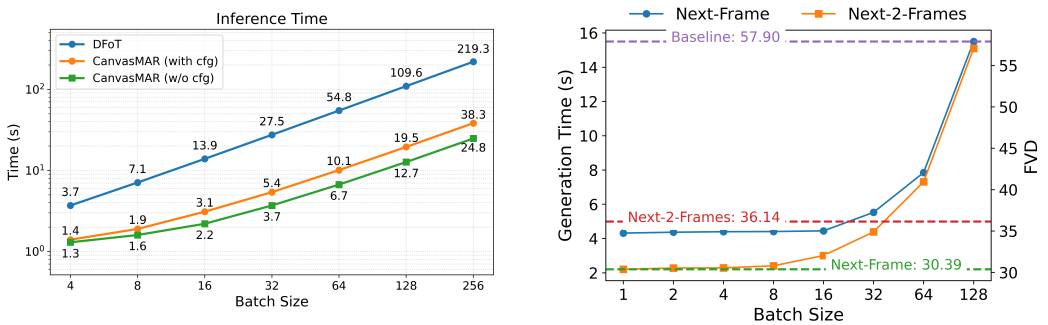
Type	Method	FVD↓
GAN	DVD-GAN-FP [6]	69.1
	TrIVD-GAN-FP [37]	**25.74**
Diffusion	VDM [23]	16.2
	RaMViD [25]	16.5
	Rolling Diffusion [42]	5.2
	DFoT* [45]	**4.3**
Spatial AR	Video Transformer [61]	170
	CogVideo [24]	109.2
	Video VQ-VAE [55]	64.3
	CCVS [30]	55.0
	Phenaki* [53]	36.4
	Transframer [39]	25.4
	MAGVIT [67]	9.9
	MAGI [74]	11.5
	Ours ($N_{AR} = 10$)	**6.3**
	Ours ($N_{AR} = 11$)	6.3
	Ours ($N_{AR} = 12$)	6.2

5.3 System-Level Comparisons

We compare CanvasMAR with well-known diffusion-based and autoregressive video models. Note that each method is **categorized by its spatial-level generation mechanism**—either refining full frames in diffusion manner or generating the frames tokens by tokens—which more directly determines visual quality than the temporal generation paradigm (generating an entire video as a whole versus rolling out frames sequentially).

As shown in Table 1, CanvasMAR achieves the second-best FVD on BAIR among all compared models and surpasses MAGVIT **under the debiased evaluation (DFVD)**, achieving **the best performance**. Note that the debiased setting is more convincing in statistics due to the much wider sample distributions, as detailed in Appendix A.

On the larger Kinetics-600 dataset in Table 2, CanvasMAR achieves the best performance among the compared spatial autoregressive models. Compared with MAGI [74], a close baseline that is both temporally and spatially autoregressive but does not use a canvas, CanvasMAR demonstrates the effectiveness of our design. CanvasMAR also rivals advanced diffusion models, yet with much



(a) (b)

Figure 5: (a) Inference latency comparison of CanvasMAR and DFoT on K600. CanvasMAR achieves much lower latency than DFoT due to the autoregressive paradigm and fewer-step sampling. (b) Performance and speedup effect of next-group prediction models. No classifier-free guidance is used.

better efficiency. To illustrate this, we compare inference latency with the advanced diffusion model DFoT [45] in Figure 5 (a). CanvasMAR is about **5.7** $\times$ **faster** at the largest batch size (GPU fully utilized), highlighting the efficiency of few-step autoregressive sampling. Figure 5 (a) reports latency—the time until the first frame appears—which is often the most relevant metric for interactive applications. We further show that even when considering total generation time (see Figure 7 in Appendix A), CanvasMAR is still about **2.7** $\times$ faster.

As shown in the gray lines of Table 1 and Table 2, further increasing the sampling steps of CanvasMAR on both datasets yields only minimal performance gains, indicating that **only a few steps are sufficient** for CanvasMAR. In contrast, the non-canvas baseline shown in Figure 3 fails to surpass CanvasMAR even with additional steps.

5.4 Multi-Frame Prediction

In diffusion-based autoregressive models [65, 45, 26, 34], predicting multiple upcoming frames (e.g., chunks) jointly is common. For token-based autoregressive models, models typically generate frames one at a time [7], where next-group prediction often leads to severe performance degradation. However, with the introduction of the canvas, we find next-group frame prediction to be stable, enabling faster generation with limited quality loss.

Empirically, we fine-tune a next-group frame prediction model (with a group size of 2) from a well-trained next-frame prediction model on BAIR, requiring only about 10% of the original training cost to converge. This indicates that substantial knowledge is transferred between next-frame and next-group settings. See Appendix B for more details of the model architecture.

As shown in Figure 5 (b), the next-2-frame model exhibits **substantial speedups at smaller batch sizes** compared to the next-frame model and converges to similar throughput at higher batch sizes. When the batch size becomes large, both models fully utilize the available GPU resources and, thanks to KV caching, reach comparable parallel efficiency. In practical scenarios, however, users typically generate only a few videos and expect low latency, where the next-2-frame prediction model is particularly advantageous. Furthermore, we train a next-2-frame prediction model without the canvas (purple curve), and our canvas-based model performs significantly better when transferred to next-group prediction because the canvas provides strong structural priors for multiple future frames.

6 Conclusion

We presented CanvasMAR, a masked autoregressive video prediction framework that improves both generation quality and sampling efficiency through canvas-based conditioning. By predicting the blurred expectation of the future frame before spatial token generation, CanvasMAR bridges fast temporal autoregression and slower spatial token autoregression, allowing conventional MAR models to use fewer sampling steps while preserving fidelity. The motion-aware sampling order further improves sampling quality by prioritizing low-motion regions during spatial token generation,

alleviating conflicts that arise when many tokens are sampled in parallel. Experiments on BAIR and Kinetics-600 show that CanvasMAR achieves competitive or superior FVD scores with fewer autoregressive steps and outperforms most prior autoregressive baselines. Our experiments focus on video prediction, where previous frames provide natural canvas guidance; extending CanvasMAR to from-scratch video generation is straightforward by generating the first frame without the canvas mechanism, similar to NOVA [7], but is not the focus of this paper. One potential limitation is that, for highly dynamic sequences, high motion variance can make the canvas overly blurred and less informative for guiding the subsequent Spatial MAR, leading to artifacts as shown in Figure 6 in Appendix A. Scaling up the model may improve this self-correction ability, especially since our largest Spatial MAR has only half the layers of NOVA [7]. Future work will also explore scaling CanvasMAR to text-conditioned and multimodal video generation.

Acknowledgement

This work is supported by the Beijing Natural Science Foundation (Z260022) and National Natural Science Foundation of China (62561160152).

References

[1] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18392–18402, 2023.

[2] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, et al. Video generation models as world simulators. *OpenAI Blog*, 1(8):1, 2024.

[3] Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. In *Forty-first International Conference on Machine Learning*, 2024.

[4] Joao Carreira, Eric Noland, Andras Banki-Horvath, Chloe Hillier, and Andrew Zisserman. A short note about kinetics-600. *arXiv preprint arXiv:1808.01340*, 2018.

[5] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. Maskgit: Masked generative image transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11315–11325, 2022.

[6] Aidan Clark, Jeff Donahue, and Karen Simonyan. Adversarial video generation on complex datasets. *arXiv preprint arXiv:1907.06571*, 2019.

[7] 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.

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[9] Yilun Du and Leslie Kaelbling. Compositional generative modeling: A single model is not all you need. *arXiv preprint arXiv:2402.01103*, 2024.

[10] Frederik Ebert, Chelsea Finn, Alex X Lee, and Sergey Levine. Self-supervised visual planning with temporal skip connections. *CoRL*, 12(16):23, 2017.

[11] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

[12] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021.

[13] Lijie Fan, Tianhong Li, Siyang Qin, Yuanzhen Li, Chen Sun, Michael Rubinstein, Deqing Sun, Kaiming He, and Yonglong Tian. Fluid: Scaling autoregressive text-to-image generative models with continuous tokens. *ArXiv*, abs/2410.13863, 2024.

[14] Ruili Feng, Han Zhang, Zhantao Yang, Jie Xiao, Zhilei Shu, Zhiheng Liu, Andy Zheng, Yukun Huang, Yu Liu, and Hongyang Zhang. The matrix: Infinite-horizon world generation with real-time moving control. *arXiv preprint arXiv:2412.03568*, 2024.

[15] Jiatao Gu, Shuangfei Zhai, Yizhe Zhang, Joshua M Susskind, and Navdeep Jaitly. Matryoshka diffusion models. In *The Twelfth International Conference on Learning Representations*, 2023.

[16] Agrim Gupta, Stephen Tian, Yunzhi Zhang, Jiajun Wu, Roberto Martín-Martín, and Li Fei-Fei. Maskvit: Masked visual pre-training for video prediction. *arXiv preprint arXiv:2206.11894*, 2022.

[17] Agrim Gupta, Lijun Yu, Kihyuk Sohn, Xiuye Gu, Meera Hahn, Fei-Fei Li, Irfan Essa, Lu Jiang, and José Lezama. Photorealistic video generation with diffusion models. In *European Conference on Computer Vision*, pages 393–411. Springer, 2024.

[18] Jian Han, Jinlai Liu, Yi Jiang, Bin Yan, Yuqi Zhang, Zehuan Yuan, Bingyue Peng, and Xiaobing Liu. Infinity: Scaling bitwise autoregressive modeling for high-resolution image synthesis. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 15733–15744, 2025.

[19] Jingwen He, Tianfan Xue, Dongyang Liu, Xinqi Lin, Peng Gao, Dahua Lin, Yu Qiao, Wanli Ouyang, and Ziwei Liu. Venhancer: Generative space-time enhancement for video generation. *arXiv preprint arXiv:2407.07667*, 2024.

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

[21] Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *Journal of Machine Learning Research*, 23(47):1–33, 2022.

[22] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

[23] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Advances in neural information processing systems*, 35:8633–8646, 2022.

[24] Wenyi Hong, Ming Ding, Wendi Zheng, Xinghan Liu, and Jie Tang. Cogvideo: Large-scale pretraining for text-to-video generation via transformers. *arXiv preprint arXiv:2205.15868*, 2022.

[25] Tobias Höppe, Arash Mehrjou, Stefan Bauer, Didrik Nielsen, and Andrea Dittadi. Diffusion models for video prediction and infilling. *arXiv preprint arXiv:2206.07696*, 2022.

[26] Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. *arXiv preprint arXiv:2506.08009*, 2025.

[27] Yang Jin, Zhicheng Sun, Kun Xu, Liwei Chen, Hao Jiang, Quzhe Huang, Chengru Song, Yuliang Liu, Di Zhang, Yang Song, et al. Video-lavit: Unified video-language pre-training with decoupled visual-motional tokenization. *arXiv preprint arXiv:2402.03161*, 2024.

[28] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *nature*, 596(7873):583–589, 2021.

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

[30] Guillaume Le Moing, Jean Ponce, and Cordelia Schmid. Ccvs: Context-aware controllable video synthesis. *Advances in Neural Information Processing Systems*, 34:14042–14055, 2021.

[31] Tianhong Li, Huiwen Chang, Shlok Mishra, Han Zhang, Dina Katabi, and Dilip Krishnan. Mage: Masked generative encoder to unify representation learning and image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2142–2152, 2023.

[32] 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.

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

[34] Kunhao Liu, Wenbo Hu, Jiale Xu, Ying Shan, and Shijian Lu. Rolling forcing: Autoregressive long video diffusion in real time. *arXiv preprint arXiv:2509.25161*, 2025.

[35] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

[36] Zhiheng Liu, Xueqing Deng, Shoufa Chen, Angtian Wang, Qiushan Guo, Mingfei Han, Zeyue Xue, Mengzhao Chen, Ping Luo, and Linjie Yang. Worldweaver: Generating long-horizon video worlds via rich perception. *arXiv preprint arXiv:2508.15720*, 2025.

[37] Pauline Luc, Aidan Clark, Sander Dieleman, Diego de Las Casas, Yotam Doron, Albin Cassirer, and Karen Simonyan. Transformation-based adversarial video prediction on large-scale data. *arXiv preprint arXiv:2003.04035*, 2020.

[38] Xin Ma, Yaohui Wang, Gengyun Jia, Xinyuan Chen, Ziwei Liu, Yuan-Fang Li, Cunjian Chen, and Yu Qiao. Latte: Latent diffusion transformer for video generation. *arXiv preprint arXiv:2401.03048*, 2024.

[39] Charlie Nash, Joao Carreira, Jacob Walker, Iain Barr, Andrew Jaegle, Mateusz Malinowski, and Peter Battaglia. Transframer: Arbitrary frame prediction with generative models. *arXiv preprint arXiv:2203.09494*, 2022.

[40] NVIDIA et al. Cosmos world foundation model platform for physical AI. *arXiv preprint arXiv:2501.03575*, 2025.

[41] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[42] David Ruhe, Jonathan Heek, Tim Salimans, and Emiel Hoogeboom. Rolling diffusion models, 2024.

[43] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement. *IEEE transactions on pattern analysis and machine intelligence*, 45(4):4713–4726, 2022.

[44] Ivan Skorokhodov, Sergey Tulyakov, and Mohamed Elhoseiny. Stylegan-v: A continuous video generator with the price, image quality and perks of stylegan2. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3626–3636, 2022.

[45] Kiwhan Song, Boyuan Chen, Max Simchowitz, Yilun Du, Russ Tedrake, and Vincent Sitzmann. History-guided video diffusion. *arXiv preprint arXiv:2502.06764*, 2025.

[46] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

[47] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

[48] Peize Sun, Yi Jiang, Shoufa Chen, Shilong Zhang, Bingyue Peng, Ping Luo, and Zehuan Yuan. Autoregressive model beats diffusion: Llama for scalable image generation. *arXiv preprint arXiv:2406.06525*, 2024.

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

[50] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

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

[52] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

[53] Ruben Villegas, Mohammad Babaeizadeh, Pieter-Jan Kindermans, Hernan Moraldo, Han Zhang, Mohammad Taghi Saffar, Santiago Castro, Julius Kunze, and Dumitru Erhan. Phenaki: Variable length video generation from open domain textual description. *arXiv preprint arXiv:2210.02399*, 2022.

[54] Vikram Voleti, Alexia Jolicoeur-Martineau, and Chris Pal. Mcvd-masked conditional video diffusion for prediction, generation, and interpolation. *Advances in neural information processing systems*, 35:23371–23385, 2022.

[55] Jacob Walker, Ali Razavi, and Aäron van den Oord. Predicting video with vqvae. *arXiv preprint arXiv:2103.01950*, 2021.

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

[57] Xiaofeng Wang, Zheng Zhu, Guan Huang, Boyuan Wang, Xinze Chen, and Jiwen Lu. Worlddreamer: Towards general world models for video generation via predicting masked tokens. *arXiv preprint arXiv:2401.09985*, 2024.

[58] Xinlong Wang, Xiaosong Zhang, Zhengxiong Luo, Quan Sun, Yufeng Cui, Jinsheng Wang, Fan Zhang, Yueze Wang, Zhen Li, Qiying Yu, et al. Emu3: Next-token prediction is all you need. *arXiv preprint arXiv:2409.18869*, 2024.

[59] Yaohui Wang, Xinyuan Chen, Xin Ma, Shangchen Zhou, Ziqi Huang, Yi Wang, Ceyuan Yang, Yinan He, Jiashuo Yu, Peiqing Yang, et al. Lavie: High-quality video generation with cascaded latent diffusion models. *International Journal of Computer Vision*, 133(5):3059–3078, 2025.

[60] Yuqing Wang, Tianwei Xiong, Daquan Zhou, Zhijie Lin, Yang Zhao, Bingyi Kang, Jiashi Feng, and Xihui Liu. Loong: Generating minute-level long videos with autoregressive language models. *arXiv preprint arXiv:2410.02757*, 2024.

[61] Dirk Weissenborn, Oscar Täckström, and Jakob Uszkoreit. Scaling autoregressive video models. *arXiv preprint arXiv:1906.02634*, 2019.

[62] Chenfei Wu, Jian Liang, Lei Ji, Fan Yang, Yuejian Fang, Daxin Jiang, and Nan Duan. Nüwa: Visual synthesis pre-training for neural visual world creation. In *European conference on computer vision*, pages 720–736. Springer, 2022.

[63] Wilson Yan, Yunzhi Zhang, Pieter Abbeel, and Aravind Srinivas. Videogpt: Video generation using vq-vae and transformers. *arXiv preprint arXiv:2104.10157*, 2021.

[64] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024.

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

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

[67] Lijun Yu, Yong Cheng, Kihyuk Sohn, José Lezama, Han Zhang, Huiwen Chang, Alexander G Hauptmann, Ming-Hsuan Yang, Yuan Hao, Irfan Essa, et al. Magvit: Masked generative video transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10459–10469, 2023.

[68] Lijun Yu, José Lezama, Nitesh B Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Vighnesh Birodkar, Agrim Gupta, Xiuye Gu, et al. Language model beats diffusion–tokenizer is key to visual generation. *arXiv preprint arXiv:2310.05737*, 2023.

[69] Shilong Zhang, Wenbo Li, Shoufa Chen, Chongjian Ge, Peize Sun, Yida Zhang, Yi Jiang, Zehuan Yuan, Binyue Peng, and Ping Luo. Flashvideo: Flowing fidelity to detail for efficient high-resolution video generation. *arXiv preprint arXiv:2502.05179*, 2025.

[70] Yifu Zhang, Hao Yang, Yuqi Zhang, Yifei Hu, Fengda Zhu, Chuang Lin, Xiaofeng Mei, Yi Jiang, Bingyue Peng, and Zehuan Yuan. Waver: Wave your way to lifelike video generation. *arXiv preprint arXiv:2508.15761*, 2025.

[71] Yuan Zhang, Jiacheng Jiang, Guoqing Ma, Zhiying Lu, Haoyang Huang, Jianlong Yuan, and Nan Duan. Generative pre-trained autoregressive diffusion transformer. *arXiv preprint arXiv:2505.07344*, 2025.

[72] Guangting Zheng, Yehao Li, Yingwei Pan, Jiajun Deng, Ting Yao, Yanyong Zhang, and Tao Mei. Hierarchical masked autoregressive models with low-resolution token pivots. *arXiv preprint arXiv:2505.20288*, 2025.

[73] Kaiwen Zheng, Yongxin Chen, Hanzi Mao, Ming-Yu Liu, Jun Zhu, and Qinsheng Zhang. Masked diffusion models are secretly time-agnostic masked models and exploit inaccurate categorical sampling. *arXiv preprint arXiv:2409.02908*, 2024.

[74] Deyu Zhou, Quan Sun, Yuang Peng, Kun Yan, Runpei Dong, Duomin Wang, Zheng Ge, Nan Duan, and Xiangyu Zhang. Taming teacher forcing for masked autoregressive video generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 7374–7384, 2025.

A Experiment Details

Training and Model Details. All models are trained on 8 NVIDIA A100 GPUs, each with 80GB memory. Comprehensive model configurations and hyperparameters are listed in Table 3.

Component	BAIR	UCF-101	Kinetics-600
Architecture			
Temporal ViT (dim / layers / patch)	768 / 8 / 2	1024 / 16 / 2	1024 / 16 / 2
Canvas ViT (dim / layers / patch)	768 / 4 / 1	1024 / 8 / 1	1024 / 8 / 1
Spatial MAR (dim / layers / depth / patch)	768 / 8 / 2 / 1	1024 / 16 / 4 / 1	1024 / 16 / 4 / 1
Additional Architecture	Expert Layernorm [64]	Expert Layernorm [64]	MM-DiT [11]
Flow decoder (dim / layers)	1280 / 3	1280 / 3	1280 / 3
VAE model	Causal 3D CNN*	Cosmos Tokenizer [40]	Non-causal 3D CNN [45]
MAR autoregressive steps#	4	8	10
Total parameters	211M	733M	733M
Training			
Batch size	192	64	192
Epochs	600	800	270
Resolution	$17 \times 64 \times 64$	$17 \times 256 \times 256$	$17 \times 128 \times 128$
Latent compression ratio	$\{1, 2\} \times 4 \times 4$	$\{1, 4\} \times 8 \times 8$	$\{1, 4\} \times 8 \times 8$
Latent resolution	$9 \times 16 \times 16$	$5 \times 32 \times 32$	$5 \times 16 \times 16$
Noise augmentation (r)	$[0.2, 0.6]$	$[0.2, 0.6]$	$[0.2, 0.6]$
Noise augmentation (r')	$[0.0, 0.3]$	$[0.0, 0.3]$	$[0.0, 0.3]$
Sampling			
Noise augmentation (r)	0.3	0.3	0.4
Noise augmentation (r')	0.2	0.2	0.2
Confidence temperature	6.5	6.5	6.5
Flow-head steps	30	30	30
Canvas CFG	1.5	1.5	2.25
Temporal CFG	1.2	1.2	1.1
Evaluation Resolution	$16 \times 64 \times 64$	$16 \times 256 \times 256$	$16 \times 64 \times 64$

Table 3: Model configurations for BAIR, UCF-101, and Kinetics-600. * Trained by ourselves with the codebase of Song et al. [45]. # Otherwise specified, we use this step number for sampling.

Sampling Protocols. As noted by Skorokhodov et al. [44], the sampling procedure can significantly influence FVD scores. To ensure fair comparisons, we follow the sampling strategies used in prior work:

- **BAIR**: For each of the 256 test videos, we use the first frame as the conditioning input and the subsequent 16-frame segment as the real distribution. We then draw K synthetic 16-frame clips per conditioning frame, resulting in $256 \times K$ generated clips in the fake distribution.

  Following Yu et al. [67], we also report the *debiased FVD* by randomly sampling one conditioning frame per test video and redefining the real distribution accordingly. We repeat this procedure 100 times to reduce sampling bias, with each repetition yielding $256 \times K$ generated clips.

  Unless otherwise specified, we set $K = 20$ and report only the debiased FVD. For system-level comparisons, we align with prior work by using $K = 100$ and reporting both standard and debiased FVD scores.

- **UCF-101**: We randomly sample 5K test videos and, for each, select a random 16-frame clip to define the real distribution, using the preceding 1 frame as a condition to generate one 16-frame clip for the fake distribution.

- **Kinetics-600**: We randomly sample K test videos and, for each, select a random 16-frame clip to define the real distribution. We use the first 5 frames of each clip as conditions and predict the remaining 11 frames, forming a 16-frame clip for the fake distribution. For the system-level study, we set $K = 50{,}000$, following prior work. For ablation studies, we set $K = 10{,}000$.

Inference Time. For completeness, we report the full generation time comparison in Figure 7.

Failure Case of CanvasMAR. As discussed in the conclusion, CanvasMAR may show sub-optimal performance on highly dynamic video predictions, where high motion variance can make the canvas less informative. See Figure 6.

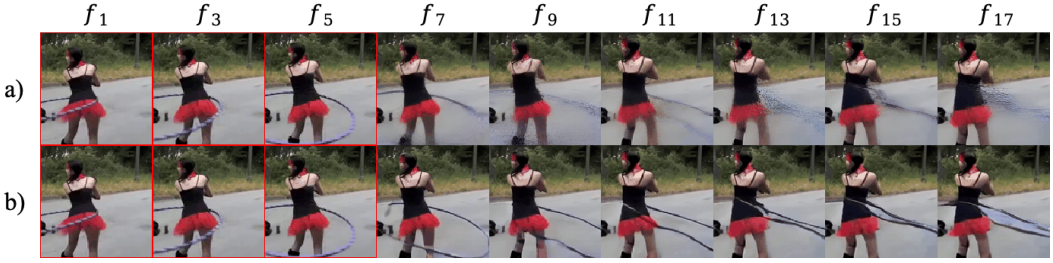


Figure 6: Failure case of CanvasMAR for videos exhibiting substantial motion. a) The predicted canvas. b) The generated frame.

B Additional Technical Details

Compositional Classifier-Free Guidance. For the Spatial MAR, the generative distribution is $p(f^{(i)} \mid z_c^{(i)}, z_t^{(i)})$, where $z_c^{(i)}$ denotes the canvas condition and $z_t^{(i)}$ denotes the temporal condition. We decompose the distribution as

$$p\big(f^{(i)} \mid z_c^{(i)}, z_t^{(i)}\big) \propto p\big(f^{(i)}\big)\, p\big(z_t^{(i)} \mid f^{(i)}\big) p\big(z_c^{(i)} \mid f^{(i)}, z_t^{(i)}\big).$$

Applying compositional CFG [9, 1] up-weights both the spatial and temporal posteriors:

$$p^{(w_s,w_t)}\big(f^{(i)} \mid z_c^{(i)}, z_t^{(i)}\big) \propto p\big(f^{(i)}\big)\, p\big(z_t^{(i)} \mid f^{(i)}\big)^{w_t} p\big(z_c^{(i)} \mid f^{(i)}, z_t^{(i)}\big)^{w_s}, \tag{9}$$

where w_s and w_t denote the guidance scales for the spatial and temporal conditions. Taking gradients of Equation (9) gives the score combination in Equation (7).

To enable CFG at inference, we adopt classifier-free training by randomly dropping conditions: with 5% probability we replace the spatial condition with a uniform mask [5]; with 5% probability we replace the temporal sequence with a learnable temporal vector; with 5% probability we drop both; otherwise we keep both.

Canvas Augmentation. The canvas is predicted in a single pass, unlike iterative generative models that refine outputs across multiple steps. This design introduces two primary sources of error. First, inaccuracies in the previously generated frame can mislead the Canvas ViT, compounding errors in the next canvas prediction. Second, even with an accurate previous frame, the one-pass Canvas ViT may produce overly blurred or imprecise canvases, which can confuse the Spatial MAR and amplify artifacts.

To mitigate these effects, we additionally adopt a noise-based augmentation strategy inspired by cascaded generation [69, 21]. When feeding the most recent frame $f^{(i-1)}$ into the Canvas ViT, we add Gaussian noise:

$$f_{\text{aug}}^{(i-1)} = f^{(i-1)} \cdot (1 - r) + \epsilon \cdot r, \tag{10}$$

where ϵ denotes Gaussian noise and r is an interpolation coefficient. Similarly, we perturb the canvas embedding:

$$z_{c,\text{aug}}^{(i)} = z_c^{(i)} \cdot (1 - r') + \epsilon' \cdot r', \tag{11}$$

where ϵ' and r' denote another Gaussian noise term and interpolation coefficient. These augmentations simulate instability in previously generated frames and canvases, encouraging both the Canvas ViT and the Spatial MAR to remain robust under imperfect conditions. During training, we sample r and r' uniformly from a predefined interval; during inference, we fix both to a constant. See Table 3 for the specific values used in our experiments.

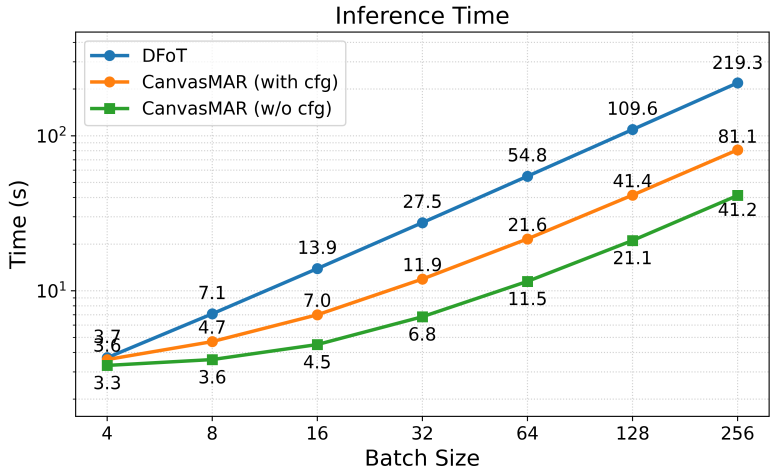


Figure 7: Inference time comparison of CanvasMAR and DFoT on Kinetics-600. CanvasMAR achieves much lower generation time than DFoT due to the fewer-step sampling.

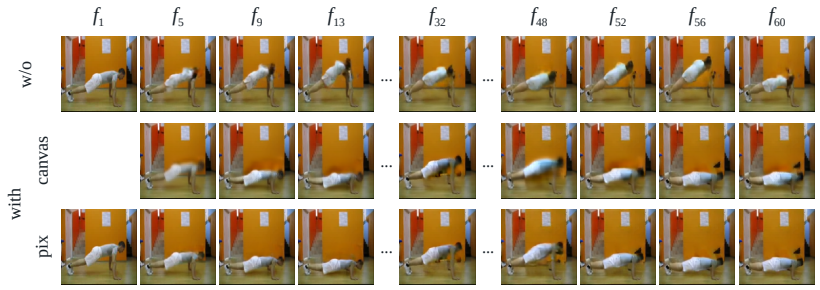


Figure 8: Qualitative example of longer video generation generated by CanvasMAR and its non-canvas variant.

Next-Group Frame Prediction. For the next-K-frame prediction model, we slightly modify the architecture by appending K *canvas heads*, each implemented as a two-layer MLP, to the output of the Canvas ViT, as shown in Figure 9. We also insert lightweight temporal attention layers after every four spatial layers, following the decomposed attention design of Gupta et al. [17], to ensure temporal consistency. These canvas heads provide the spatial conditioning canvases for generating the next K frames within the Spatial MAR.

C Broader Impacts

CanvasMAR improves the efficiency and fidelity of autoregressive video prediction, which may benefit applications that require fast future-frame forecasting, such as robotics, embodied agents, interactive simulation, and video understanding systems. Its few-step sampling design can also reduce inference latency and computational cost, making high-quality video prediction more accessible in resource-constrained settings.

At the same time, advances in video generation and prediction can contribute to the broader capability of synthesizing realistic visual content. Such techniques may be misused to create misleading synthetic videos or to support disinformation if deployed without provenance tracking, watermarking, or appropriate access controls. We therefore release and document CanvasMAR as a research contribution, and will follow responsible release practices, including clear model documentation, usage restrictions where appropriate, and compatibility with content authenticity or detection tools.

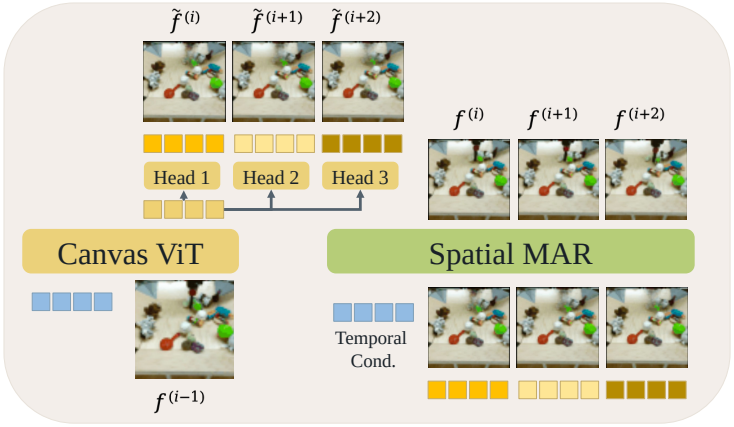


Figure 9: Next-group frame prediction.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state the main contributions of CanvasMAR, including canvas-based conditioning, motion-aware sampling, and compositional classifier-free guidance. The claimed gains are supported by the component-level and system-level experiments in Section 5, and the scope is limited to video prediction in Section 6.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The conclusion discusses failure cases on highly dynamic videos where the canvas can become overly blurred, with examples in Appendix A.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.

- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper does not present formal theoretical conclusions.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper describes the architecture, training objectives, sampling protocol, datasets, and evaluation settings in Section 4, Section 5, Appendix A, and Appendix B. Model configurations and hyperparameters are listed in Table 3.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case

of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.

- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The paper uses public datasets and cites the relevant resources. Code and instructions for faithfully reproducing the main experimental results will be provided in the supplementary material, together with the architecture, training settings, and sampling protocols described in the main paper and appendix.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The experimental setup describes the model implementation, training objectives, datasets, and evaluation tasks in Section 5. Additional dataset sampling protocols and model hyperparameters are provided in Appendix A and Table 3.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We follow the standard evaluation protocols used in previous work for FVD-based video prediction comparisons, as described in Appendix A. For example in BAIR, the reported debiased FVD already uses repeated conditioning-frame sampling 100 times to reduce sampling bias.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix A states that all models are trained on NVIDIA A100 80G GPUs, and the paper reports inference latency in Figure 5 and Appendix A.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: The research uses standard public video benchmarks and does not involve human subjects, private data, or deployment decisions. We have reviewed the NeurIPS Code of Ethics and are not aware of any deviation from it.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: Appendix C discusses potential positive impacts, including efficient video prediction for robotics, embodied agents, simulation, and video understanding. It also discusses potential misuse for misleading synthetic video generation and suggests responsible release practices.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper does not release a high-risk pretrained generative model or a newly scraped dataset as part of the submission. Therefore, release safeguards are not applicable to the current paper.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The paper credits the datasets, codebases, models, and methods it builds upon through citations, including BAIR, UCF-101, Kinetics-600, NOVA, OmniTokenizer, and DFoT. These assets are publicly available with licenses or terms of use, and we respect their corresponding usage requirements.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not introduce or release a new dataset, benchmark, code package, or model checkpoint as a new asset. It proposes and evaluates a method using existing datasets and baselines.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The research does not involve crowdsourcing experiments or human-subject studies. All evaluations are performed on existing video datasets.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The research does not involve crowdsourcing or human-subject studies, so IRB approval or equivalent review is not applicable. The experiments use existing public benchmark datasets.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The core method development does not use LLMs as an important, original, or non-standard component. Any ordinary writing, editing, or formatting assistance would not affect the methodology, scientific rigor, or originality of the research.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.