

VFRTok: Variable Frame Rates Video Tokenizer with Duration-Proportional Information Assumption

Tianxiong Zhong^{1*}, Xingye Tian², Boyuan Jiang², Xuebo Wang^{2†},
Xin Tao², Pengfei Wan², Zhiwei Zhang^{1†}

¹Beijing Institute of Technology, ²Kling Team, Kuaishou Technology
inkosizhong@gmail.com, {tianxingye, jiangboyuan, wangxuebo}@kuaishou.com,
{taoxin, wanpengfei}@kuaishou.com, zwzhang@bit.edu.cn,

Abstract

Modern video generation frameworks based on Latent Diffusion Models suffer from inefficiencies in tokenization due to the Frame-Proportional Information Assumption. Existing tokenizers provide fixed temporal compression rates, causing the computational cost of the diffusion model to scale linearly with the frame rate. The paper proposes the Duration-Proportional Information Assumption: the upper bound on the information capacity of a video is proportional to the duration rather than the number of frames. Based on this insight, the paper introduces VFRTok, a Transformer-based video tokenizer, that enables variable frame rate encoding and decoding through asymmetric frame rate training between the encoder and decoder. Furthermore, the paper proposes Partial Rotary Position Embeddings (RoPE) to decouple position and content modeling, which groups correlated patches into unified tokens. The Partial RoPE effectively improves content-awareness, enhancing the video generation capability. Benefiting from the compact and continuous spatio-temporal representation, VFRTok achieves competitive reconstruction quality and state-of-the-art generation fidelity while using only 1/8 tokens compared to existing tokenizers. The code and weights are released at: <https://github.com/KwaiVGI/VFRTok>.

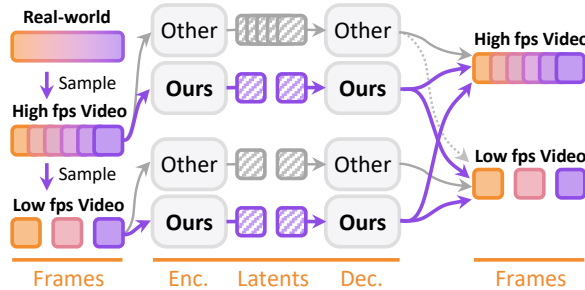


Figure 1: VFRTok is based on the Duration-Proportional Information Assumption. The number of tokens for other tokenizers grows with frame rate. VFRTok maintains a fixed length latents tied to video duration and supports asymmetric frame-rate encoding and decoding.

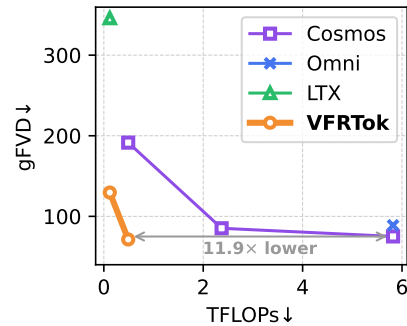


Figure 2: Efficiency-quality trade-off, where lower-left indicates better performance. VFRTok provides a more efficient latent representation.

*This work was conducted during the author's internship at Kling Team.

†Corresponding Authors.

1 Introduction

Recently, Latent Diffusion Model (LDM) is widely used in image [2, 21, 24, 25, 36, 38, 39] and video generation [1, 18, 22, 35, 41], comprising two main components: a tokenizer and a diffusion model. The tokenizer compresses data from the original high-dimensional pixel space to a low-dimensional latent space, which reduces the training and inference overhead of the Diffusion Transformers (DiT) by a quadratic factor. The video tokenizers [1, 11, 18, 33, 35, 41] eliminate intra- and inter-frame redundancy in the video by simultaneously compressing both temporal and spatial dimensions.

Existing video tokenizers [1, 11, 18, 33, 35, 41] are built upon the Frame-Proportional Information Assumption, which assumes a fixed compression rate for a given number of video frames (Figure 1). These tokenizers are trained on and designed to generate videos with a fixed frame rate. High frame rate videos require a larger number of tokens for representation, resulting in the number of tokens increase linearly with the frame rate, which significantly increases the computational overhead.

Video is the result of continuous space-time being sampled uniformly. The amount of observable information in continuous space-time serves as the natural upper bound on the information contained in the video. Intuitively, when the video frame rate increases from 12 frame per second (FPS) to 24 FPS, the change can be clearly observed, whereas the difference between 60 FPS to 120 FPS yields more subtle changes. When a camera samples a motion trajectory $x(t)$ at a sampling frequency f_s , the resulting discrete samples can be used to estimate the continuous trajectory. An interpolation algorithm is typically employed to reconstruct the continuous motion trajectory $\hat{x}(t)$ from these discrete observations. According to interpolation error estimation theory in numerical analysis, the upper bound of the estimation error E_{max} is related to f_s as follows:

$$E_{max} = \sup_t |x(t) - \hat{x}(t)| \leq \frac{C \cdot \sup_t |x^{(k)}(t)|}{f_s^k}. \quad (1)$$

where k represents the order of accuracy of the interpolation algorithm, $\sup_t |x^{(k)}(t)|$ is an upper bound on the k^{th} derivative of the true trajectory $x(t)$, and C is a positive constant. Equation (1) implies that the information gain diminishes as the frame rate increases.

Motivated by this insight, we propose the Duration-Proportional Information Assumption, which guides the design of compression rates that scale with video duration (Figure 1). Specifically, we introduce a **Variable Frame Rates video Tokenizer**, VFRTok, which is a query-based Transformer tokenizer and enables the encoder and decoder to process different frame rates. VFRTok uses asymmetric frame rate training between the encoder and decoder to learn continuous spatio-temporal representations, that enables the generation of videos at arbitrary frame rates.

Furthermore, we observed that both existing tokenizers [1, 11, 33] and VFRTok exhibit a grid-based pattern, where each latent token tends to attend to a fixed spatial location within the temporal sequence. To achieve a more compact representation, we aim to strengthen VFRTok’s content-awareness. Our analysis illustrates that the latent representations in VFRTok are strongly influenced by the positional prior introduced by Rotary Position Embeddings (RoPE), which hinders effective content modeling. Therefore, we propose Partial RoPE, which applies RoPE only to a subset of the attention heads, encouraging a functional separation between positional and content encoding. Experimental results show that the Partial RoPE effectively enhances the model’s capacity for content modeling.

Benefiting from these designs, VFRTok efficiently reconstructs and generates videos. For example, as shown in Figure 2, VFRTok achieves better generation results than existing tokenizers [1, 33] while requiring $11.9\times$ less computation overhead. Meanwhile, VFRTok natively supports video frame interpolation, enabling frame rates to be increased from 12 FPS to 120 FPS. In summary, we highlight the main contributions:

1. We propose the Duration-Proportional Information Assumption and the first high-compression video tokenizer with variable frame rate.
2. We introduce Partial RoPE to mitigate the influence of video patch position priors on latent tokens and enhance content-awareness.
3. Experiments show that we can achieve comparable reconstruction and state-of-the-art generation while using only $1/8$ tokens compared to existing tokenizers.

2 Related Work

2.1 Video Tokenizer

Video data consists of a series of continuously changing frames and video tokenizers compress the video in both temporal and spatial dimensions. Therefore, video generation task must consider inter-frame consistency to avoid problems such as flickering and jittering. Early LDM [22] directly uses image tokenizers in a frame-by-frame compression pattern. In contrast, modern video tokenizers [1, 11, 16, 18, 31, 33, 35, 41] generally use 3D computation modules that include the temporal dimension.

Mainstream video tokenizers generally provide a compression rate of $4 \times 8 \times 8$ [16, 18, 33, 35, 41], which meaning a 16-frame video with 256×256 resolution can be compressed into $4 \times 32 \times 32 = 4096$ tokens. A few works propose tokenizers with higher compression rates [1, 11, 20, 31]. For instance, Cosmos Tokenizer [1] leverages wavelet transform and provides a series of tokenizers with compression rates from $4 \times 8 \times 8$ to $8 \times 16 \times 16$. LTX-VAE [11] cascades additional downsample layers and achieves a compression rate of up to $8 \times 32 \times 32$ through a spatial-to-depth approach.

However, existing video tokenizers [1, 11, 16, 18, 31, 33, 35, 41] rely on the Frame-Proportional Information Assumption and process sequences with fixed frame rates. Different from these tokenizers, which model discrete video frame pixel space, VFRTok models continuous spatio-temporal information using asymmetric frame rate training strategy between the encoder and decoder.

2.2 Query-based Visual Tokenizer

Unlike traditional grid-based tokenizers [2, 24, 25], query-based image tokenizers [5, 6, 37] encode images into compact 1D latent representations. These tokenizers employ a Transformer-based framework and use learnable latent tokens to query information from the image patches. For instance, TiTok [37] proposes a VQ-based tokenizer and represent an image with 32 tokens for reconstruction and MaskGIT [4]-style generation. SoftVQ-VAE [5] leverages soft categorical posteriors to increase the representation capacity of the latent space, which can be applied to both autoregressive- and diffusion-based image generation. MAETok [6] further improves the diffusibility [6, 36] of vanilla AutoEncoder (AE) by regularizing it with mask modeling and auxiliary decoders. LARP [32] extends the query-based structure to video tokenizer, adapt to autoregressive video generative models.

The query-based image tokenizers [5, 6, 37] achieve comparable or better performance than the grid-based image tokenizers [2, 24, 25], using significantly fewer tokens. This shows that the query-based structure has the potential to better eliminate redundant information. More importantly, it provides the possibility to encode variable-length data into fixed-length latent representations.

3 Method

3.1 Architecture

Given a video $X \in \mathbb{R}^{F \times H \times W \times 3}$, we want to obtain its compressed representation $Z = \mathcal{E}(X) \in \mathbb{R}^{N \times d}$, where N represents the number of tokens, and the reconstructed video $\hat{X} = \mathcal{D}(Z) \in \mathbb{R}^{F \times H \times W \times 3}$. Figure 3 illustrates the architecture of VFRTok. We adopt the AE architecture and use ViT [9] as the backbone for both the encoder and decoder. On the encoder side, the input video X is patchified into a series of spatial-temporal patch tokens $x \in \mathbb{R}^{\frac{F}{p_F} \times \frac{H}{p_H} \times \frac{W}{p_W} \times h}$, where $p_F \times p_H \times p_W$ is the patch size and h is the hidden dimension. Similarly, the decoder recovers the reconstructed patch tokens $\hat{x} \in \mathbb{R}^{\frac{F}{p_F} \times \frac{H}{p_H} \times \frac{W}{p_W} \times h}$ back to the pixel-space $\hat{X}$.

VFRTok is based on the Duration-Proportional Information Assumption, which requires a continuous spatio-temporal representation. We adopt a query-based approach [5, 6, 37] that encodes video by querying grid-based patch tokens with fixed-length latent tokens, and decodes by reversing this process, where patch tokens query latent representations. To align equal-duration videos of different frame rates, we modify the temporal modeling in 3D RoPE [19, 28, 34] from frame-index-based position encoding [33] to a timestamp-based approach, detailed in Section 3.2. Furthermore, as described in Section 3.3, we propose an improved RoPE implementation, Partial RoPE, which applies RoPE to a subset of attention heads to decouple positional information from content modeling.

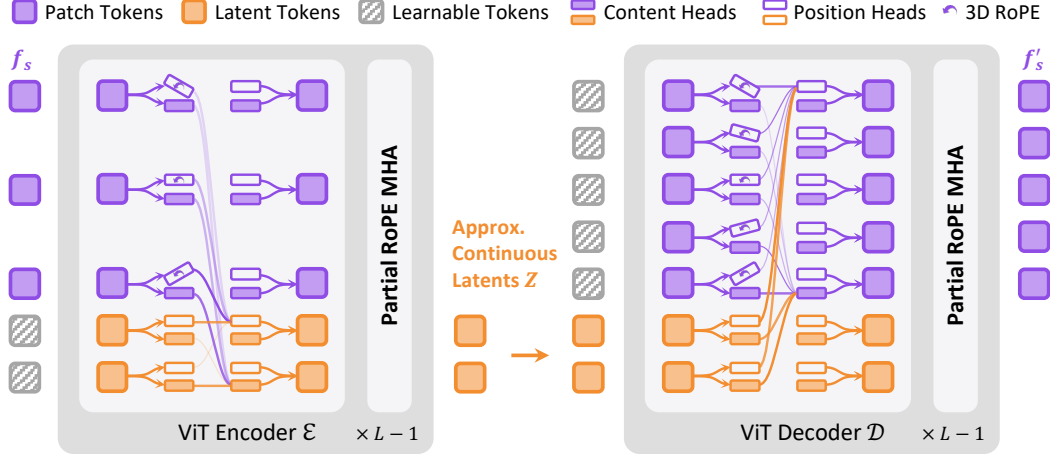


Figure 3: VFRTok adopts a query-based ViT architecture. VFRTok models variable-length patch tokens with fixed-length latent tokens, to support the encoding and decoding of variable frame rate videos. VFRTok further introduces Partial RoPE, which applies RoPE to a subset of attention heads to decouple positional information from content modeling.

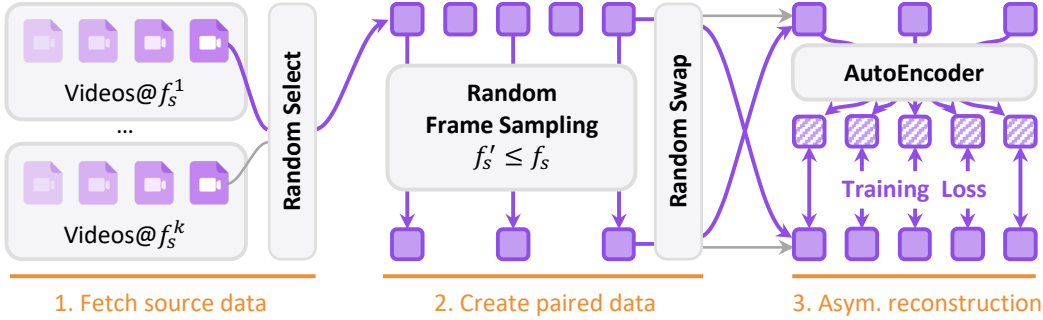


Figure 4: Asymmetric frame rate training strategy between encoder and decoder. We construct paired high and low frame rate videos to learn continuous spatio-temporal representations.

3.2 Duration-Proportional Compression

VFRTok accepts data of any frame rate but with equal duration, learns the continuous spatio-temporal information it represents, and encodes this information into fixed-length latent representations. Specifically, it concatenates learnable latent tokens $z \in \mathbb{R}^{N \times h}$ with the input x before ViT encoding. The patch tokens are dropped at the bottleneck of the encoder, and only the latent tokens are retained and mapped to the low-dimension compressed representation Z . On the decoder side, Z is first remapped to the hidden feature $\hat{z}$, and concatenated with the learnable patch tokens $\hat{x}$. As shown in Figure 3, based on the flexible architecture, VFRTok allows the encoder and decoder to apply different frame rates. Since they represent data of the same duration, videos with higher FPS contain more frames, which in turn translates into more video patches.

To achieve Duration-Proportional compression, we use 3D RoPE [28] to model the positional dependencies among patch tokens and replace the frame-driven rotation angle to a timestamp-driven formulation. Specifically, we first generalize RoPE [28] to video data by splitting the channels in each attention head into three parts, which are used to encode the positional information along the temporal, vertical, and horizontal spatial axis. Given a patch token at temporal-spatial position (t, j, k) in a $F \times H \times W$ patch token space, the rotation matrix $R_{t,j,k} \in \mathbb{R}^{n \times n}$ can be represented as:

$$R_{t,j,k} = \begin{pmatrix} R_t^T & 0 & 0 \\ 0 & R_j^H & 0 \\ 0 & 0 & R_k^W \end{pmatrix}, R_i^* = \begin{pmatrix} R_{i,1}^* & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & R_{i,\frac{n}{6}}^* \end{pmatrix}, R_{i,c}^* = \begin{pmatrix} \cos \theta_{i,c}^* & -\sin \theta_{i,c}^* \\ \sin \theta_{i,c}^* & \cos \theta_{i,c}^* \end{pmatrix}, \quad (2)$$

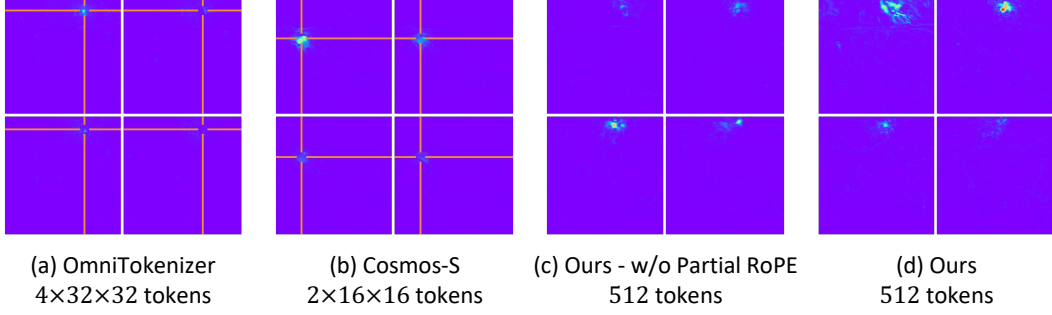


Figure 5: Visualization of the region affected by a single token. The heat map is an overlay of 100 samples, showing the 1st, 5th, 9th, and 13th frames for each method. The reference lines are drawn in the grid-based approaches to indicate the spatial position of the token within the grid.

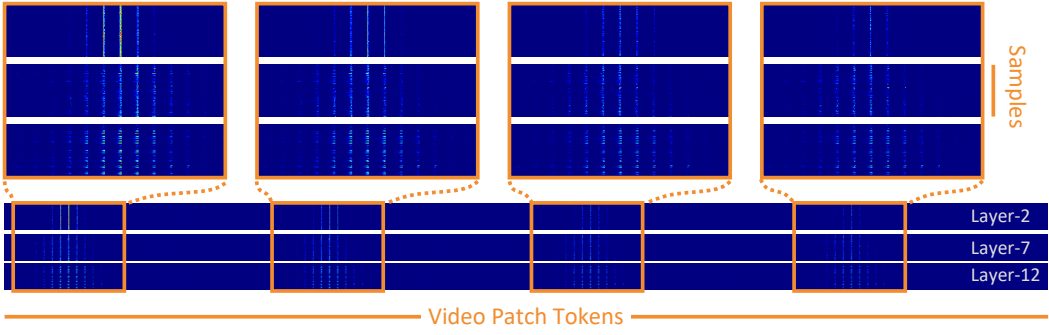


Figure 6: The latent-to-patch attention map from different Transformer layers reveals the interference of position prior on content modeling. Each row represents the intensity of information flow across patch tokens, and rows correspond to different samples.

where n represents the number of channels of an attention head, $\theta_{i,c}$ is the rotation angle of the c -th channel of the i -th token. Specifically, as shown in Equation (3), VFRTok directly uses the spatial index $\{j, k\}$ of the patch to calculate the rotation angle $\{\theta_{j,c}^H, \theta_{k,c}^W\}$, and converts the temporal index t to a timestamp t/f_s to compute the corresponding rotation angles $\theta_{t,c}^F$.

$$\theta_{t,c}^F = C \times \frac{t}{f_s} \times 10000^{-\frac{6c}{n}}, \theta_{j,c}^H = j \times 10000^{-\frac{6c}{n}}, \theta_{k,c}^W = k \times 10000^{-\frac{6c}{n}}, \quad (3)$$

where f_s is the frame rate and C is an optional normalization coefficient. This makes videos of the equal duration share the same maximum rotation angle. Meanwhile, as the frame rate increases, the angular difference between adjacent frames decreases.

As shown in Figure 4, we employ the encoder-decoder asymmetric frame rate training strategy to guide the latents in encoding continuous spatio-temporal information. Firstly, we divide the dataset into multiple buckets according to their original frame rate. For each batch, we randomly fetch data with frame rate f_s from a bucket. Secondly, we randomly select a downsampling factor $\tau \leq 1$ from a predefined range and extract frames with $f'_s = f_s \cdot \tau$ from the video. The two video sequences are randomly swapped to determine the frame rate of VFRTok's encoder f_s^E and decoder f_s^D . Finally, it encodes the video sequence with f_s^E and learns to reconstruct the video with f_s^D . The configurations vary across GPUs, enabling VFRTok to learn more continuous spatio-temporal representations.

3.3 Partial RoPE

To analyze what information in the video is compressed by each token, we use PCA [14] to ablate latent tokens individually, retain the remaining tokens for reconstruction, and observe the degradation position of the reconstructed video.

$$\Delta X_{\setminus i} = |\mathcal{D}(Z) - \mathcal{D}(Z_{\setminus i})|, Z_{\setminus i} = (Z_0, \dots, Z_{i-1}, \text{PCA}_i^{-1}(\mathbf{0}), Z_{i+1}, \dots, Z_N), \quad (4)$$

where $\text{PCA}_i^{-1}(\cdot)$ is the inverse PCA based on the coefficient corresponding to the i -th token Z_i across on a batch of samples, and $|\cdot|$ is an element-wise absolute. As shown in Figure 5(a,b,c), the latents of existing methods [1, 33] and the vanilla query-based approach exhibits a regular grid property. The token corresponds to pixels at a relatively fixed position across different frames. Existing methods [1, 33] are based on grid designs, and their tokens strictly follow the guidance of the grid. In contrast, our query-based approach is expected to provide more content-aware latent expressions. Therefore, we investigate the underlying cause.

We visualize the latent-to-patch attention map of the decoder across all layers and heads. Formally, for a given attention map $A \in \mathbb{R}^{(M+N) \times (M+N)}$, where $M = \frac{F}{p_F} \times \frac{H}{p_H} \times \frac{W}{p_W}$ is the number of patch tokens, it can be considered as a block matrix:

$$A = \begin{pmatrix} A_{\text{patch}} \in \mathbb{R}^{M \times M} & A_{\text{latent} \rightarrow \text{patch}} \in \mathbb{R}^{M \times N} \\ A_{\text{patch} \rightarrow \text{latent}} \in \mathbb{R}^{N \times M} & A_{\text{latent}} \in \mathbb{R}^{N \times N} \end{pmatrix}. \quad (5)$$

For a given sample, we extract each column from $A_{\text{latent} \rightarrow \text{patch}}$, which represents the intensity of information flow across patch tokens. We concatenate together the arrays of multiple samples to form Figure 6, where the rows correspond to samples and the columns correspond to video patch tokens. Figure 6 reveals that the attention is dominated by positional prior. Specifically, the attention distribution shows a long-short pattern across different samples, where the long periodicity arises from temporal priors, while short periodicity originates from spatial priors.

To alleviate this problem, we propose Partial RoPE, which divides the attention heads into position heads and content heads. For the position heads, we adopt the original 3D RoPE, while for the content heads, we simply remove the RoPE to adequately learn content information. We use τ_{RoPE} to control the proportion of position heads. The experimental results demonstrate that Partial RoPE effectively enhances the content-awareness and the generation quality of VFRTok.

4 Experiments

4.1 Setup

Training details. We train VFRTok ϕ using the standard reconstruction objective.

$$\mathcal{L} = \mathcal{L}_{\text{recon}} + \lambda_1 \mathcal{L}_{\text{percept}} + \lambda_2 \cdot \lambda_{\nabla} \mathcal{L}_{\text{adv}}, \quad \lambda_{\nabla} = \frac{\|\nabla_{\phi} (\mathcal{L}_{\text{recon}} + \lambda_1 \mathcal{L}_{\text{percept}})\|}{\|\nabla_{\phi} \mathcal{L}_{\text{adv}}\|}, \quad (6)$$

where $\mathcal{L}_{\text{recon}}$, $\mathcal{L}_{\text{percept}}$, and $\mathcal{L}_{\text{adv}}$ are the L1 reconstruction loss, perceptual loss [13, 17], and adversarial loss [10], respectively, and λ_{∇} represents adaptive weight. The hyperparameters are set to $\lambda_1 = 1$ and $\lambda_2 = 0.2$. The patch size of VFRTok is set to $p_F \times p_H \times p_W = 4 \times 8 \times 8$. The Partial RoPE ratio is set to $\tau_{\text{RoPE}} = 0.5$, indicating that 6 of the 12 attention heads employ RoPE. An implicit advantage of VFRTok is that the number of tokens N and channels d can be easily adjusted. To balance generation quality and training efficiency, we provide VFRTok-L and VFRTok-S with the same latent capacity but differ in token count, $Z^L \in \mathbb{R}^{512 \times 32}$ and $Z^S \in \mathbb{R}^{128 \times 128}$. To evaluate the tokenizers, we modified LightningDiT-XL/1 [36] to support video generation. If not specified, the models process video with 2/3s duration, which is 16 frames at $f_s = 24$. For fair comparison, the first frame of the image and video joint tokenizers [1, 11, 33] decoded using the image token, is not counted in the metrics, nor counted in the model calculation costs.

Datasets. VFRTok is trained in a three-stage manner. First, it is initialized on the ImageNet-1K [8] for 30,000 steps with a batch size of 512. Then, it is pre-trained on the K600 dataset [3] for 200,000 steps with a batch size of 64, employing asymmetric FPS training $f_s^{\mathcal{E}} = f_s^{\mathcal{D}} \in \{12, 18, 24, 30\}$. Finally, 22 sequences of 120 FPS data from the BVI-HFR dataset [7] are added for 100,000 steps with a batch size of 16, using FPS settings $f_s^{\mathcal{E}}, f_s^{\mathcal{D}} \in \{12 + 6k \mid k = 0, 1, \dots, 18\}$. Reconstruction evaluation is performed on the K600 [3] validation set and the UCF101 [26] dataset. VFRTok-L and VFRTok-S are trained on a single node equipped with 8 H800 GPUs, requiring 4 days, respectively. The DiT models [36] are trained and evaluated on the K600 [3] and UCF101 [26] datasets, respectively, using label-based Classifier-Free Guidance [12] (CFG). To demonstrate the advantages of VFRTok in high frame rate video generation, the DiT model was also trained on 60 FPS data from the LAVIB [27] dataset. DiTs [36] are trained for 100,000 steps with a batch size of 128 on UCF101 [26] and K600 [3] datasets, and 50,000 steps with a batch size of 48 on LAVID [27] dataset, respectively. The

Table 1: Comparison of reconstruction performance across multiple datasets for different tokenizers. Gray highlights indicate cases where VFRTok is superior or comparable.

Method	#Tokens	#Dim.	K600				UCF101			
			PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	rFVD $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	rFVD $\downarrow$
Omni [33]	4096	8	29.35	0.9143	0.0573	5.14	28.95	0.9239	0.0505	10.21
Cosmos-L [1]	4096	16	33.34	0.9284	0.0546	3.28	33.42	0.9372	0.0439	5.55
Cosmos-M [1]	2048	16	31.66	0.9068	0.0710	6.77	31.70	0.9177	0.0575	13.67
Cosmos-S [1]	512	16	28.46	0.8445	0.1209	70.26	28.26	0.8577	0.1046	104.51
LTX [11]	128	128	32.04	0.9100	0.0582	22.11	32.02	0.9202	0.0508	35.32
VFRTok-L	512	32	31.63	0.9104	0.0394	4.64	31.54	0.9193	0.0391	13.79
VFRTok-S	128	128	31.55	0.9089	0.0401	6.02	31.50	0.9178	0.0401	15.55

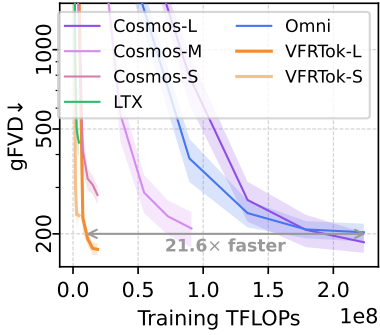


Figure 7: Convergence speed of different tokenizers on UCF101.

Table 2: Comparison of unconditional and CFG video generation in terms of gFVD $\downarrow$ and TFLOPs $\downarrow$. Best results are **bolded**; gray highlights indicate VFRTok-S superiority.

Method	TFLOPs	K600		UCF101	
		w/o CFG	w/ CFG	w/o CFG	w/ CFG
Omni [33]	5.82	521.89	242.54	480.60	88.89
Cosmos-L [1]	5.82	620.07	302.58	476.08	75.11
Cosmos-M [1]	2.37	554.95	125.02	497.01	85.22
Cosmos-S [1]	0.49	569.58	210.21	678.37	191.49
LTX [11]	0.12	615.28	358.48	735.38	345.82
VFRTok-L	0.49	323.37	124.78	377.50	71.34
VFRTok-S	0.12	412.97	131.34	443.41	129.55

training cost correlates with the number of latent tokens, ranging from 5 hours using 8 H800 GPUs (VFRTok-S) to 3 days using 16 H800 GPUs (Cosmos-L [1]).

Metrics. For the reconstruction task, we use PSNR, SSIM, and LPIPS [40] to perform frame-wise evaluation. Meanwhile, we use reconstruction FVD [30] (rFVD) as a spatio-temporal metric. For the generation task, we use generation FVD [30] (gFVD) to evaluate frame quality with and without CFG [12]. We use floating-point operations (TFLOPs) to evaluate the calculation costs of all DiTs.

4.2 Video Reconstruction

Table 1 reports reconstruction metrics of various methods [1, 11, 33] across different datasets [3, 26]. It shows that all models in the VFRTok family share similar reconstruction quality, attributed to their shared latent token capacity. VFRTok achieves comparable quality with Cosmos-M [1] and OmniTokenizer [33], using only 1/4 latent capacity. In comparison, LTX-VAE [11] performs better than VFRTok on PSNR, but VFRTok achieves significantly better LPIPS [40] and rFVD [30]. It is worth noting that Cosmos [1] and LTX-VAE [11] use larger and licensed training dataset, so VFRTok still has the potential to achieve better results.

4.3 Video Generation

As shown in Table 2, VFRTok-L achieves state-of-the-art (SOTA) performance in the unconditional and CFG [12] generation on UCF101 [26] and K600 [3] dataset. The overhead of VFRTok-L is only 8.4% of OmniTokenizer and Cosmos-L, and 20.6% of Cosmos-M. Furthermore, we use gray highlighting to indicate cases where VFRTok-S is superior. For example, VFRTok-S consistently outperforms LTX-VAE [11] and Cosmos-S [1] in all tasks. We show the convergence speed of different methods on the UCF101 [26] dataset in Figure 7, where the error bars reflect different CFG scales $\{2, 3, 4\}$. Due to the high computational cost of DiT [36] when using existing tokenizers [1, 33], we randomly conduct a fair comparison on 1,000 samples. The experimental result shows that

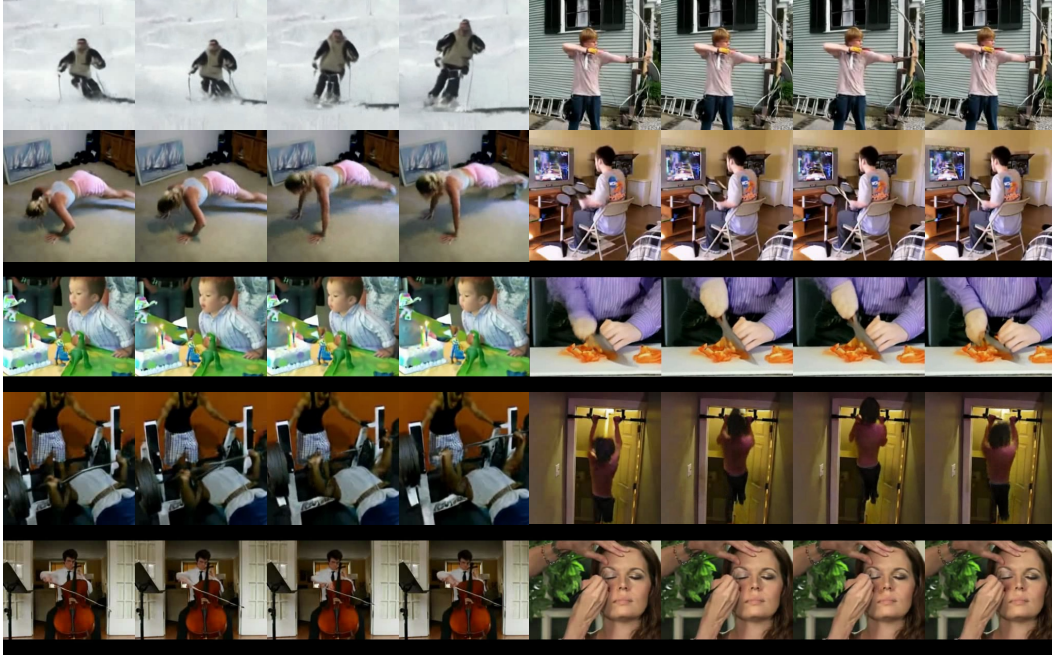


Figure 8: Qualitative results for video generation with CFG on the UCF101 dataset.

Table 3: Comparison of unconditional video generation on LAVID at 60 FPS.

Method	#Tokens	TFLOPS↓	gFVD↓
Omni [33]	10240	22.67	375.47
Cosmos-L [1]	10240	22.67	1552.04
Cosmos-M [1]	5120	7.95	1176.80
Cosmos-S [1]	1280	1.35	863.36
VFRTok-L	512	0.49	148.68

Table 4: Effectiveness of Partial RoPE.

τ_{RoPE}	Reconstruction				Generation
	PSNR↑	SSIM↑	LPIPS↓	rFVD↓	gFVD↓
0	12.17	0.2204	0.7897	4652.25	-
0.25	29.24	0.8886	0.0545	18.38	199.86
0.5	30.87	0.9106	0.0451	16.33	147.41
0.75	30.91	0.9094	0.0437	16.17	174.44
1	30.85	0.9083	0.0441	16.84	208.42

VFRTok converges significantly faster; for example, VFRTok-L achieves a maximum convergence speed of $21.6\times$ that of OmniTokenizer [33]. Qualitative results of generation are shown in Figure 8.

We also provide the generation results on the LAVIB [27] 60 FPS dataset in Table 3. As shown, all baseline methods [1, 11, 33] require $2.5\times$ the tokens to represent videos with higher frame rates, resulting in a near-quadratic increase in computation cost. For example, for videos of the same duration, the computational cost of DiT based on OmniTokenizer [33] at 24 FPS is 5.82, where as at 60 FPS it increases to 22.67. In contrast, VFRTok maintains a constant token count regardless of frame rate, which not only improves efficiency but also enables significantly faster DiT convergence. As a result, VFRTok-L achieves the optimal gFVD [30] score among all compared methods.

4.4 Ablation Study

Partial RoPE. We perform an ablation study on Partial RoPE by training DiT [36] with varying Partial RoPE factors $\tau_{\text{RoPE}} \in \{0, 0.25, 0.5, 0.75, 1\}$. As shown in Table 4, smaller τ_{RoPE} fails to provide adequate positional guidance, resulting in poorer reconstruction and generation quality, while higher τ_{RoPE} results in excessive positional bias (Figure 6), leading to suboptimal performance. Specifically, we find that when $\tau_{\text{RoPE}} = 0$, ViT [2] entirely loses its capacity for positional encoding, rendering it incapable of reconstructing or generating videos. We adopt $\tau_{\text{RoPE}} = 0.5$ as our default setting, as it achieves an optimal balance between position priors and content modeling. The visualization of latent-to-patch attention map belonging to the position heads and content heads is illustrated in Figure 9. VFRTok effectively decouples the two patterns, and different samples in the content

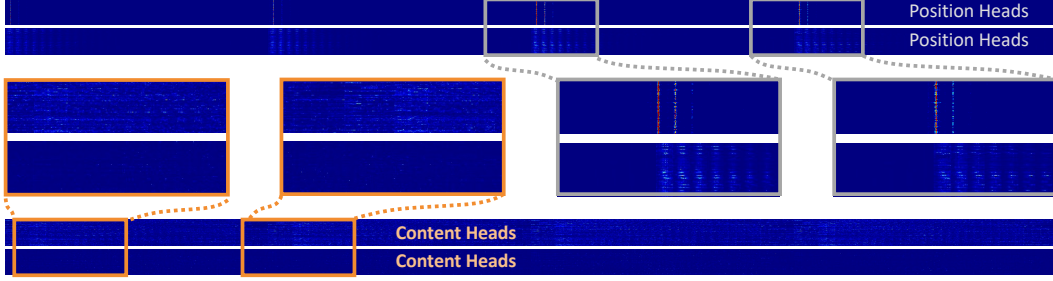


Figure 9: VFRTok decouples positional encoding and content modeling through Partial RoPE. The Position heads share similar patterns across samples (rows) indicating reduced sensitivity to content. The content heads attend to distinct patches in different samples (zoom in for details).

Table 5: Symmetric and asymmetric reconstruction performance on Adobe240fps dataset. *Symm* represents a variation of VFRTok which is trained on the symmetric reconstruction task.

Methods	Symmetric Reconstruction (PSNR)			Asymmetric Reconstruction (PSNR)		
	30fps	60fps	120fps	30 → 60fps	30 → 120fps	60 → 120fps
Symm	25.92	24.71	23.89	22.36	21.60	23.33
Ours	25.87 ↓0.05	24.53 ↓0.18	23.95 ↑0.03	24.05 ↑1.69	23.56 ↑1.96	23.90 ↑0.66

head exhibit greater diversity in attention patterns. The PCA analysis in Figure 5(d) also shows that adopting partial RoPE can increase the variability of video regions influenced by individual tokens.

Asymmetric vs. Symmetric Training. We train a VFRTok variant under a symmetric encoding strategy. Specifically, we initialized from the K600 [3] pre-trained model in stage 1 for quick verification. In stage 2, which uses both K600 [3] and BVI-HFR [7] datasets, we disabled asymmetric encoding, forcing both encoder and decoder to operate at the same frame rate. For fair comparison, we adopt Adobe240fps [29] for evaluation. As shown in Table 5, this symmetric training yields a marginally better reconstruction in the symmetric setting but suffers a pronounced degradation under asymmetric encoding. Note that our stage 1 initialization itself used asymmetric frame rates, which partially narrows the gap in asymmetric reconstruction. Overall, these results confirm that asymmetric training is crucial for enabling VFRTok to generate videos at arbitrary frame rates.



Figure 10: Results of video frame interpolation from 12 FPS to 30, 60, and 120 FPS. The original 12 FPS frame positions are indicated by triangle markers.

4.5 Video Frame Interpolation

VFRTok is designed for LDM, but also exhibits strong capabilities in Video Frame Interpolation (VFI). We evaluate VFRTok on sequences exhibiting large motion from the public UVG [23] dataset.

As shown in Figure 10, VFRTok successfully interpolates 12 FPS videos to 30, 60 and 120 FPS. A key advantage of VFRTok is its ability to perform variable frame rate interpolation, supporting arbitrary input and output frame rates. Performance on VFI can be enhanced by increasing the capacity of the latent space $\mathbb{R}^{N \times d}$ of VFRTok. Comparison with FLAVR [15] is shown in the Appendix B.

5 Conclusion

We propose the Duration-Proportional Information Assumption, where the upper bound on the observable information capacity of a video is proportional to the video duration. Under this assumption, we introduce VFRTok, a Transformer-based video tokenizer capable of encoding and decoding videos at variable frame rates. Furthermore, we introduce Partial Rotary Position Embeddings to decouple positional encoding from content modeling, thereby enhancing content-awareness, ultimately improving generation performance. Experiments show that VFRTok achieves comparable reconstruction performance and better generation quality compared to existing tokenizers using 1/8 tokens, while being $11.9\times$ faster. Meanwhile, VFRTok converges significantly faster than existing works, reducing computational cost by up to $21.6\times$. When the frame rate increases, VFRTok does not require additional denoising tokens in DiT as frame rate increases, further demonstrating its efficiency advantage. Finally, we also find that VFRTok has the potential for video interpolation, capable of interpolating 12 FPS videos up to 120 FPS.

Limitations. Dense attention limits VFRTok’s scalability to long videos. Segmenting videos into temporal slices with causal window attention is a potential solution that we leave for future work.

References

- [1] Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, et al. Cosmos world foundation model platform for physical ai. [arXiv preprint arXiv:2501.03575](#), 2025.
- [2] Fan Bao, Shen Nie, Kaiwen Xue, Yue Cao, Chongxuan Li, Hang Su, and Jun Zhu. All are worth words: A vit backbone for diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22669–22679, 2023.
- [3] Joao Carreira, Eric Noland, Andras Banki-Horvath, Chloe Hillier, and Andrew Zisserman. A short note about kinetics-600. [arXiv preprint arXiv:1808.01340](#), 2018.
- [4] 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.
- [5] Hao Chen, Ze Wang, Xiang Li, Ximeng Sun, Fangyi Chen, Jiang Liu, Jindong Wang, Bhiksha Raj, Zicheng Liu, and Emad Barsoum. Softvq-vae: Efficient 1-dimensional continuous tokenizer. [arXiv preprint arXiv:2412.10958](#), 2024.
- [6] Hao Chen, Yujin Han, Fangyi Chen, Xiang Li, Yidong Wang, Jindong Wang, Ze Wang, Zicheng Liu, Difan Zou, and Bhiksha Raj. Masked autoencoders are effective tokenizers for diffusion models. [arXiv preprint arXiv:2502.03444](#), 2025.
- [7] Duolikun Danier, Fan Zhang, and David R Bull. Bvi-vfi: a video quality database for video frame interpolation. *IEEE Transactions on Image Processing*, 32:6004–6019, 2023.
- [8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.
- [9] 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.
- [10] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11): 139–144, 2020.

- [11] Yoav HaCohen, Nisan Chiprut, Benny Brazowski, Daniel Shalem, Dudu Moshe, Eitan Richardson, Eran Levin, Guy Shiran, Nir Zabari, Ori Gordon, et al. Ltx-video: Realtime video latent diffusion. [arXiv preprint arXiv:2501.00103](#), 2024.
- [12] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. [arXiv preprint arXiv:2207.12598](#), 2022.
- [13] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part II 14*, pages 694–711. Springer, 2016.
- [14] Ian T Jolliffe. *Principal component analysis for special types of data*. Springer, 2002.
- [15] Tarun Kalluri, Deepak Pathak, Manmohan Chandraker, and Du Tran. Flavr: Flow-agnostic video representations for fast frame interpolation. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2071–2082, 2023.
- [16] 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.
- [17] Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric. In *International conference on machine learning*, pages 1558–1566. PMLR, 2016.
- [18] Zhimin Li, Jianwei Zhang, Qin Lin, Jiangfeng Xiong, Yanxin Long, Xincheng Deng, Yingfang Zhang, Xingchao Liu, Minbin Huang, Zedong Xiao, et al. Hunyuan-dit: A powerful multi-resolution diffusion transformer with fine-grained chinese understanding. [arXiv preprint arXiv:2405.08748](#), 2024.
- [19] Zikang Liu, Longteng Guo, Yepeng Tang, Tongtian Yue, Junxian Cai, Kai Ma, Qingbin Liu, Xi Chen, and Jing Liu. Vrope: Rotary position embedding for video large language models. [arXiv preprint arXiv:2502.11664](#), 2025.
- [20] Guoqing Ma, Haoyang Huang, Kun Yan, Liangyu Chen, Nan Duan, Shengming Yin, Changyi Wan, Ranchen Ming, Xiaoni Song, Xing Chen, et al. Step-video-t2v technical report: The practice, challenges, and future of video foundation model. [arXiv preprint arXiv:2502.10248](#), 2025.
- [21] Nanye Ma, Mark Goldstein, Michael S Albergo, Nicholas M Boffi, Eric Vanden-Eijnden, and Saining Xie. Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In *European Conference on Computer Vision*, pages 23–40. Springer, 2024.
- [22] 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.
- [23] Alexandre Mercat, Marko Viitanen, and Jarno Vanne. Uvg dataset: 50/120fps 4k sequences for video codec analysis and development. In *Proceedings of the 11th ACM multimedia systems conference*, pages 297–302, 2020.
- [24] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023.
- [25] 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.
- [26] Khuram 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.
- [27] Alexandros Stergiou. Lavib: A large-scale video interpolation benchmark. [arXiv preprint arXiv:2406.09754](#), 2024.
- [28] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.
- [29] Shuochen Su, Mauricio Delbracio, Jue Wang, Guillermo Sapiro, Wolfgang Heidrich, and Oliver Wang. Deep video deblurring for hand-held cameras. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1279–1288, 2017.

- [30] Thomas Unterthiner, Sjoerd Van Steenkiste, Karol Kurach, Raphaël Marinier, Marcin Michalski, and Sylvain Gelly. Fvd: A new metric for video generation. 2019.
- [31] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwei Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. arXiv preprint arXiv:2503.20314, 2025.
- [32] Hanyu Wang, Saksham Suri, Yixuan Ren, Hao Chen, and Abhinav Shrivastava. Larp: Tokenizing videos with a learned autoregressive generative prior. arXiv preprint arXiv:2410.21264, 2024.
- [33] Junke Wang, Yi Jiang, Zehuan Yuan, Bingyue Peng, Zuxuan Wu, and Yu-Gang Jiang. Omnitokenizer: A joint image-video tokenizer for visual generation. Advances in Neural Information Processing Systems, 37:28281–28295, 2024.
- [34] Xilin Wei, Xiaoran Liu, Yuhang Zang, Xiaoyi Dong, Pan Zhang, Yuhang Cao, Jian Tong, Haodong Duan, Qipeng Guo, Jiaqi Wang, et al. Videorope: What makes for good video rotary position embedding? arXiv preprint arXiv:2502.05173, 2025.
- [35] 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.
- [36] Jingfeng Yao, Bin Yang, and Xinggang Wang. Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. arXiv preprint arXiv:2501.01423, 2025.
- [37] Qihang Yu, Mark Weber, Xueqing Deng, Xiaohui Shen, Daniel Cremers, and Liang-Chieh Chen. An image is worth 32 tokens for reconstruction and generation. Advances in Neural Information Processing Systems, 37:128940–128966, 2024.
- [38] Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. arXiv preprint arXiv:2410.06940, 2024.
- [39] Kaiwen Zha, Lijun Yu, Alireza Fathi, David A Ross, Cordelia Schmid, Dina Katabi, and Xiuye Gu. Language-guided image tokenization for generation. arXiv preprint arXiv:2412.05796, 2024.
- [40] 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, pages 586–595, 2018.
- [41] Zangwei Zheng, Xiangyu Peng, Tianji Yang, Chenhui Shen, Shenggui Li, Hongxin Liu, Yukun Zhou, Tianyi Li, and Yang You. Open-sora: Democratizing efficient video production for all. arXiv preprint arXiv:2412.20404, 2024.

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: We have stated main claims in the abstract and introduction.

Guidelines:

- The answer NA 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 NA 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: We discussed limitations and possible solutions in Section 5.

Guidelines:

- The answer NA 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: [\[Yes\]](#)

Justification: We provide theoretical assumptions and proofs in Section 1.

Guidelines:

- The answer NA 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: We present full implementation details in Section 3.

Guidelines:

- The answer NA 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: We provide code and environment description in the supplementary material.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.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://nips.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, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We present all the training and inference details in Section 4.1.

Guidelines:

- The answer NA 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 report the error bars of different methods in Figure 7.

Guidelines:

- The answer NA 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: We present the compute resources in Section 4.1.

Guidelines:

- The answer NA 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 experiments do not involve human subjects. We only use public datasets.

Guidelines:

- The answer NA 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: [NA]

Justification: This paper is a foundational research on video tokenizer.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA 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., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification:

Guidelines:

- The answer NA 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: <https://github.com/huggingface/pytorch-image-models> - Apache 2.0 license. <https://paperswithcode.com/dataset/kinetics-600> - CC BY 4.0 license. <https://paperswithcode.com/dataset/ucf101> - MIT license. <https://paperswithcode.com/dataset/lavib> - CC BY-NC-SA 4.0 license. <https://data.bris.ac.uk/data/dataset/ca830349ffcba535c2045ca9d2304faf> - Non-Commercial Government Licence for public sector information.

Guidelines:

- The answer NA 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: [NA]

Justification:

Guidelines:

- The answer NA 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: [NA]

Justification:

Guidelines:

- The answer NA 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: [NA]

Justification:

Guidelines:

- The answer NA 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 rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification:

Guidelines:

- The answer NA 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 (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.

A Implementation Details

The detailed configuration of VFRTok and LightningDiT are shown in Table 6 and Table 7.

Table 6: Training configuration of VFRTok.

Configuration	Value
image resolution	256×256
enc/dec hidden dimension	768
enc/dec #position heads	6
enc/dec #content heads	6
enc/dec #layers	12
enc/dec patch size	$4 \times 8 \times 8$
enc/dec positional embedding	3D RoPE (video), 1D APE (latent)
optimizer	AdamW
weight decay	$1e-4$
optimizer momentum	$\beta_1, \beta_2 = 0.9, 0.95$
global batch size	512 (stage1), 64 (stage2), 16 (stage3)
training steps	30k (stage1), 200K (stage2), 100K (stage3)
base learning rate	$1e-4$ (stage1 & stage2), $1e-5$ (stage2)
learning rate schedule	cosine
augmentation	horizontal flip, center crop
perceptual weight λ_1	1
discriminator	DINOv2-S
discriminator weight λ_2	0.2
discriminator start	30K
discriminator LeCAM	0.001

Table 7: Training and inference configuration of LightningDiT-XL.

Configuration	Value
hidden dimension	1152
#heads	16
#layer	28
patch size	1
positional embedding	APE
optimizer	AdamW
weight decay	0
optimizer momentum	$\beta_1, \beta_2 = 0.9, 0.95$
global batch size	128 (UCF101/K600), 48 (LAVI)
training steps	100K (UCF101/K600), 50K (LAVI)
base learning rate	$1e-4$
learning rate schedule	constant
augmentation	center crop
diffusion sampler	Euler
diffusion steps	50
CFG interval start	0.1
timestamp shift	2

Learnable token details. We adopted the same design as existing image 1D tokenizers: on the encoder side, we learn N independent latent tokens, whereas on the decoder side, we use a single shared token. Although we experimented with replacing the decoder’s shared token with a fixed-length set of independent tokens, which improves reconstruction fidelity, this change eliminated the decoder’s ability to flexibly handle variable-frame-rate decoding.

B Quantitative Results for Video Frame Interpolation

Table 8 shows quantitative video-interpolation results comparing FLAVR [15] and VFRTok. Although FLAVR is a strong video-interpolation baseline and outperforms VFRTok on this task, VFRTok was not designed primarily for interpolation. First, VFRTok employs an extremely high compression rate to enable efficient video generation, creating a tighter bottleneck than dedicated interpolation models. Second, our training set contains only 22 clips at 120fps and no 60fps videos, whereas interpolation models are typically trained on large-scale, high-frame-rate data. In summary, while VFRTok can perform interpolation, its principal application remains general video generation.

Table 8: Quantitative results for video interpolation.

	12 $\rightarrow$ 24 (2 $\times$)		30 $\rightarrow$ 60 (2 $\times$)		12 $\rightarrow$ 48 (4 $\times$)		30 $\rightarrow$ 120 (4 $\times$)		15 $\rightarrow$ 120 (8 $\times$)	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
FLAVR	26.05	0.7852	34.22	0.9529	22.17	0.6435	27.59	0.8372	21.70	0.6135
Ours	22.93	0.6724	24.06	0.7324	22.08	0.6433	23.68	0.7215	19.58	0.5279